



The Weblate Manual

Yayım 4.17

Michal Čihař

17 Nis 2023

1	Kullanıcı belgeleri	1
1.1	Weblate temelleri	1
1.2	Hesap açmak ve kullanıcı profili	1
1.3	Weblate kullanarak çeviri yapmak	10
1.4	Çevirileri indirmek ve yüklemek	20
1.5	Sözlük	23
1.6	Denetimler ve düzeltmeler	26
1.7	Arama	60
1.8	Çeviri iş akışları	65
1.9	Sık sorulan sorular	69
1.10	Desteklenen dosya biçimleri	77
1.11	Sürüm denetimi bütünleştirilmesi	103
1.12	Weblate REST API uygulaması	111
1.13	Weblate istemcisi	160
1.14	Weblate Python API	164
2	Yönetici belgeleri	167
2.1	Yapılandırma yönergeleri	167
2.2	Weblate dağıtımları	234
2.3	Weblate yazılımını güncellemek	235
2.4	Weblate yedeğini alma ve taşıma	244
2.5	Kimlik doğrulama	251
2.6	Erişim denetimi	261
2.7	Çeviri projeleri	271
2.8	Dil tanımları	289
2.9	Sürekli yerleştirme	292
2.10	Çeviri lisanslarını belirlemek	300
2.11	Çeviri süreci	302
2.12	Denetimler ve düzeltmeler	308
2.13	Kendiliğinden önerileri yapılandırmak	318
2.14	Eklentiler	329
2.15	Çeviri belleği	347
2.16	Yapılandırma	349
2.17	Örnek yapılandırma	376
2.18	Yönetim komutları	391
2.19	Duyurular	402
2.20	Bileşen listeleri	405
2.21	İsteğe bağlı Weblate modülleri	406
2.22	Weblate platformunu özelleştirmek	411
2.23	Yönetim arayüzü	414
2.24	Weblate için destek almak	422

2.25	Yasal belgeler	425
3	Katılımcı belgeleri	427
3.1	Weblate için katkıda bulunmak	427
3.2	Weblate kodlarına katkıda bulunmak	429
3.3	Weblate kaynak kodları	433
3.4	Weblate hatalarını ayıklamak	434
3.5	Weblate iç yapısı	436
3.6	Eklentileri geliřtirmek	437
3.7	Weblate ön yüzü	439
3.8	Weblate sorunlarını bildirmek	440
3.9	Weblate sınama paketi ve sürekli bütünleřtirme	440
3.10	Veri řemaları	442
3.11	Weblate yayınları	446
3.12	Güvenlik ve gizlilik	447
3.13	Weblate modüllerine katkıda bulunmak	448
3.14	Weblate hakkında	449
3.15	Lisans	449
4	Deęiřiklik gemiři	450
4.1	Weblate 4.17	450
4.2	Weblate 4.16.4	451
4.3	Weblate 4.16.3	451
4.4	Weblate 4.16.2	451
4.5	Weblate 4.16.1	451
4.6	Weblate 4.16	451
4.7	Weblate 4.15.2	452
4.8	Weblate 4.15.1	452
4.9	Weblate 4.15	452
4.10	Weblate 4.14.2	453
4.11	Weblate 4.14.1	454
4.12	Weblate 4.14	454
4.13	Weblate 4.13.1	455
4.14	Weblate 4.13	455
4.15	Weblate 4.12.2	456
4.16	Weblate 4.12.1	456
4.17	Weblate 4.12	456
4.18	Weblate 4.11.2	457
4.19	Weblate 4.11.1	457
4.20	Weblate 4.11	457
4.21	Weblate 4.10.1	458
4.22	Weblate 4.10	458
4.23	Weblate 4.9.1	459
4.24	Weblate 4.9	459
4.25	Weblate 4.8.1	460
4.26	Weblate 4.8	460
4.27	Weblate 4.7.2	461
4.28	Weblate 4.7.1	461
4.29	Weblate 4.7	461
4.30	Weblate 4.6.2	462
4.31	Weblate 4.6.1	462
4.32	Weblate 4.6	462
4.33	Weblate 4.5.3	463
4.34	Weblate 4.5.2	463
4.35	Weblate 4.5.1	464
4.36	Weblate 4.5	464
4.37	Weblate 4.4.2	465
4.38	Weblate 4.4.1	465

4.39	Weblate 4.4	465
4.40	Weblate 4.3.2	466
4.41	Weblate 4.3.1	467
4.42	Weblate 4.3	467
4.43	Weblate 4.2.2	468
4.44	Weblate 4.2.1	468
4.45	Weblate 4.2	468
4.46	Weblate 4.1.1	469
4.47	Weblate 4.1	469
4.48	Weblate 4.0.4	471
4.49	Weblate 4.0.3	471
4.50	Weblate 4.0.2	471
4.51	Weblate 4.0.1	472
4.52	Weblate 4.0	472
4.53	Weblate 3.x serisi	473
4.54	Weblate 2.x serisi	484
4.55	Weblate 1.x serisi	495
4.56	Weblate 0.x serisi	500
Python Modül Dizini		504
HTTP Routing Table		505
Dizin		508

1.1 Weblate temelleri

1.1.1 Proje ve bileşen yapısı

Weblate üzerinde çeviriler, projeler ve bileşenler olarak düzenlenir. Bir projede çok sayıda bileşen bulunabilir ve bunlar ayrı dillerin çevirilerini içerebilir. Bileşen çevrilebilir bir dosyaya karşılık gelir *GNU gettext* ya da *Android dizge kaynakları*). Projeler, bileşeni mantıksal kümeler halinde düzenlemenize yardımcı olmak için düşünülmüştür (tek bir uygulamada kullanılan tüm çevirileri gruplamak gibi).

İçeride, her projenin varsayılan olarak içindeki diğer bileşenlere yayılan ortak dizge çevirileri vardır. Bu özellik, yinelenen ve çok sürümlü çeviri yükünü hafifletir. Çevirilerin farklılaşması durumunda *Bileşen yapılandırması* için *Çevirilerin yayılmasını sağlamak* seçeneği kullanılarak çevirilerin yayılması devre dışı bırakılabilir.

Ayrıca bakınız:

../devel/integration

1.2 Hesap açmak ve kullanıcı profili

1.2.1 Hesap açılışı

Varsayılan olarak herkes projelere göz atabilir, çevirileri görüntüleyebilir ya da çeviri önerisinde bulunabilir. Yalnızca kayıtlı kullanıcıların değişiklikleri gerçekten kaydetmesine izin verilir ve çevirmen olarak yapılan her çevirinin hesabı tutulur.

Birkaç basit adımı izleyerek hesap açabilirsiniz:

1. Hesap açma formunu kimlik doğrulama bilgilerinizle doldurun.
2. Size gönderilen e-postadaki bağlantıyı izleyerek hesabınızı etkinleştirin.
3. İsteğe bağlı olarak, hangi dilleri bildiğinizi seçmek için profilinizi ayarlayın.

1.2.2 Pano

Oturum açtığınızda, proje ve bileşen özetinin yanında ilgili çevirilerdeki ilerlemeleri görürsünüz.

Varsayılan olarak, izlediğiniz projelerin bileşenleri görüntülenir ve çeviri dilleriniz seçilir.

İpucu: Gezinme sekmelerini kullanarak farklı görünümlere geçebilirsiniz.

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 'Your profile' and a list of tabs: 'Languages', 'Preferences' (active), 'Notifications', 'Account', 'Profile', 'Teams', 'Licenses', 'Audit log', and 'API access'. The 'Preferences' panel is open, showing settings for the user's profile. It includes a checkbox for 'Hide completed translations on the dashboard', a section for 'Translation editor mode' with a dropdown set to 'Full editor', a section for 'Zen editor mode' with a dropdown set to 'Top to bottom', a section for 'Number of nearby strings' with a text input set to '15', a checkbox for 'Show secondary translations in the Zen mode' which is checked, a checkbox for 'Hide source if a secondary translation exists', a section for 'Editor link' with a text input, a section for 'Special characters' with a text input, and a section for 'Default dashboard view' with two radio buttons: 'Watched translations' (selected) and 'Suggested translations'. A 'Save' button is at the bottom of the panel.

Powered by Weblate 4.17 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Menüde şu seçenekler bulunur:

- Ana menüdeki *Projeler* > *Tüm projelere göz at* Weblate kopyasındaki tüm projenin çeviri durumunu görüntüler.
- Ana menüden bir dil seçtiğinizde *Diller*, tüm projelerin çeviri durumunu, birincil dillerinizden birine göre süzülmuş olarak görüntüler.
- *İzlenen çeviriler* panoda yalnızca izlemekte olduğunuz projelerin çeviri durumunu, birincil dillerinize göre süzülmuş olarak görüntüler.

Ek olarak, açılır menüde herhangi bir sayıda *bileşen listesi*, Weblate yöneticisi tarafından önceden yapılandırılmış proje bileşenleri kümelerini de görüntüleyebilir. Bilgi almak için: :ref:`componentlists`.

Kişisel varsayılan pano görünümünüzü, kullanıcı profili ayarlarınızın *Ayarlar* bölümünden yapılandırabilirsiniz.

Not: Weblate, `settings.py` dosyasında `:settings: 'SINGLE_PROJECT'` seçeneği ile tek bir proje için yapı-

landırıldığında (bilgi almak için: [Yapılandırma](#)), kullanıcı tek bir projeye veya bileşene yönlendirileceğinden pano görüntülenmez.

1.2.3 Kullanıcı profili

Kullanıcı profiline, üst menünün sağ üst köşesindeki kullanıcı simgenize ve ardından menüden *Ayarlar* seçeneğine tıklayarak erişebilirsiniz.

Kullanıcı profilinizde ayarlarınız bulunur. Sürüm denetimi sistemi işlemlerinde adınız ve e-posta adresiniz kullanılır. Bu nedenle bu bilgilerin doğru olmasını sağlayın.

Not: Tüm dil seçimlerinde yalnızca şu anda çevirisi yapılan diller sunulur.

İpucu: Çevirmek istediğiniz diğer dilleri kullanılabilir kılmak için düğmeye tıklayarak dil isteğinde bulunan ya da ekleyin.

Diller

1.2.4 Arayüz dili

Kullanıcı arayüzünün görüntülenmesini istediğiniz dili seçin.

Çevrilen diller

Hangi dilleri çevirmeyi yeğlediğinizi seçin. Bunlar izlenen projelerin ana sayfasında görüntülenir. Böylece bu dillerdeki çevirilere daha kolay erişebilirsiniz.

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/WebateOrg — Hungarian GPL-3.0	✓						
WeblateOrg/WebateOrg — Czech GPL-3.0	✓						
WeblateOrg/WebateOrg — Hebrew GPL-3.0	✓						

İkincil diller

Çeviri sırasında size rehber olarak görüntülenecek ikincil dilleri belirtebilirsiniz. Aşağıdaki görselde, İbranice dilinin ikincil dil olarak kullanılma örneğini görebilirsiniz:

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area shows a translation task for the string 'Soubory'. The interface includes a top navigation bar with links to Dashboard, Projects, Languages, and Checks. The main translation area has input fields for Hebrew, English, and Czech. The Czech field contains the text 'Soubory'. Below the input fields are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. The right sidebar contains a 'Glossary' section and a 'String information' section. The 'String information' section shows details about the string, including its location, age, and the translation file. The bottom of the page features a footer with the text 'Powered by Weblate 4.17' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

1.2.5 Ayarlar

Varsayılan pano görünümü

Ayarlar sekmesinde, var olan hangi pano görünümünün varsayılan olarak görüntüleneceğini seçebilirsiniz. *Bileşen listesi* görünümünü seçerseniz, *Varsayılan bileşen listesi* açılır menüsünden hangi bileşen listesinin görüntüleneceğini seçmeniz gerekir.

Ayrıca bakınız:

Bileşen listeleri

Düzenleyici bağlantısı

Varsayılan olarak *Bileşen yapılandırması* içinde yapılandırılmış tarayıcıda bir kaynak kodu bağlantısı görüntülenir.

İpucu: *Düzenleyici bağlantısı* ayarını yaparak, çevrilmiş dizgelerin sürüm denetimi sistemi kaynak kodu dosyasını açmak için yerel düzenleyicinizi kullanırsınız. *Kalıp kodlaması* kullanabilirsiniz.

Genellikle `editor://open/?file={{filename}}&line={{line}}` gibi bir şey iyi bir seçenektir.

Ayrıca bakınız:

Düzenleyici için özel adres iletişim kurallarının kaydedilmesi ile ilgili bilgi almak için: [Nette belgeleri](#).

Özel karakterler

Sanal klavye içine eklenecek ek özel karakterler.

1.2.6 Bildirimler

Bildirimler sekmesinden abone olabileceğiniz bildirimleri seçebilirsiniz. İzlenen veya yönetilen projeler için seçtiğiniz bildirimler size e-posta ile gönderilir.


Bazı bildirimler yalnızca dillerinizdeki olaylar için gönderilir (çevrilecek yeni dizgeler olduğunda gibi), bazıları ise bileşen düzeyinde gönderilir (birleştirme hataları gibi). Bu iki bildirim grubu, ayarlarda görsel olarak ayrılmıştır.

İzlenen projeler ve yönetilen projeler için bildirimleri açıp kapatabilirsiniz. Her bir proje veya bileşen için başka ince ayarlar da yapılabilir (ya da sessize alınabilir). Bileşen özeti sayfasını açın ve *İzleme* menüsünden uygun seçeneği seçin.

Katkıda bulunan projeler kendiliğinden izlensin seçeneği etkinleştirilmişse, bir çeviri yaptığınız dizgelerin bulunduğu projeleri kendiliğinden izlemeye başlarsınız. Varsayılan değer *DEFAULT_AUTO_WATCH* seçeneğine bağlıdır.

Not: Kendi işlemlerinizi için bildirim almazsınız.

İpucu: Bildirim gönderimi sınırlıdır. Günde 1000 taneden fazla e-posta almazsınız. Bunu aşan bildirimler gönderilmez.

 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 4.17
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

1.2.7 Hesap

Hesap sekmesi, temel hesap bilgilerini ayarlamanızı, Weblate üzerinde oturum açmak için çeşitli hizmetleri bağlamanızı, hesabınızı tümüyle silmenizi ya da kullanıcı verilerinizi indirmenizi sağlar (bilgi almak için: [Weblate kullanıcı verilerini dışa aktarma](#)).

Not: Hizmetlerin listesi Weblate yapılandırmanıza bağlıdır. Ancak GitLab, GitHub, Google, Facebook veya Bitbucket ya da diğer OAuth 2.0 hizmeti sağlayıcıları gibi yaygın kullanılan siteleri içerecek şekilde yapılabilir.

 Weblate
 Dashboard Projects Languages Checks
 + ⌵ ⋮

 Your profile
 Languages Preferences Notifications **Account** Profile Teams Licenses Audit log API access

Account

Username

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

Full name






E-mail

You can add another e-mail address below.


Commit e-mail

Save

Current user identities

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

Add new association



E-mail

Removal

Account removal deletes all your private data.

Remove my account

User data

You can download all your private data.

Download user data

1.2.8 Profil

Bu sayfadaki tüm alanlar isteğe bağlıdır ve istediğiniz zaman silinebilir. Bu bilgileri yazarak, bu verileri kullanıcı profilinizin görüldüğü her yerde paylaşmamıza izin vermiş olursunuz.

Sürüm denetimi işlemelerinde hesap e-posta adresiniz yerine özel işleme e-posta adresi kullanılır. Gerçek e-posta adresinizin sızmasını önlemek için bunu kullanın. Farklı e-posta adresi kullanmanın diğer sunucularda yaptığınız katkılarını ilişkilendirilmesini engelleyebileceğini unutmayın (örneğin, katkılarınız artık GitHub profilinizle ilişkilendirilmez). Özel e-posta site genelinde, `PRIVATE_COMMIT_EMAIL_OPT_IN` seçeneği kullanılarak açılabilir.

Avatar her kullanıcı için görüntülenebilir (`ENABLE_AVATARS` seçeneğine bağlı olarak). Bu görseller <https://gravatar.com/> üzerinden alınır.

1.2.9 Lisanslar

1.2.10 API erişimi

API erişim kodunuzu buradan alabilir ya da sıfırlayabilirsiniz.

1.2.11 Denetim günlüğü

Denetim günlüğü, hesabınızla yapılan işlemlerin kaydını tutar. Hesabınızla ilgili her önemli işlem için IP adresini ve kullanılan tarayıcıyı kaydeder. Kritik işlemler ayrıca birincil e-posta adresine bir bildirim gönderilmesini tetikler.

Ayrıca bakınız:

Ters vekil sunucu arkasında çalıştırmak

1.3 Weblate kullanarak çeviri yapmak

Weblate kullanarak çeviri yapmaya ilgi gösterdiğiniz için teşekkür ederiz. Projeler doğrudan çeviri yapılabilecek ya da hesabı olmayan kullanıcılar tarafından yapılan önerileri kabul edecek şekilde ayarlanabilir.

Genel olarak, iki çeviri kipi vardır:

- Proje doğrudan çevirileri kabul eder
- Proje yalnızca belirli bir oy sayısına ulaşıldığında kendiliğinden onaylanan önerileri kabul eder

Çeviri iş akışı hakkında ayrıntılı bilgi almak için [Çeviri iş akışları](#) bölümüne bakabilirsiniz.

Çeviri projesi görünürlüğü seçenekleri:

- Herkes tarafından görülebilir
- Yalnızca belirli bir çevirmen grubu tarafından görülebilir

Ayrıca bakınız:

Erişim denetimi, Çeviri iş akışları

1.3.1 Çeviri projeleri

Çeviri projeleri ilgili bileşenleri içerir. Aynı yazılım, kitap veya proje için kaynaklar.

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate' logo and links to 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, the 'WeblateOrg' project is selected. A 'translated 85%' badge is visible. The main area has tabs for 'Components', 'Languages', 'Info', 'Search', 'Insights', 'Files', 'Tools', 'Manage', and 'Share'. The 'Components' tab is active, showing a table of translation components. The table has columns: Component, Translated, Unfinished, Unfinished words, Checks, Suggestions, and Comments. Three components are listed: 'Android' (79% translated, 30 unfinished words, 3 checks), 'Language names' (95% translated, 4 unfinished words, 5 checks), and 'Glossary' (100% translated, 0 unfinished words, 0 checks). A 'Not watching' button is on the right. At the bottom, there's a button 'Add new translation component' and a footer with 'Powered by Weblate 4.17' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

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

1.3.2 Çeviri bağlantıları

Bir bileşene eriştikten sonra, bir dizi bağlantı ile çevirisine geçilebilir. Çeviri ayrıca *Çevrilmemiş dizgeler* ya da *Tamamlanmamış dizgeler* gibi bölümlere ayrılmıştır. Tüm proje hatasız bir şekilde çevrilmiş olsa bile, *Tüm dizgeler* seçeneği hala kullanılabilir. Alternatif olarak, belirli bir dizgeyi veya terimi bulmak için arama alanı kullanılabilir.

Dashboard
Projects
Languages
Checks

WeblateOrg / Django / Czech
translated 96%

Overview
Info
Search
Insights
Files
Tools
Manage
Share
Watching

Translation status

26	Strings	<div><div></div></div>	96%	Browse Translate
185	Words	<div><div></div></div>	93%	

Strings status

26	All strings — 185 words	Browse	Translate	Zen
25	Translated strings — 173 words	Browse	Translate	Zen
1	Unfinished strings — 12 words	Browse	Translate	Zen
1	Untranslated strings — 12 words	Browse	Translate	Zen
1	Unfinished strings without suggestions — 12 words	Browse	Translate	Zen
3	Strings with any failing checks — 11 words	Browse	Translate	Zen
3	Translated strings with any failing checks — 11 words	Browse	Translate	Zen
1	Failing check: Unchanged translation — 4 words	Browse	Translate	Zen
1	Failing check: Mismatched full stop — 4 words	Browse	Translate	Zen
1	Failing check: Python format — 3 words	Browse	Translate	Zen

Other components

Component	Translated	Unfinished	Unfinished words	Unfinished characters	Checks	Suggestions	Comments
Android MIT	76%	3	3	10			
Language names GPL-3.0	✓						
Glossary WeblateOrg GPL-3.0	✓						
Djangojs GPL-3.0	✓						

[Browse all components](#)

Powered by Weblate 4.17
[About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

1.3.3 Öneriler

Not: Güncel izinler Weblate yapılandırmanıza bağlı olarak değişebilir.

Anonim kullanıcılar yalnızca (varsayılan olarak) öneride bulunabilir. Ayrıca, çeviriyle ilgili belirsizliğin ortaya çıktığı ve diğer çevirmenlerin bunu onaylayacağı durumlarda, oturum açmış kullanıcılar da öneride bulunabilir.

Öneriler, geçerli çeviriyle eşleşen yinelenmeleri ve önerileri kaldırmak için günlük olarak taranır.

1.3.4 Yorumlar

Üç tür yorum yapılabilir: Çeviriler için, kaynak dizgeler için ve *Kaynak onaylama kullanılsın* seçeneği ile etkinleştirildiğinde kaynak dizge hataları için. Tartışmak istediğiniz konuya uygun türü seçin. Kaynak dizge yorumları, her durumda özgün dizge hakkında geri bildirim vermek için iyidir. Örneğin ifadenin düzeltilmesi ya da bağlam hakkında açıklama istenebilir.

Tüm yorumlarda Markdown söz dizimi kullanabilir ve @kullanıcı_adı yaparak diğer kullanıcıları anabilirsiniz.

Ayrıca bakınız:

report-source, *Kaynak dizge onayları*, *Kaynak onaylama kullanılsın*

1.3.5 Çeşitler

Çeşitler, dizgenin farklı uzunluktaki çeşitlerini gruplandırmak için kullanılır. Böylece projenizin ön yüzünde ekran ya da pencere boyutuna bağlı olarak farklı dizgeler kullanılabilir.

Ayrıca bakınız:

variants, *Çeşitler*

1.3.6 Etiketler

Etiketler, yerelleştirme iş akışını daha da özelleştirmek üzere (örneğin dizge grupları tanımlamak için) bir projedeki dizgeleri sınıflamak için kullanılır.

Weblate tarafından şu etiketler kullanılır:

Kendiliğinden çevrilmiş

Dizge *Kendiliğinden çeviri* ile çevrilmiş.

Kaynağın onaylanması gerekli

Dizge *Kaynak dizge onayları* ile onaylanmak üzere imlenmiş.

Ayrıca bakınız:

labels

1.3.7 Çeviri

Çeviri sayfasında, kaynak dizge ve çevirisi için bir düzenleme alanı gösterilir. Çevirinin çoğul olması durumunda, her biri çeviri yapılan dildeki çoğul biçim sayısı kadar açıklanmış ve etiketlenmiş kaynak dizge ve düzenleme alanı görüntülenir.

Tüm özel boşluk karakterlerinin altı kırmızı renkle çizilir ve gri simgelerle görüntülenir. Çevirmeni olası bir biçimlendirme sorununa karşı uyarmak için birden fazla boşluğun altı kırmızı renkle çizilir.

Bu sayfada, çoğu proje kaynak kodundan (bağlam, yorumlar ya da iletinin kullanıldığı yer gibi) gelen çeşitli ek bilgiler görüntülenebilir. Çevirmenlerin ayarlarından seçtikleri ikincil diller için çeviri alanları, kaynak dizgenin üstünde görüntülenir (bilgi almak için: *İkincil diller*).

Çevirmenler, çevirinin altında başkaları tarafından yapılmış, kabul edilecek (✓), değişikliklerle kabul edilecek (≡) ya da silinmiş (🗑️) önerileri görebilir.

Çoğul biçimler

Sayıları hesaba katmak için biçim değiştiren sözcüklere çoğul denir. Her dilin kendi çoğul tanımı vardır. Örneğin İngilizce, bir biçim destekler. Örneğin “araba” tekil tanımında, dolaylı olarak bir arabayı çağırır. Çoğul biçimde “arabalar” iki veya daha fazla arabayı (ya da bir ad olarak araba kavramını) çağırır. Örneğin Çekçe ya da Arapça gibi dillerde daha fazla çoğul biçim vardır ve çoğul biçim kuralları farklıdır.

Weblate, bu biçimlerin her biri için, ilgili her bir dilde (her çoğul biçimi ayrı çevirebilen) tam destek sunar. Alanların sayısı ve çevrilmiş uygulamada ya da projede nasıl kullanıldığı, yapılandırılmış çoğul biçim formülüne bağlıdır. Weblate temel bilgileri gösterir. Unicode Konsorsiyumu tarafından yayınlanmış [Dil çoğul kuralları](#) içinde ayrıntılı bilgi bulabilirsiniz.

Ayrıca bakınız:

Çoğul formülü

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The main content area displays the translation form for the string '%(count)s word'. The form includes fields for English (Singular: '%(count)s word', Plural: '%(count)s words'), Czech (One: '%(count)s slovo', Few: '%(count)s slova', Many: '%(count)s slov'), and a plural formula: '(n==1) ? 0 : (n>2 && n<=4) ? 1 : 2'. The form also has a 'Needs editing' checkbox and buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'.

The right sidebar contains several sections: 'Glossary' (English to Czech), 'String information' (Screenshot context, Explanation, Labels, Flags, Source string location, String age, Source string age, Translation file), and 'Comments' (New comment, Comment on this string, Scope, Is your comment specific to this translation, New comment, Save).

Alternatif çeviriler

4.13 sürümünde geldi.

Not: Şu anda bu özellik yalnızca *Çok değerli CSV dosyası* için destekleniyor.

Bazı biçimlerde, tek bir dizge için birden fazla çeviri bulunabilir. *Araçlar* menüsünü kullanarak daha fazla alternatif çeviri ekleyebilirsiniz. Boş alternatif çeviriler kaydedildikten sonra kendiliğinden kaldırılır.

Klavye kısayolları

Çeviri sırasında şu klavye kısayolları kullanılabilir:

Klavye kısayolu	Açıklama
Alt+Home	Geçerli aramada ilk çeviriye gider.
Alt+End	Geçerli aramada son çeviriye gider.
Alt+PageUp ya da Ctrl+↑ ya da Alt+↑ ya da Cmd+↑	Geçerli aramada önceki çeviriye gider.
Alt+PageDown ya da Ctrl+↓ ya da Alt+↓ ya da Cmd+↓	Geçerli aramada sonraki çeviriye gider.
Ctrl+Enter ya da Cmd+Enter	Geçerli formu gönderir. Bu seçenek, çeviriye düzenlerken <i>Kaydet ve ilerle</i> düğmesine basmakla aynıdır.
Ctrl+Shift+Enter ya da Cmd+Shift+Enter	Düzenlenmesi gerekli imini kaldırır ve çeviriye gönderir.
Alt+Enter ya da Option+Enter	Dizgeyi öneri olarak gönderir. Bu seçenek, çeviriye düzenlerken <i>Suggest</i> düğmesine basmakla aynıdır.
Ctrl+E ya da Cmd+E	İmleci çeviri alanına taşır.
Ctrl+U ya da Cmd+U	İmleci yorum alanına taşır.
Ctrl+M ya da Cmd+M	<i>Kendiliğinden öneriler</i> sekmesini görüntüler. Bilgi almak için: Kendiliğinden öneriler .
Ctrl+1 to Ctrl+9 ya da Cmd+1 to Cmd+9	Kaynak dizgeden belirtilen numaralı yerleştirilebilir ögeyi kopyalar.
Ctrl+M ardından 1 ile 9 arası ya da Cmd+M ardından 1 ile 9 arası	Belirtilen numaralı makine çevirisini geçerli çeviriye kopyalar.
Ctrl+I ardından 1 ile 9 arası ya da Cmd+I ardından 1 ile 9 arası	Denetim hataları listesindeki bir ögeyi yok sayar.
Ctrl+J ya da Cmd+J	<i>Yakındaki dizgeler</i> sekmesini görüntüler.
Ctrl+S ya da Cmd+S	İmleci arama alanına taşır.
Ctrl+O ya da Cmd+O	Kaynak dizgeyi kopyalar.
Ctrl+Y ya da Cmd+Y	<i>Düzenlenmesi gerekli</i> imini değiştirir.

Sanal klavye

Çeviri alanının hemen üstünde küçük bir görsel klavye satırı görüntülenir. Bu özellik, yerel noktalama işaretlerini (satır her dil için yerel olduğundan) ya da başka türlü yazılması zor karakterleri kullanmak için yararlı olabilir.

Görüntülenen simgeler üç gruba ayrılır:

- *Kullanıcı profili* içinde kullanıcı tarafından yapılandırılmış *Özel karakterler*
- Weblate tarafından sağlanan her dile özel karakterler (örneğin tırnak işaretleri ya da sağdan sola yazılan dillere özgü karakterler)
- *SPECIAL_CHARS* kullanılarak yapılandırılmış karakterler

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Hebrew' language. The main area is titled 'Translation' and shows a text input field with the Hebrew word 'קבצים'. Below the input field are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. To the right, there is a 'Glossary' section with a search bar and a 'String information' section showing details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'. At the bottom, there is a table with columns 'Language' and 'Target string' showing translations for Czech, Hungarian, and English.

Çeviri bağlamı

Bu bağlamsal açıklama, geçerli dizge hakkında açıklayıcı bilgiler sağlar.

Dizge öznitelikleri

İleti kimliği, bağlam (`msgctxt`) ya da kaynak kodundaki konum gibi öğeler.

Ekran görüntüleri

Çevirmenleri dizgenin nerede ve nasıl kullanıldığı konusunda daha iyi bilgilendirmek için Weblate üzerine ekran görüntüleri yüklenebilir. Bilgi almak için: *Dizgeler için görsel bağlam*.

Yakındaki dizgeler

Çeviri dosyasındaki komşu iletileri görüntüler. Bunlar genellikle çevrilen dizgeye benzer bağlamda bulunur ve çevirinin tutarlı olmasında yararlıdır.

Diğer kullanımlar

Bir iletinin birden fazla yerde (örneğin birden fazla bileşende) kullanılması durumunda, bunların tutarsız olduğu algılanırsa, bu sekmede tümü görüntülenir (bilgi almak için: [Tutarsız](#)). Kullanılacak çeviriyi buradan seçebilirsiniz.

Çeviri belleği

Geçmişte çevrilmiş benzer dizgelere bakabilirsiniz. Bilgi almak için: [Çeviri belleği](#).

Sözlük

Geçerli iletideki sözcüklerin proje sözlüğündeki karşılıklarını görüntüler.

Son değişiklikler

Weblate üzerinden yakın zamanda bu iletiyi değiştiren kişilerin listesi.

Proje

Çevirmenler için proje bilgileri ya da dizgenin projenin kullandığı sürüm denetimi sistemi deposundaki klasörü ya da bağlantısı.

Doğrudan bağlantılar istiyorsanız, çeviri biçimi bunu desteklemelidir.

Çeviri geçmişi

Her değişiklik varsayılan olarak (bileşen ayarlarından kapatılmadığı sürece) veri tabanına kaydedilir ve geri alınabilir. İsteğe bağlı olarak, temel sürüm denetimi sistemindeki herhangi bir işlemi geri alabilirsiniz.

Çevrilmiş dizgenin uzunluğu

Weblate, çevrilen dizgenin çok uzun olmamasını sağlamak için çevirinin uzunluğunu birkaç şekilde sınırlayabilir:

- Çeviri için varsayılan sınırlama, kaynak dizgeden on kat daha uzun olmasıdır. Bu özellik, `LI-MIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH` seçeneği ile kapatılabilir. Bu sınırlamaya takılıyorsanız, yanlışlıkla iki dilli olarak ayarlanmış tek dilli bir çeviri kullanıyor olabilirsiniz ve Weblate gerçek kaynak dizgenin çeviri anahtarını yanlış anlıyor olabilir. Bilgi almak için [İki dilli ve tek dilli biçimler](#).
- Çeviri dosyasından ya da im ile tanımlanan en fazla karakter uzunluğu. Bilgi almak için: [Çevirinin en fazla uzunluğu](#).
- Piksel cinsinden, imler ile tanımlanan en fazla görüntülenme boyutu. Bilgi almak için: [Çevirinin en fazla uzunluğu](#).

1.3.8 Kendiliğinden öneriler

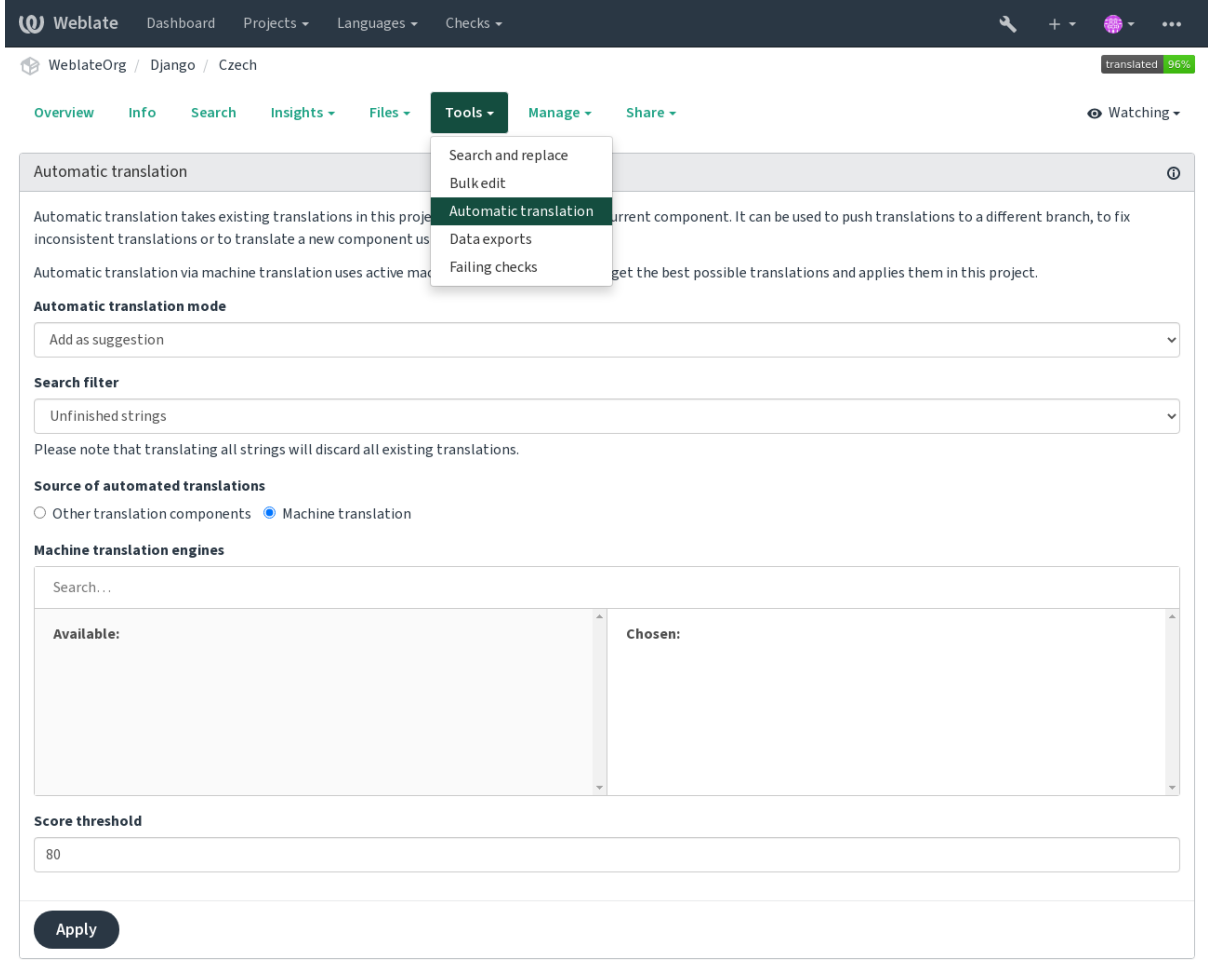
Yapılandırmaya ve çevrilen dilinize bağlı olarak, Weblate çeşitli makine çevirisi hizmetlerinden ve [Çeviri belleği](#) üzerinden öneriler sunar. Tüm makine çevirileri, her çeviri sayfasındaki tek bir sekmede görülebilir.

Ayrıca bakınız:

Desteklenen araçların listesini [Kendiliğinden önerileri yapılandırmak](#) bölümünde bulabilirsiniz.

1.3.9 Kendiliğinden çeviri

Dış kaynakları kullanarak çeviriyi başlatmak için kendiliğinden çeviri özelliğini kullanabilirsiniz. Bu araca *Kendiliğinden çeviri* adı verilmiştir ve bir bileşen ve dil seçtikten sonra *Araçlar* menüsünden erişilebilir:



İki şekilde işlem yapılabilir:

- Çeviriler için diğer Weblate bileşenleri kaynak olarak kullanılabilir.
- Belirli bir kalite eşliğinin üzerindeki çevirilerle seçilen makine çevirisi hizmetleri kullanılabilir.

Hangi dizgelerin kendiliğinden çevrileceğini de seçebilirsiniz.

Uyarı: Bu işlemin, *Tüm dizgeler* gibi geniş kapsamlı süzgeçlerle kullanıldığında var olan çevirilerin üzerine yazacağını unutmayın.

Farklı bileşenler arasında çeviriyi birleştirmek (uygulama ile site gibi) ya da var olan çevirileri (çeviri belleği) kullanarak yeni bir bileşen için çevirileri önceden hazırlamak gibi çeşitli durumlarda kullanışlıdır.

Kendiliğinden çevrilmiş etiketinin görüntülenmesi sorunu düzeltildi.

Ayrıca bakınız:

Bileşenler arasında çevirilerin tutarlılığını sağlamak

1.3.10 Hızı sınırlamak

Arayüzün kötüye kullanılmasını önlemek için, arama yapma, iletişim formları gönderme ya da çeviri gibi çeşitli işlemlere hız sınırlaması uygulanır. Hız sınırlamasına takıldığınızda, işlemi yeniden yapabilmek için belirli bir süre beklemeniz gerekir.

Varsayılan sınırlar ve ince ayarlar yönetim rehberinde açıklanmıştır. Bilgi almak için: [Hızı sınırlamak](#).

1.3.11 Arama ve değiştirme

Terminolojiyi verimli şekilde değiştirin ya da *Araçlar* menüsündeki *Arama ve değiştirme* seçeneğini kullanarak dizgeleri toplu olarak düzeltin.

İpucu: Dizgeleri karıştırma konusunda endişelenmeyin. Bu işlem, asıl değişiklik onaylanmadan önce düzenlenen dizgelerin ön izlemesinin görüntülendiği iki adımda yapılır.

1.3.12 Toplu düzenleme

Toplu düzenleme, belirli sayıda dizge üzerinde tek bir işlem yapılmasına izin verir. Dizgeleri arama ile tanımlarsınız ve eşleşen dizgelere uygulanacak bir değişiklik ayarlarsınız. Şu işlemleri yapabilirsiniz:

- Dizge durumunu değiştirme (örneğin onaylanmamış tüm dizgeler onaylanabilir).
- Çeviri imlerini ayarlama (bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#))
- Dizge etiketlerini ayarlama (bilgi almak için: [labels](#))

İpucu: Bu araca *Toplu düzenleme* adı verilmiştir ve her projenin, bileşenin ya da çevirinin *Araçlar* menüsünden erişilebilir.

Ayrıca bakınız:

[Toplu düzenleme eklentisi](#)

1.3.13 Matris görünümü

Farklı dilleri verimli bir şekilde karşılaştırmak için matris görünümünü kullanabilirsiniz. Her bileşen sayfasında *Araçlar* menüsü altından erişilebilir. Öncelikle karşılaştırmak istediğiniz tüm dilleri seçin ve seçiminizi onaylayın. Ardından hızlı bir şekilde açıp düzenlemek için herhangi bir çeviriye tıklayabilirsiniz.

Matris görünümü, farklı dillerdeki eksik çevirileri bulmak ve bunları tek bir görünümünden hızlı bir şekilde eklemek için de çok iyi bir başlangıç noktasıdır.

1.3.14 Zen kipi

Zen kipine, bir bileşeni çevirirken sağ üstteki *Zen* düğmesine tıklanarak geçilebilir. Çalışma görünümünü basitleştirerek kullanıcı arayüzündeki *Yakındaki dizgeler* ya da *Sözlük* gibi ek öğelerini kaldırır. Çeviri ve onaylama işlemlerinin hızlıca yapılmasını sağlar.

Zen düzenleyiciyi, [Kullanıcı profili](#) içindeki *Ayarlar* sekmesini kullanarak varsayılan düzenleyiciniz olarak ayarlayabilirsiniz. Kişisel tercihinize göre çevirileri *Yukarıdan aşağıya* ya da *Yan yana* görüntülenecek şekilde ayarlayabilirsiniz.

1.4 Çevirileri indirmek ve yüklemek

Bir çevirinin dosyalarını dışa aktarabilir, değişiklikler yapabilir ve bunları yeniden içe aktarabilirsiniz. Böylece, çevrimdışı çalışabilir ve yapılan değişiklikleri var olan çeviriyle yeniden birleştirebilirsiniz. Bu yöntem, bu arada kaynak değiştirilmiş olsa bile işe yarar.

Not: Kullanılabilecek seçenekler *access control* ayarlarıyla sınırlı olabilir.

1.4.1 Çevirileri indirmek

Çevrilebilecek dosyalar proje ya da bileşen panosundaki *Dosyalar* menüsünden indirilebilir.

İlk seçenek, dosyayı depoda bulunan özgün biçiminde indirmektir. Bu durumda, çeviride bekleyen değişiklikler işlenir ve güncel dosya herhangi bir dönüştürme yapılmadan oluşturulur.

Çeviriyi ayrıca, yaygın olarak kullanılan yerelleştirme biçimlerinden birine dönüştürülmüş olarak da indirebilirsiniz. Dönüştürülmüş dosyalar, ek bağlam, yorumlar veya imler gibi Weblate tarafından sağlanan verilerle zenginleştirilir. *Dosyalar* ↓ *İndirmeyi özelleştir* menüsünden çeşitli dosya biçimlerini seçebilirsiniz:

- Gettext PO (`po`)
- Gettext uzantılı XLIFF (`xliff`)
- XLIFF 1.1 (`xliff11`)
- TermBase eXchange (`tbx`)
- Çeviri belleği takası (`tmx`)
- Gettext MO (yalnızca çeviri Gettext PO kullanıyorsa kullanılabilir) (`mo`)
- CSV (`csv`)
- Excel Open XML (`xlsx`)
- JSON (yalnızca tek dilli çeviriler için kullanılabilir) (`json`)
- Android dizge kaynağı (yalnızca tek dilli çeviriler için kullanılabilir) (`aresource`)
- iOS dizgeleri (yalnızca tek dilli çeviriler için kullanılabilir) (`strings`)

İpucu: Dönüştürülen dosyalarda bulunan içerik, dosya biçimi özelliklerine göre farklılık gösterir. *Çeviri türü yetenekleri* içinde özet bilgiler bulabilirsiniz.

Quick downloads

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

Customize download

All strings

File format

☒ gettext PO ☐ XLIFF 1.1 with gettext extensions ☐ XLIFF 1.1 ☐ TBX ☐ TMX ☐ gettext MO ☐ CSV ☐ XLSX

Download

Powered by Weblate 4.17 About Weblate Legal Contact Documentation Donate to Weblate

Ayrıca bakınız:

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

1.4.2 Çevirileri yüklemek

Değişikliklerinizi yaptıktan sonra, *Dosyalar* menüsünden *Çeviriyi yükle* seçeneğini kullanın.

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, a breadcrumb trail shows 'WeblateOrg / Django / Czech'. A status bar on the right indicates 'translated 96%'. The main content area has a 'Files' tab selected, which has opened a dropdown menu with options: 'Download translation', 'Customize download', and 'Upload translation'. The 'Upload' form is visible below the menu. It includes a 'File' section with a 'Choose File' button and 'No file chosen' text. Below this is the 'File upload mode' section with four radio buttons: 'Add as translation', 'Add as suggestion', 'Add as translation needing edit', and 'Replace existing translation file'. The 'Processing of "Needs editing" strings' section has a dropdown menu set to 'Do not import'. The 'Conflict handling' section has a dropdown menu set to 'Change translated strings'. Below these are input fields for 'Author name' (filled with 'Weblate Test') and 'Author e-mail' (filled with 'weblate@example.org'). At the bottom of the form is an 'Upload' button. The footer of the page contains the text 'Powered by Weblate 4.17' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Desteklenen dosya biçimleri

Desteklenen biçimdeki herhangi bir dosya karşıya yüklenebilir. Ancak yine de çeviri için kullanılan ile aynı dosya biçiminin kullanılması önerilir. Yoksa bazı özellikler düzgün çevrilemeyebilir.

Ayrıca bakınız:

Desteklenen dosya biçimleri, Çevirileri indirmek ve yüklemek

İçe aktarma yöntemleri

Çeviri dosyaları yüklenirken şu seçenekler sunulur:

Çeviri olarak eklensin (**translate**)

İçe aktarılan dizgeler, var olan dizgelere çeviri olarak eklenir. En yaygın kullanım örneği ve varsayılan davranış budur.

Yalnızca yüklenen dosyadaki çeviriler kullanılır ve ek içerikler kullanılmaz.

Öneri olarak eklensin (**suggest**)

İçe aktarılan dizgeler öneri olarak eklenir. Yüklenen dizgelerinizin onaylanmasını istediğinizde bunu kullanın.

Yalnızca yüklenen dosyadaki çeviriler kullanılır ve ek içerikler kullanılmaz.

Düzenlenmesi gerekli çeviri olarak eklensin (**fuzzy**)

İçe aktarılan dizgeler, düzenlenmesi gerekli çeviriler olarak eklenir. Bu seçenek, çevirilerin kullanılmasını ve aynı zamanda onaylanmasını istediğinizde yararlı olabilir.

Yalnızca yüklenen dosyadaki çeviriler kullanılır ve ek içerikler kullanılmaz.

Var olan çeviri dosyası değiştirilsin (replace)

Var olan dosya yeni içerikle değiştirilir. Bu seçenek var olan çevirilerin kaybına neden olabileceğinden dikkatli kullanın.

Kaynak dizgeler güncellensin (source)

İki dilli çeviri dosyasındaki kaynak dizgeleri günceller. Bu seçenek, *PO dosyalarını POT ile eşleştirerek güncelleme (msgmerge)* eklentisi gibi işler.

Bu seçenek yalnızca bazı dosya biçimleri için kullanılabilir.

Yeni dizgeler eklensin (add)

Çeviriye yalnızca yeni dizgeleri ekler. Var olan dizgeleri atlar.

Hem yeni dizgeleri eklemek hem de var olan çevirileri güncellemek istiyorsanız, dosyayı *Çeviri olarak eklensin* seçeneği ile ikinci kez yükleyin.

Bu seçenek yalnızca *Dizge yönetimi* açıkken kullanılabilir.

Yüklenen dosyadan yalnızca kaynak, çeviri ve anahtar (bağlam) kullanılır.

Ayrıca bakınız:

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

Çakışmaları işlemek

Yüklenen dizgelerden önceden çevrilmiş olanların nasıl işleneceğini belirler.

Düzenlenmesi gerekli dizgeler

İçe aktarılan dosyadaki düzenlenmesi gereken dizgelerin nasıl işleneceği ile ilgili bir seçenek de vardır. Bu tür dizgeler şu üç yöntemden biriyle işlenebilir: İçe aktarılmasın, Düzenlenmesi gerekli olarak içe aktarılsın ya da Çevrilmiş olarak içe aktarılsın.

Sorumluluğu değiştirmek

Yönetici izinleriyle, yüklenen dosyanın yazarını da belirtebilirsiniz. Dosyayı başka bir şekilde aldıysanız ve asıl yazarın çevirmen hesabını var olan çevirilerle birleştirmek isterseniz yararlı olabilir.

1.5 Sözlük

Her projede, terminolojiyi tutmak için kolay erişilen bir veya daha fazla sözlük bulunabilir. Sözlük, çevirilerin tutarlılığının korunmasını kolaylaştırır.

Her dilin bağımsız olarak yönetilebilen bir sözlüğü vardır. Ancak proje yöneticilerinin ve farklı dillerdeki çevirmenlerin de diller arası tutarlılığı korumalarına yardımcı olan tek bir bileşen biçiminde tutulurlar. Çevirmekte olan dizge içindeki sözcüklere karşılık gelen sözlük terimleri, çeviri düzenleyicisinin yan çubuğunda görüntülenir.

İpucu: Siz etkinleştirmedığınız sürece kalite denetimlerinde sözlük terimleri kullanılmaz. Bilgi almak için *Sözlük ile uyumlu değil*.

1.5.1 Sözlükleri yönetmek

4.5 sürümünde değişti: Sözlükler artık normal çeviri bileşenleridir ve yorum yapmak, uzak bir depoda tutmak veya açıklama eklemek gibi tüm Weblate özellikleri kullanılabilir.

Sözlük olarak kullanılsın ögesini açarak herhangi bir bileşeni sözlük olarak kullanabilirsiniz. Bir proje için birden çok sözlük oluşturabilirsiniz.

Belirli bir projen için boş bir sözlük, projeyle birlikte kendiliğinden oluşturulur. Sözlükler, aynı projenin tüm bileşenleri arasında ve isteğe bağlı olarak ilgili sözlük bileşeninden *Projelerde paylaşım* ile diğer projelerle paylaşılır.

Sözlük bileşeni, Weblate üzerindeki renkli etiket eklenmiş diğer bileşenlere benzer:

The screenshot displays the Weblate web interface for a project named 'WeblateOrg / Glossary WeblateOrg / Czech'. The top navigation bar includes 'Dashboard', 'Projects', 'Languages', and 'Checks'. The main content area is divided into several sections:

- Translation status:** Shows '2 Strings' and '3 Words' both at 100% completion. Buttons for 'Add new glossary term', 'Browse', and 'Translate' are present.
- Strings status:** Shows '2 All strings — 3 words' and '2 Translated strings — 3 words'. Each entry has 'Browse', 'Translate', and 'Zen' buttons.
- Other components:** A table listing components with their translation progress and counts.

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

Powered by Weblate 4.17 About Weblate Legal Contact Documentation Donate to Weblate

Tüm sözlük terimlerine göz atın:

The screenshot shows the 'Browse' view of the Weblate Glossary. It includes navigation controls (back, forward, page 1/1), a search bar, and a table of terms. The table has columns for 'English' and 'Czech'.

English	Czech
machine translation	strojový překlad
project	projekt

Powered by Weblate 4.17 About Weblate Legal Contact Documentation Donate to Weblate

ya da bunları herhangi bir çeviri olarak düzenleyin.

1.5.2 Sözlük terimleri

Sözlük terimleri, normal dizgelerle aynı şekilde çevrilir. Her terim için *Araçlar* menüsünü kullanarak ek özellikleri değiştirebilirsiniz.

The screenshot displays the Weblate web interface for managing a glossary. The main area is titled 'Glossary term' and contains a form for editing a term. The 'English' field contains the word 'project', and the 'Czech' field contains 'projekt'. Below these fields is a checkbox for 'Needs editing' and a text area for 'Explanation'. At the bottom of the form are buttons for 'Save and continue', 'Save and stay', 'Suggest', 'Skip', and 'Tools'. The 'Tools' dropdown menu is open, showing several options: 'Delete string', 'Mark as untranslatable', 'Mark as forbidden translation', 'Mark as terminology', and 'Add variant of this string'. To the right of the main form is a sidebar with two sections: 'Glossary' and 'String information'. The 'Glossary' section shows the current term 'project' and its Czech translation 'projekt'. The 'String information' section shows details about the string, including its age and the translation file it belongs to.

Powered by Weblate 4.17 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Çevrilemez terimler

4.5 sürümünde geldi.

Belirli sözlük terimi çevirilerini, toplu olarak düzenleyerek, imine yazarak ya da *Araçlar* ↓ *Çevrilemez olarak imle* seçeneğini kullanarak salt okunur olarak imlemek, çevrilemeyecekleri anlamına gelir. Bu seçeneği, diğer dillerde değiştirilmemesi gereken marka adları veya diğer terimler için kullanabilirsiniz. Salt okunur terimler sözlük yan çubuğunda görsel olarak vurgulanır.

Ayrıca bakınız:

Parametreleri kullanarak davranışı özelleştirmek

Yasaklı çeviriler

4.5 sürümünde geldi.

Belirli sözlük terimi çevirilerini, toplu olarak düzenleyerek, imine yazarak ya da *Araçlar* ↓ *Yasak çeviri olarak imle* seçeneğini kullanarak yasak olarak imlemek, **kullanılmayacakları** anlamına gelir. Bu seçeneği, anlamı belirsiz ya da beklenmedik olan sözcükleri açıklığa kavuşturmak için kullanabilirsiniz.

Ayrıca bakınız:

Parametreleri kullanarak davranışı özelleştirmek

Terminoloji

4.5 sürümünde geldi.

Belirli sözlük terimlerini, toplu olarak düzenleyerek, imine yazarak ya da *Araçlar* ↓ *Terminoloji olarak imle* seçeneğini kullanarak terminoloji olarak imlemek, kayıtların tüm dillerdeki sözlüklere ekleneceği anlamına gelir. Bu seçeneği, tüm dillerde anlamının tutarlı olması ve iyi düşünülmesi gereken önemli terimler için kullanabilirsiniz.

Ayrıca bakınız:

Parametreleri kullanarak davranışı özelleştirmek

Çeşitler

Çeşitler, dizgeleri birlikte gruplandırmak için genel bir yol sağlar. Tüm terim çeşitleri, çeviri sırasında sözlük kenar çubuğunda listelenir.

İpucu: Bir terim için kısaltmalar ya da daha kısa ifadeler eklemek için bunu kullanabilirsiniz.

Ayrıca bakınız:

variants

1.6 Denetimler ve düzeltmeler

Kalite denetimleri, yaygın çevirmen hatalarını yakalamaya yardımcı olarak çevirinin iyi durumda olmasını sağlar. Yanlış pozitif sonuçlarda denetimler yok sayılabilir.

Denetimi başarısız olan bir çeviri gönderildikten hemen sonra kullanıcıya görüntülenir:

1.6.1 Kendiliğinden düzeltmeler

Kalite denetimleri yanında, Weblate çevrilmiş dizgelerdeki bazı yaygın hataları kendiliğinden düzeltebilir. Hatalı çeviriler eklememesi için dikkatli kullanın.

Ayrıca bakınız:

AUTOFIX_LIST

Sondaki üç nokta değiştirici

Kaynak dizge ile tutarlı olması için sondaki üç ayrı noktayı (. . .) bir üç nokta karakteri (...) ile değiştirir.

Sıfır genişlikli boşluk kaldırma

Çeviride genellikle sıfır genişlikli boşluklar istenmez. Bu düzeltme, kaynak dizgede de bulunmadığı sürece bunları kaldırır.

Denetim karakterlerini kaldırma

Çevirideki tüm denetim karakterlerini kaldırır.

Devanagari danda

Devanagari'deki yanlış cümle sonunu Devanagari danda (॥) ile değiştirir.

Güvenli olmayan HTML temizleme

“Güvenli html” imi ile açıldığında, HTML markup kodunu temizler.

Ayrıca bakınız:

Güvenli olmayan HTML

Başta ve sonraki beyaz boşlukları düzeltme

Başta ve sondaki boşlukları kaynak dizgeyle tutarlı hale getirir. Dizgenin işlenecek bölümlerini atlamak için `ignore-begin-space` ve `ignore-end-space` imleri ile ince ayar yapılabilir.

1.6.2 Kalite denetimleri

Weblate, dizgeler üzerinde çok çeşitli kalite denetimleri yapar. Aşağıdaki bölümde bunlar daha ayrıntılı olarak açıklanmaktadır. Ayrıca dile özgü denetimler de vardır. Hatalı çalışan bir denetim varsa lütfen bir hata bildiriminde bulunun.

Ayrıca bakınız:

CHECK_LIST, *Parametreleri kullanarak davranışı özelleştirmek*

1.6.3 Çeviri denetimleri

Her çeviri değişikliğinde yapılır ve çevirmenlerin çeviri kalitesini korumasına yardımcı olur.

BBCode kodlaması

Özet

Çevirideki BBCode kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.BBCodeCheck`

Belirteci denetle

`bbcode`

Yok sayılacak im

`ignore-bbcode`

BBCode, bir iletinin önemli bölümlerini koyu veya italik olarak vurgulamak gibi basit kod imlerini ele alır.

Bu denetim, kaynaktaki kod imlerinin çeviride de bulunmasını sağlar.

Not: BBCode algılama yöntemi şu anda oldukça basittir. Bu nedenle bu denetim yanlış pozitif sonuçlar verebilir.

Art arda yinelenen sözcükler var

4.1 sürümünde geldi.

Özet

Metinde aynı sözcük üst üste iki kez geçiyor:

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.duplicate.DuplicateCheck`

Belirteci denetle

`duplicate`

Yok sayılacak im

`ignore-duplicate`

Bir çeviride ardışık yinelenen sözcüklerin olup olmadığını denetler. Bu durum genellikle çeviride bir hata olduğunu gösterir.

İpucu: Bu denetim, yanlış pozitif sonuçları önlemek için dile özgü kuralları kullanır. Sizin durumunuzda yanlış bir şekilde tetiklenirse, bize bildirin. Bilgi almak için: [Weblate sorunlarını bildirmek](#).

Sözlük ile uyumlu değil

4.5 sürümünde geldi.

Özet

Çeviri, bir sözlükte tanımlanmış terimlerle uyumlu değil.

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.glossary.GlossaryCheck`

Belirteci denetle

`check_glossary`

Etkinleştirilecek im

`check-glossary`

Yok sayılacak im

`ignore-check-glossary`

Bu denetimin `check-glossary` imi ile açılması gerekir (bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)). Lütfen etkinleştirmeden önce şunları göz önüne alın:

- Dizgenin çeviri ile tam olarak eşleşip eşlemediğine bakar. Sözlükte tüm çeşitlerdeki terimlerin bulunması beklenir.
- Her dizgeyi sözlük ile karşılaştırarak denetlemek zorlayıcı olur. Weblate üzerinde dizgeleri içe aktarma ya da çevirme gibi denetimlerin çalıştırılması gereken tüm işlemleri yavaşlatır.
- Ayrıca *Çeviri değiştirilmemiş* içinde çevrilemez sözlük terimlerini de değerlendirir.

Ayrıca bakınız:

Sözlük, Parametreleri kullanarak davranışı özelleştirmek, Çeviri imleri

Çift boşluk

Özet

Çeviride çift boşluk var

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.DoubleSpaceCheck`

Belirteci denetle

`double_space`

Yok sayılacak im

`ignore-double-space`

Alanla ilgili diğer denetimlerde yanlış pozitif sonuçları önlemek için çeviride çift boşluk olup olmadığını denetler.

Kaynakta çift boşluk bulunduğunda bunun kasıtlı olduğu varsayılarak denetim yok sayılır.

Biçimlendirilmiş dizgeler

Kaynak dizgelerdeki biçimlendirmenin çeviride de olup olmadığını denetler. Çeviride biçim dizelerinin atlanması genellikle ciddi sorunlara neden olur. Bu nedenle dizgelerdeki biçimlendirme genellikle kaynakla aynı olmalıdır.

Weblate, biçim dizgelerinin çeşitli dillerde denetlenmesini destekler. Denetim, yalnızca bir dizge uygun şekilde imlenirse (C biçimi için *c-format* gibi) kendiliğinden etkinleştirilmez. Gettext bunu kendiliğinden ekler. Ancak büyük olasılıkla diğer dosya biçimleri için ya da PO dosyalarınız **xgettext** tarafından oluşturulmamışsa el ile eklemeniz gerekir.

Bu, her birim içinde (bilgi almak için [Kaynak dizgeler ile ilgili ek bilgiler](#)) ya da [Bileşen yapılandırması](#) içinde yapılabilir. Her bileşen için tanımlanması daha basittir, ancak dizgenin bir biçimlendirme dizgesi olarak yorumlanmaması durumunda yanlış pozitif sonuçlara yol açabilir. Ancak biçim dizgesi söz dizimi kullanılır.

İpucu: Weblate üzerinde belirli bir biçim denetimi yoksa, genel [Yer belirticiler](#) seçeneğini kullanabilirsiniz.

Denetimin yanında, bu seçenek çeviri dizgelerine kolayca eklemek için biçimlendirme dizgelerini vurgular:

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 '%(count)s word' from English to Czech. The interface includes a search bar, a main translation area with input fields for English, Czech (One, Few, Many), and a plural formula. A sidebar on the right shows string information like context, explanation, labels, flags, and source string location. A bottom section shows nearby strings and a history of changes.

AngularJS değer ekleme dizgesi

Özet

AngularJS değer ekleme dizgeleri kaynak ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.angularjs.AngularJSInterpolationCheck`

Belirteci denetle

`angularjs_format`

Etkinleştirilecek im

`angularjs-format`

Yok sayılacak im`ignore-angularjs-format`**Adlandırılmış biçim dizgesi örneği**`Hesap bakiyeniz {{amount}} {{ currency }}`**Ayrıca bakınız:**[Biçimlendirilmiş dizgeler](#), [AngularJS](#) metin değer ekleme**C biçimi****Özet**

C biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı`weblate.checks.format.CFormatCheck`**Belirteci denetle**`c_format`**Etkinleştirilecek im**`c-format`**Yok sayılacak im**`ignore-c-format`**Basit biçim dizgesi örneği**`Sepette %d elma var`**Konum biçim dizgesi örneği**`Hesap bakiyeniz %1$d %2$s`**Ayrıca bakınız:**[Biçimlendirilmiş dizgeler](#),
[C biçim dizgeleri](#), [C printf biçimi](#)**C# biçimi****Özet**

C# biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı`weblate.checks.format.CSharpFormatCheck`**Belirteci denetle**`c_sharp_format`**Etkinleştirilecek im**`c-sharp-format`**Yok sayılacak im**`ignore-c-sharp-format`**Konum biçim dizgesi örneği**`Sepette {0} elma var`

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [C# dizge biçimi](#)

ECMAScript kalıbı sabitleri

Özet

ECMAScript kalıbı sabitleri kaynakla aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.ESTemplateLiteralsCheck`

Belirteci denetle

`es_format`

Etkinleştirilecek im

`es-format`

Yok sayılacak im

`ignore-es-format`

Değer ekleme örneği

Sepette \${number} elma var

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Kalıp sabitleri](#)

i18next değer eklemesi

4.0 sürümünde geldi.

Özet

I18next değer eklemesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.I18NextInterpolationCheck`

Belirteci denetle

`i18next_interpolation`

Etkinleştirilecek im

`i18next-interpolation`

Yok sayılacak im

`ignore-i18next-interpolation`

Değer ekleme örneği

Sepette {{number}} elma var

İç içe geçme örneği

Sepette \$t(number) elma var

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [i18next değer eklemesi](#)

ICU MessageFormat

4.9 sürümünde geldi.

Özet

ICU MessageFormat dizgelerinde söz dizimi hataları ve/veya yer belirtici uyumsuzlukları var.

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.icu.ICUMessageFormatCheck`

Belirtici denetle

`icu_message_format`

Etkinleştirilecek im

`icu-message-format`

Yok sayılacak im

`ignore-icu-message-format`

Değer ekleme örneği

Sepette `{number, plural, one {bir elma} other {# elma var}}`.

Bu denetim, hem saf ICU MessageFormat iletilerini hem de basit XML kod imleri olan ICU biçimini destekler. XML desteğini seçerek ya da belirli alt denetimleri devre dışı bırakarak `icu-flags:*` seçeneği ile bu denetimin davranışını yapılandırabilirsiniz. Örneğin, aşağıdaki im çoğul alt iletilerin doğrulanmasını devre dışı bırakırken XML desteğini etkinleştirir:

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

<code>xml</code>	Basit XML kod imleri desteğini etkinleştirir. Varsayılan olarak, XML kod imleri gevşek bir şekilde işlenir. Bir kod iminin mantıklı bir parçası olmadığında başıboş <code><</code> karakterleri yok sayılır.
<code>strict-xml</code>	Katı XML kod imleri desteğini etkinleştirir. Bir kod iminin parçası değilse tüm <code><</code> karakterlerinden kaçınılmalıdır.
<code>-highlight</code>	Düzenleyicide yer belirtici öğelerin vurgulanmasını kapatır.
<code>-require_other</code>	Alt iletilerde <code>other</code> seçicisi olması zorunluluğunu devre dışı bırakır.
<code>-submessage</code>	Alt ileti seçicilerin kaynakla eşleşip eşleşmediği denetimini atlar.
<code>-types</code>	Yer belirtici türlerinin kaynak ile uygunluğu denetimini atlar.
<code>-extra</code>	Kaynak dizgede bulunmayan bir yer belirticinin çeviride bulunma denetimini atlar.
<code>-missing</code>	Kaynak dizgede bulunan bir yer belirticinin çeviride bulunmama denetimini atlar.

Ayrıca, `strict-xml` etkinleştirilmemiş ancak `xml` etkinleştirilmiş olduğunda, tüm XML kod imlerinin belirli bir dizgeyle başlamasını zorunlu kılmak için `icu-tag-prefix:PREFIX` imini kullanabilirsiniz. Örneğin, aşağıdaki im, yalnızca `<x:` ile başlayan XML kod imlerinin eşleştirilmesini sağlar:

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

Bu, `<x:link>buraya tıklayın</x:link>` ile eşleşir, ama `bu` ile eşleşmez.

Ayrıca bakınız:

ICU MessageFormat söz dizimi, *Biçimlendirilmiş dizgeler*, *ICU: İletileri biçimlendirmek*, *Format.JS: İleti söz dizimi*

Java biçimi

Özet

Java biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.JavaFormatCheck`

Belirteci denetle

`java_printf_format`

Etkinleştirilecek im

`java-printf-format`

Yok sayılacak im

`ignore-java-printf-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Konum biçim dizgesi örneği

Hesap bakiyeniz %1\$d %2\$s

4.14 sürümünde değişti: Bu eskiden `java-format` imi ile değiştirilirdi. GNU gettext ile tutarlılık için değiştirildi.

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Java biçim dizgesi](#)

Java MessageFormat

Özet

Java MessageFormat dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.JavaMessageFormatCheck`

Belirteci denetle

`java_format`

Koşulsuz olarak etkinleştirilecek im

`java-format`

Kendiliğinden algılamayı etkinleştirecek im

`auto-java-messageformat`, yalnızca kaynakta bir biçim dizgesi varsa denetimi etkinleştirir

Yok sayılacak im

`ignore-java-format`

Konum biçim dizgesi örneği

Sepette {0} elma var

4.14 sürümünde değişti: Bu eskiden `java-messageformat` imi ile değiştiriliyordu. GNU gettext ile tutarlılık için değiştirildi.

Bu denetim, biçim dizgesinin Java MessageFormat sınıfı için geçerli olduğunu doğrular. Yaylı parantezlerdeki biçim dizgelerini eşleştirmenin yanında, özel bir anlamı olduğu için tek tırnakları da doğrular. Tek tırnak ”” şeklinde yazılmalıdır. Eşleştirilmediğinde, alıntılamanın başlangıcı olarak kabul edilir ve dizge işlenirken gösterilmez.

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, Java Message biçimi

JavaScript biçimi**Özet**

JavaScript biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.JavaScriptFormatCheck`

Belirteci denetle

`javascript_format`

Etkinleştirilecek im

`javascript-format`

Yok sayılacak im

`ignore-javascript-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, JavaScript dizge biçimi

Lua biçimi**Özet**

Lua biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.LuaFormatCheck`

Belirteci denetle

`lua_format`

Etkinleştirilecek im

`lua-format`

Yok sayılacak im

`ignore-lua-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, Lua dizge biçimi

Object Pascal biçimi

Özet

Object Pascal biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.ObjectPascalFormatCheck`

Belirteci denetle

`object_pascal_format`

Etkinleştirilecek im

`object-pascal-format`

Yok sayılacak im

`ignore-object-pascal-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, Object Pascal biçimlendirme dizgeleri, Free Pascal biçimlendirme dizgeleri Delphi biçimlendirme dizgeleri

Yüzde yer belirtecileri

4.0 sürümünde geldi.

Özet

Yüzde yer belirtecileri kaynakla aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.PercentPlaceholdersCheck`

Belirteci denetle

`percent_placeholders`

Etkinleştirilecek im

`percent-placeholders`

Yok sayılacak im

`ignore-percent-placeholders`

Basit biçim dizgesi örneği

Burada %number% elma var

Ayrıca bakınız:

Biçimlendirilmiş dizgeler,

Perl biçimi

Özet

Perl biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.PperlFormatCheck`

Belirteci denetle

`perl_format`

Etkinleştirilecek im

`perl-format`

Yok sayılacak im

`ignore-perl-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Konum biçim dizgesi örneği

Hesap bakiyeniz %1\$d %2\$s

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, *Perl sprintf*, *Perl biçim dizgeleri*

PHP biçimi

Özet

PHP biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.PHPFormatCheck`

Belirteci denetle

`php_format`

Etkinleştirilecek im

`php-format`

Yok sayılacak im

`ignore-php-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Konum biçim dizgesi örneği

Hesap bakiyeniz %1\$d %2\$s

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, *PHP sprintf* belgeleri, *PHP biçim dizgeleri*

Python parantez biçimi

Özet

Python ayracı biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.PythonBraceFormatCheck`

Belirteci denetle

`python_brace_format`

Etkinleştirilecek im

`python-brace-format`

Yok sayılacak im

`ignore-python-brace-format`

Basit biçim dizgesi

Burada {} elma var

Adlandırılmış biçim dizgesi örneği

Hesabınızda {amount} {currency} var

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Python parantez biçimi](#), [Python biçim dizgeleri](#)

Python biçimi

Özet

Python biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.PythonFormatCheck`

Belirteci denetle

`python_format`

Etkinleştirilecek im

`python-format`

Yok sayılacak im

`ignore-python-format`

Basit biçim dizgesi

Sepette %d elma var

Adlandırılmış biçim dizgesi örneği

Hesabınızda %(amount)d %(currency)s var

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Python dizge biçimlendirmeleri](#), [Python biçim dizgeleri](#)

Qt biçimi

Özet

Qt biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.qt.QtFormatCheck`

Belirteci denetle

`qt_format`

Etkinleştirilecek im

`qt-format`

Yok sayılacak im

`ignore-qt-format`

Konum biçim dizgesi örneği

Burada %1 elma var

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, `Qt QString::arg()`

Qt çoğul biçimi

Özet

Qt çoğul biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.qt.QtPluralCheck`

Belirteci denetle

`qt_plural_format`

Etkinleştirilecek im

`qt-plural-format`

Yok sayılacak im

`ignore-qt-plural-format`

Çoğul biçim dizgesi örneği

Burada %Ln elma(lar) var

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, `Qt i18n rehberi`

Ruby biçimi

Özet

Ruby biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.ruby.RubyFormatCheck`

Belirteci denetle

`ruby_format`

Etkinleştirilecek im

`ruby-format`

Yok sayılacak im

`ignore-ruby-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Konum biçim dizgesi örneği

Hesabınızda %1\$f %2\$s var

Adlandırılmış biçim dizgesi örneği

Hesabınızda %+.2<amount>f %<currency>s var

Adlandırılmış kalıp dizgesi

Hesabınızda %{amount} %{currency} var

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Ruby Kernel#sprintf](#)

Şema biçimi

Özet

Şema biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.SchemeFormatCheck`

Belirteci denetle

`scheme_format`

Etkinleştirilecek im

`scheme-format`

Yok sayılacak im

`ignore-scheme-format`

Basit biçim dizgesi örneği

Burada ~d elma var

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Srfi 28](#), [Chicken şeması biçimi](#), [Guile şeması biçiminde çıktı](#)

Vue I18n biçimlendirmesi

Özet

Vue I18n biçimlendirmesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.VueFormattingCheck`

Belirteci denetle

`vue_format`

Etkinleştirilecek im

`vue-format`

Yok sayılacak im

`ignore-vue-format`

Adlandırılmış biçimlendirme

Burada `{count}` elma var

Rails i18n biçimlendirmesi

Burada `%{count}` elma var

Bağlantılı yerel iletiler

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

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, *Vue I18n biçimi*, *Vue I18n bağlantılı yerel iletiler*

Çevrilmiş

Özet

Bu dizge geçmişte çevrilmiş

Kapsam

tüm dizgeler

Denetim sınıfı

`weblate.checks.consistency.TranslatedCheck`

Belirteci denetle

`translated`

Yok sayılacak im

`ignore-translated`

Bir dizgenin zaten çevrilmiş olduğu anlamına gelir. Bu durum, çeviriler sürüm denetimi sisteminden geri alındığında ya da başka bir şekilde kaybolduğunda ortaya çıkabilir.

Tutarsız

Özet

Bu dizge bu projede farklı şekillerde çevrilmiş ya da bazı bileşenlerde çevrilmemiş.

Kapsam

tüm dizgeler

Denetim sınıfı

`weblate.checks.consistency.ConsistencyCheck`

Belirteci denetle

`inconsistent`

Yok sayılacak im

`ignore-inconsistent`

Weblate, çeviri tutarlılığını korumanıza yardımcı olmak için bir projedeki tüm çevirilerde aynı dizgenin çevirilerini denetler.

Bir proje içindeki bir dizgenin çevirileri farklıysa hata ortaya çıkar. Bu durum, görüntülenen denetimlerde de tutarsızlıklara yol açabilir. Bu dizgenin diğer çevirilerini *Diğer kullanımlar* sekmesinde bulabilirsiniz.

Bu denetim, bir projede *Çevirilerin yayılmasını sağlamak* seçeneği açık olan tüm bileşenlere uygulanır.

İpucu: Başarımı olumsuz etkileyeceğinden, denetim eşleşme sayısını sınırlar ve tüm tutarsızlıkları bulamayabilir.

Not: Bu denetim, dizgenin bir bileşene çevrilip başka bir bileşene çevrilmemesi durumunda da tetiklenir. Bazı bileşenlerde çevrilmemiş dizgeleri el ile hızlı bir şekilde çevirebilmek için, *Diğer kullanımlar* sekmesindeki her satırda görüntülenen *Bu çeviri kullanılsın* düğmesine tıklanabilir.

Yeni eklenen ancak başka bir bileşende zaten çevrilmiş olan dizgelerin kendiliğinden çevrilmesi için *Kendiliğinden çeviri* eklentisini kullanabilirsiniz.

Ayrıca bakınız:

Bileşenler arasında çevirilerin tutarlılığını sağlamak

Kashida harfi kullanılmış

3.5 sürümünde geldi.

Özet

Dekoratif kashida harfleri kullanılmamalıdır

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.KashidaCheck`

Belirteci denetle

`kashida`

Yok sayılacak im

`ignore-kashida`

Dekoratif Kashida harfleri çeviride kullanılmamalıdır. Bunlar Tatweel olarak da bilinir.

Ayrıca bakınız:

[Vikipedi üzerinde Kashida](#)

Markdown bağlantıları

3.5 sürümünde geldi.

Özet

Markdown bağlantıları kaynakla aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.MarkdownLinkCheck`

Belirteci denetle

`md-link`

Etkinleştirilecek im

`md-text`

Yok sayılacak im

`ignore-md-link`

Markdown bağlantıları kaynak ile aynı değil.

Ayrıca bakınız:

[Markdown bağlantıları](#)

Markdown başvuruları

3.5 sürümünde geldi.

Özet

Markdown bağlantı başvuruları kaynakla aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.MarkdownRefLinkCheck`

Belirteci denetle

`md-reflink`

Etkinleştirilecek im

`md-text`

Yok sayılacak im

`ignore-md-reflink`

Markdown bağlantı başvuruları kaynakla aynı değil.

Ayrıca bakınız:

[Markdown bağlantıları](#)

Markdown söz dizimi

3.5 sürümünde geldi.

Özet

Markdown söz dizimi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.MarkdownSyntaxCheck`

Belirteci denetle

`md-syntax`

Etkinleştirilecek im

`md-text`

Yok sayılacak im

`ignore-md-syntax`

Markdown söz dizimi kaynaktaki ile aynı değil

Ayrıca bakınız:

[Markdown span bileşenleri](#)

Çevirinin en fazla uzunluğu

Özet

Çeviri belirtilen uzunluğu aşmamalı

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.MaxLengthCheck`

Belirteci denetle

`max-length`

Etkinleştirilecek im

`max-length`

Yok sayılacak im

`ignore-max-length`

Çevirilerin kullanılacak alana sığacak şekilde kabul edilebilir uzunlukta olup olmadığını denetler. Bu denetim yalnızca çeviri karakterlerinin uzunluğuna bakar.

Diğer denetimlerden farklı olarak, im `max-length:100` gibi bir `anahtar:değer` çifti olarak ayarlanmalıdır.

İpucu: Bu denetim, metni işlemek için orantılı yazı tipleri kullanırken en iyi ölçünün ne olmayabileceği hakkında karakter sayısına bakar. *Çevirinin en fazla uzunluğu* denetimi, metnin gerçek görünümünü denetler.

`replacements:` imi, dizgeyi denetlemeden önce yerleştirilebilir öğeleri genişletmek için de yararlı olabilir.

`xml-text` imi de kullanıldığında, uzunluk hesaplamasında XML kod imleri yok sayılır.

Çevirinin en fazla uzunluğu

Özet

Çevrilmiş metin belirtilen boyutu aşmamalıdır

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.render.MaxSizeCheck`

Belirteci denetle

`max-size`

Etkinleştirilecek im

`max-size`

Yok sayılacak im

`ignore-max-size`

3.7 sürümünde geldi.

Çeviri yapılan metin belirtilen boyuttan uzun olmamalıdır. Metni satır kaydırma ile oluşturur ve belirtilen sınırlara uyup uymadığını denetler.

Bu denetim için bir ya da iki parametre gereklidir. En fazla genişlik ve en fazla satır sayısı. Satır sayısının belirtilmesi durumunda metin bir satır olarak kabul edilir.

Kullanılan yazı tipini `font-*` komutlarıyla da yapılandırabilirsiniz (bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)). Örneğin aşağıdaki çeviri imleri, ubuntu yazı tipi boyutu 22 ile oluşturulan metnin iki satıra ve 500 piksele sığması gerektiğini belirtir:

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

İpucu: Bir bileşendeki tüm dizgelerde aynı yazı tipini yapılandırmak için [Bileşen yapılandırması](#) içindeki `font-*` komutlarını ayarlamak isteyebilirsiniz. Her dizgeyi ayrı özelleştirmeniz gerekirse bu değerleri her dizgeye özel olarak değiştirebilirsiniz.

`replacements:` imi, dizgeyi denetlemeden önce yerleştirilebilir öğeleri genişletmek için de yararlı olabilir.

`xml-text` imi de kullanıldığında, uzunluk hesaplamasında XML kod imleri yok sayılır.

Ayrıca bakınız:

[Yazı tiplerini yönetmek](#), [Parametreleri kullanarak davranışı özelleştirmek](#), [Çevirinin en fazla uzunluğu](#)

\n aynı değil

Özet

Çevirideki \n sabit yeni satır sayısı kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EscapedNewlineCountingCheck`

Belirteci denetle

`escaped_newline`

Yok sayılacak im

`ignore-escaped-newline`

Kaçışlı yeni satır kodları genellikle program çıktısını biçimlendirmek için önemlidir. Çevirideki `\n` kodlarının sayısı kaynakla aynı değilse denetim hatası oluşur.

İki nokta üst üste farklı

Özet

Kaynağın ve çevirinin ikisi de iki nokta üst üste ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndColonCheck`

Belirteci denetle

`end_colon`

Yok sayılacak im

`ignore-end-colon`

Kaynaktaki iki nokta üst üste işaretlerinin çeviri ile aynı olup olmadığını denetler. İki nokta üst üste işaretlerinin varlığı, kullanılmadıkları çeşitli dillerde de denetlenir (Çince ya da Japonca gibi).

Ayrıca bakınız:

[Vikipedi üzerinde iki nokta üst üste](#)

Üç nokta karakteri farklı

Özet

Kaynağın ve çevirinin ikisi de üç nokta karakteri ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndEllipsisCheck`

Belirteci denetle

`end_ellipsis`

Yok sayılacak im

`ignore-end-ellipsis`

Kaynağın sonundaki üç nokta karakterinin çeviri ile aynı olup olmadığını denetler. Bu denetim yalnızca gerçek üç nokta karakterine (...) bakar. Üç ayrı nokta karakterini (. . .) denetlemez.

Üç nokta karakteri çıktıda genellikle üç ayrı nokta karakterinden daha güzel görüntülenir ve metinden konuşmaya dönüştürme uygulamalarında daha iyi bir ses çıkarır.

Ayrıca bakınız:

[Vikipedi üzerinde üç nokta karakteri](#)

Ünlem işareti farklı

Özet

Kaynağın ve çevirinin ikisi de ünlem işareti ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndExclamationCheck`

Belirteci denetle

`end_exclamation`

Yok sayılacak im

`ignore-end-exclamation`

Kaynaktaki ünlem işaretlerinin çeviri ile aynı olup olmadığını denetler. Ünlem işaretlerinin varlığı, kullanılmadıkları çeşitli diller için de denetlenir (Çince, Japonca, Korece, Ermenice, Limbu, Myanmar ya da Nko).

Ayrıca bakınız:

[Vikipedi üzerinde ünlem işareti](#)

Nokta işareti farklı

Özet

Kaynağın ve çevirinin ikisi de nokta ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndStopCheck`

Belirteci denetle

`end_stop`

Yok sayılacak im

`ignore-end-stop`

Kaynaktaki nokta işaretlerinin çeviri ile aynı olup olmadığını denetler. Nokta işaretlerinin varlığı, kullanılmadıkları çeşitli diller için de denetlenir (Çince, Japonca, Devanagari ya da Urduca).

Ayrıca bakınız:

[Vikipedi üzerinde nokta işareti](#)

Soru işareti farklı

Özet

Kaynağın ve çevirinin ikisi de soru işareti ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndQuestionCheck`

Belirteci denetle

`end_question`

Yok sayılacak im

`ignore-end-question`

Kaynaktaki soru işaretlerinin çeviri ile aynı olup olmadığını denetler. Soru işaretlerinin varlığı, kullanılmadıkları çeşitli diller için de denetlenir (Ermenice, Arapça, Çince, Korece, Japonca, Etiyopyaca, Vai ya da Kıpti).

Ayrıca bakınız:

[Vikipedi üzerinde soru işareti](#)

Noktalı virgül farklı

Özet

Kaynağın ve çevirinin ikisi de noktalı virgül ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndSemicolonCheck`

Belirteci denetle

`end_semicolon`

Yok sayılacak im

`ignore-end-semicolon`

Kaynak cümlelerin sonundaki noktalı virgül işaretlerinin çeviri ile aynı olup olmadığını denetler.

Ayrıca bakınız:

[Vikipedi üzerinde noktalı virgül](#)

Satır sonları farklı

Özet

Çevirideki yeni satır sayısı kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.NewLineCountCheck`

Belirteci denetle

`newline-count`

Yok sayılacak im

`ignore-newline-count`

Yeni satırlar genellikle program çıktısını biçimlendirmek için önemlidir. Çevirideki yeni satır sayısı kaynakla aynı değilse denetim hatası oluşur.

Çoğul biçim eksik

Özet

Bazı çoğul biçimler çevrilmemiş

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.consistency.PluralsCheck`

Belirteci denetle

`plurals`

Yok sayılacak im

ignore-plurals

Bir kaynak dizgenin tüm çoğul biçimlerinin çevrilip çevrilmediğini denetler. Her çoğul biçimin nasıl kullanıldığının ayrıntıları dizge tanımında bulunabilir.

Çoğul biçimlerin doldurulmaması, bazı durumlarda çoğul biçim kullanımdayken hiçbir şeyin görüntülenmemesine yol açar.

Yer belirticiler

3.9 sürümünde geldi.

Özet

Çeviride bazı yer belirticiler eksik

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

weblate.checks.placeholders.PlaceholderCheck

Belirteci denetle

placeholders

Etkinleştirilecek im

placeholders

Yok sayılacak im

ignore-placeholders

4.3 sürümünde değişti: Kurallı ifadeyi yer belirtici olarak kullanabilirsiniz.

4.13 sürümünde değişti: `case-insensitive` imi kullanıldığında, yer belirticiler büyük/küçük harfe duyarsız olarak işlenir.

Çeviride bazı yer belirticiler eksik. Bunlar ya çeviri dosyasından çıkarılır ya da el ile `placeholders` imi kullanılarak tanımlanır. Fazlası iki nokta üst üste işaretiyle ayrılabilir, boşluk olan dizgeler alıntılanabilir:

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

Yer belirticiler için belirli söz dizimlerin varsa, bir kurallı ifade kullanabilirsiniz:

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

Büyük/küçük harfe duyarlı olmayan yer belirticileriniz de olabilir:

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

Ayrıca bakınız:

Parametreleri kullanarak davranışı özelleştirmek

Noktalama aralığı

3.9 sürümünde geldi.

Özet

Çift noktalama işaretinin önündeki birleşik boşluk eksik

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.PunctuationSpacingCheck`

Belirteci denetle

`punctuation_spacing`

Yok sayılacak im

`ignore-punctuation-spacing`

Çift noktalama işaretinden önce bölünemeyen boşluk olup olmadığını denetler (ünlem işareti, soru işareti, noktalı virgül ve iki nokta). Bu kural yalnızca Fransızca ya da Breton gibi belirli birkaç dilde kullanılır. Burada çift noktalama işaretinden önce boşluk olması bir yazım kuralıdır.

Ayrıca bakınız:

[Vikipedi üzerinde Fransız ve İngiliz boşlukları](#)

Kurallı ifade

3.9 sürümünde geldi.

Özet

Çevirideki kurallı ifade aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.placeholders.RegexCheck`

Belirteci denetle

`regex`

Etkinleştirilecek im

`regex`

Yok sayılacak im

`ignore-regex`

Çeviri kurallı ifadeyle eşleşmiyor. İfade çeviri dosyasından çıkarılır ya da el ile `regex` imi kullanılarak tanımlanır:

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

Tekil ile çoğul biçim aynı

Özet

Bazı çoğul biçimler tekil biçimler ile aynı

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.consistency.SamePluralsCheck`

Belirteci denetle

same-plurals

Yok sayılacak im

ignore-same-plurals

Bazı çoğul biçimlerin çeviride aynı olup olmadığını denetler. Çoğu dilde farklı olmaları gerekir.

Başta yeni satır var**Özet**

Kaynağın ve çevirinin ikisi de yeni satırla başlamıyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

weblate.checks.chars.BeginNewlineCheck

Belirteci denetle

begin_newline

Yok sayılacak im

ignore-begin-newline

Yeni satırlar genellikle iyi bir nedenden dolayı kaynak dizgelerde görünür. Eksik ya da fazla olması, çevrilmiş metin kullanıma sunulduğunda biçimlendirme sorunlarına yol açabilir.

Ayrıca bakınız:

Sondaki yeni satır

Baştaki boşluk**Özet**

Kaynağın ve çevirinin ikisi de aynı sayıda boşlukla başlamıyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

weblate.checks.chars.BeginSpaceCheck

Belirteci denetle

begin_space

Yok sayılacak im

ignore-begin-space

Bir dizgenin başındaki boşluk genellikle arayüzde girinti sağlamak için kullanılır ve bu nedenle korunması önemlidir.

Sondaki yeni satır**Özet**

Kaynağın ve çevirinin ikisi de yeni satırla bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

weblate.checks.chars.EndNewlineCheck

Belirteci denetle

end_newline

Yok sayılacak im

`ignore-end-newline`

Yeni satırlar genellikle iyi bir nedenden dolayı kaynak dizgelerde görünür. Eksik ya da fazla olması, çevrilmiş metin kullanıma sunulduğunda biçimlendirme sorunlarına yol açabilir.

Ayrıca bakınız:

Başta yeni satır var

Sondaki boşluk

Özet

Kaynağın ve çevirinin ikisi de boşlukla bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndSpaceCheck`

Belirteci denetle

`end_space`

Yok sayılacak im

`ignore-end-space`

Kaynağın sonundaki boşlukların çeviri ile aynı olup olmadığını denetler.

Sondaki boşluk genellikle arkadan gelen ögeleri ayırmak için kullanılır. Bu nedenle kaldırılması görünümü bozabilir.

Çeviri değiştirilmemiş

Özet

Kaynak ile çeviri aynı

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.same.SameCheck`

Belirteci denetle

`same`

Yok sayılacak im

`ignore-same`

Kaynak ve karşılık gelen çeviri dizgeleri, çoğul biçimlerden en az birine kadar aynıysa gerçekleşir. Tüm dillerde yaygın olarak bulunan bazı dizgeler yok sayılır ve çeşitli markup kodları ayıklanır. Bu, yanlış pozitif sonuçların sayısını azaltır.

Bu denetim, yanlışlıkla çevrilmemiş dizgelerin bulunmasına yardımcı olabilir.

Bu denetimin varsayılan davranışı, yerleşik kara listeden sözcüklerin denetime katılmamasıdır. Bunlar sıklıkla çevrilmeyen sözcüklerdir. Bu, yalnızca birkaç dilde aynı olan tek bir sözcükten oluşan kısa dizgelerde yanlış pozitif sonuçlardan kaçınmak için yararlıdır. Bu kara liste, dizgeye veya bileşene `strict-same` imi eklenerek devre dışı bırakılabilir.

4.17 sürümünde değişti: `check-glossary` imiyle (bilgi almak için: *Sözlük ile uyumlu değil*), çevrilemeyecek sözlük terimleri denetime katılmaması sağlanır.

Ayrıca bakınız:

Sözlük ile uyumlu değil, Bileşen yapılandırması, Parametreleri kullanarak davranışı özelleştirmek

Güvenli olmayan HTML

3.9 sürümünde geldi.

Özet

Çeviride güvenli olmayan HTML kodlaması var

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.SafeHTMLCheck`

Belirteci denetle

`safe-html`

Etkinleştirilecek im

`safe-html`

Yok sayılacak im

`ignore-safe-html`

Çeviride, güvenli olmayan HTML biçimlendirmesi kullanılmış. Bu denetimin `safe-html` imiyle etkinleştirilmesi gerekir (bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)). İmlemeyi kendiliğinden yapabilen bir kendiliğinden düzeltici de vardır.

İpucu: `md-text` imi de kullanıldığında, Markdown biçimindeki bağlantılar da kullanılabilir.

Ayrıca bakınız:

HTML denetimi [Ammonia](#) kitaplığı kullanılarak yapılır.

Adres

3.5 sürümünde geldi.

Özet

Çeviride bir adres bulunmuyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.URLCheck`

Belirteci denetle

`url`

Etkinleştirilecek im

`url`

Yok sayılacak im

`ignore-url`

Çeviride bir adres yok. Bu denetim, yalnızca birimin adres içeriyor olarak imlenmesi durumunda tetiklenir. Bu durumda çeviride geçerli bir adres bulunması gerekir.

XML kodlaması

Özet

Çevirideki XML kod imleri kaynakla aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.XMLTagsCheck`

Belirteci denetle

`xml-tags`

Yok sayılacak im

`ignore-xml-tags`

Bu durum genellikle sonuçtaki çıktının farklı görüneceği anlamına gelir. Çoğu durumda bu, çevirinin değiştirilmesinden istenen bir sonuç değildir, ancak bazen de öyledir.

Kaynaktaki XML kod imlerinin çeviri ile aynı olup olmadığını denetler.

Denetim, XML benzeri dizgeler için kendiliğinden etkinleştirilir. Açmaya zorlamak için bazı durumlarda `xml-text` imini eklemeniz gerekebilir.

Not: `safe-html` imi bu denetimi devre dışı bırakır çünkü yaptığı HTML temizliği geçerli bir XML olmayan HTML markup kodu üretebilir.

XML söz dizimi

Özet

Çeviri geçerli bir XML kodu değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.XMLValidityCheck`

Belirteci denetle

`xml-invalid`

Yok sayılacak im

`ignore-xml-invalid`

XML kod imi geçersiz.

Denetim, XML benzeri dizgeler için kendiliğinden etkinleştirilir. Açmaya zorlamak için bazı durumlarda `xml-text` imini eklemeniz gerekebilir.

Not: `safe-html` imi bu denetimi devre dışı bırakır çünkü yaptığı HTML temizliği geçerli bir XML olmayan HTML markup kodu üretebilir.

Sıfır genişlikli boşluk

Özet

Çeviride fazladan sıfır genişlikli boşluk karakteri var

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.ZeroWidthSpaceCheck`

Belirteci denetle

`zero-width-space`

Yok sayılacak im

`ignore-zero-width-space`

Sıfır genişlikli boşluk (<U+200B>) karakterleri, sözcüklerin içindeki iletileri bölmek için kullanılır (sözcük kaydırma için).

Genellikle yanlışlıkla eklendiklerinden, çeviride bulunduklarında bu denetim tetiklenir. Bazı programlarda bu karakter kullanıldığında sorunlar olabilir.

Ayrıca bakınız:

[Vikipedi üzerinde sıfır genişlikli boşluk](#)

1.6.4 Kaynak denetimleri

Kaynak denetimleri, geliştiricilerin kaynak dizgelerin kalitesini artırmasına yardımcı olabilir.

Üç nokta

Özet

Dizge, üç nokta karakteri (...) yerine üç ayrı nokta (...) kullanıyor

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.source.EllipsisCheck`

Belirteci denetle

`ellipsis`

Yok sayılacak im

`ignore-ellipsis`

Bu denetim hatası dizgede üç nokta (...) karakteri kullanılması gerekiyorken üç ayrı nokta (...) kullanıldığında görülür.

Çoğu durumda Unikod karakterini kullanmak daha iyi bir yaklaşımdır ve daha iyi işlenir. Metinden konuşmaya çevirme uygulamalarında da daha iyi bir ses çıkarabilir.

Ayrıca bakınız:

[Vikipedi üzerinde üç nokta karakteri](#)

ICU MessageFormat söz dizimi

4.9 sürümünde geldi.

Özet

ICU MessageFormat dizgelerinde söz dizimi hataları var.

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.icu.ICUSourceCheck`

Belirteci denetle

`icu_message_format_syntax`

Etkinleştirilecek im

`icu-message-format`

Yok sayılacak im

`ignore-icu-message-format`

Ayrıca bakınız:

ICU MessageFormat

Uzun süredir çevrilmemiş

4.1 sürümünde geldi.

Özet

Dizge uzun zamandır çevrilmemiş

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.source.LongUntranslatedCheck`

Belirteci denetle

`long_untranslated`

Yok sayılacak im

`ignore-long-untranslated`

Dizgenin uzun süredir çevrilmemiş olması, kaynak dizgede çeviri yapmayı zorlaştıran bir sorun olduğunu gösterebilir.

Birden çok denetim hatası

Özet

Birkaç dildeki çevirilerde denetim hataları var

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.source.MultipleFailingCheck`

Belirteci denetle

`multiple_failures`

Yok sayılacak im

`ignore-multiple-failures`

Bu dizgenin çevirisinde çok sayıda kalite denetimi hatası vardır. Bu durum genellikle kaynak dizgeyi geliştirmek için bir şeyler yapılabileceğini gösterir.

Bu denetim hatası genellikle bir cümlenin sonundaki eksik bir noktadan ya da çevirmenlerin çeviride düzeltme eğiliminde oldukları benzer küçük sorunlardan kaynaklanabilir ve kaynak dizgede düzeltmek daha iyi olacaktır.

Birden çok değişken adlandırılmamış

4.1 sürümünde geldi.

Özet

Dizgede birden çok adlandırılmamış değişken var. Çevirmenler istese de bunların sıralamasını değiştiremez.

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.format.MultipleUnnamedFormatsCheck`

Belirteci denetle

`unnamed_format`

Yok sayılacak im

`ignore-unnamed-format`

Dizgede birden çok adlandırılmamış değişken var. Çevirmenler istese de bunların sıralamasını değiştiremez.

Çevirmenlerin bunların sıralamasını değiştirebilmesi için adlandırılmış değişkenler kullanmayı değerlendirin.

Çoğul değil

Özet

Dizge çoğul olarak kullanılmış. Ancak çoğul biçimler kullanılmamış.

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.source.OptionalPluralCheck`

Belirteci denetle

`optional_plural`

Yok sayılacak im

`ignore-optional-plural`

Dizge çoğul olarak kullanılmış, ancak çoğul biçimleri kullanmıyor. Çeviri sisteminizin bunu desteklemesi durumunda, çoğul farkındalığı olan çeşidini kullanmalısınız.

Örneğin, Python üzerindeki gettext ile şöyle olabilir:

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

1.7 Arama

3.9 sürümünde geldi.

İstedığınız dizgeleri bulmak için boole işlemleri, parantezler veya alana özgü arama kullanan gelişmiş sorgular kullanılabilir.

Herhangi bir alan belirtilmediğinde, arama kaynak, hedef ve bağlam dizgelerinde yapılır.

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

Search

All strings ▾

Sort By ▾

Advanced query builder

Source strings ▾ Search for... ☐ Exact Add

String changed after ▾ mm/dd/yyyy ☐ Add

Query examples

Review strings changed by other users	<code>changed:>=2023-03-17 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:>=2023-03-17 AND state:<=needs-editing</code>	Add
Translated strings in a certain language	<code>is:translated AND language:cs</code>	Add

Search

1.7.1 Basit arama

Arama kutusuna yazılan herhangi bir ifade sözcüklere ayrılır. Bunlardan herhangi birini içeren dizgeler görüntülenir. İfadenin tümüne uyan dizgeleri aramak için, “arama ifadesini” tırnak arasına alın (tek (') ya da çift (")) tırnak kullanabilirsiniz): "Alıntılanmış bir dizge" ya da 'başka bir alıntılanmış dizge' gibi.

1.7.2 Alanlar

source:METİN

Büyük/küçük harfe duyarsız kaynak dizge arama.

target:METİN

Hedef dizgede büyük/küçük harfe duyarsız arama.

context:METİN

Bağlam dizgesinde büyük/küçük harfe duyarsız arama.

key:METİN

Anahtar dizgesinde büyük/küçük harfe duyarsız arama.

note:METİN

Büyük/küçük harfe duyarsız kaynak dizge açıklaması arama.

location:METİN

Konum dizgesinde büyük/küçük harfe duyarsız arama.

priority:SAYI

Dizge önceliği.

id:NUMBER

Eşsiz dizge belirteci.

position:NUMBER

Dizgenin çeviri dosyasındaki konumu.

added:TARİHSAAT

Dizgenin Weblate üzerine ne zaman eklendiğini belirten zaman damgası.

state:METİN

Dizge durumuna göre arama (approved, translated, needs-editing, empty, read-only), [Alan işlemleri](#) desteklenir.

pending:BOOLEAN

Sürüm denetimi sistemine boşaltılmak üzere bekleyen dizgeler.

has:METİN

Öznitelikleri olan dizgelerde arama - plural, context, suggestion, comment, check, dismissed-check, translation, variant, screenshot, flags, explanation, glossary, note, label.

is:METİN

Bekleyen çevirilerde arama (pending). Ayrıca tüm dizge durumları da aranabilir (approved, translated, untranslated, needs-editing, read-only).

language:METİN

Hedef dizge dili.

component:METİN

Bileşen kısaltması ya da adına göre büyük/küçük harfe duyarsız arama. Bilgi almak için: [Bileşen kısaltması](#) ve [Bileşen adı](#).

project:METİN

Proje kısaltmasına göre arama. Bilgi almak için: :ref:`project-slug`.

changed_by:METİN

Belirtilen kullanıcı adlı yazar tarafından değiştirilmiş dizgeleri arama.

changed:TARİHSAAT

Belirtilen tarihte içeriği değiştirilmiş dizgeleri arama. *Alan işlemleri* desteklenir.

change_time:TARİHSAAT

Belirtilen tarihte değiştirilmiş dizgeleri arama. *Alan işlemleri* desteklenir. `changed` seçeneğinden farklı olarak içerik değiştirilmese de `change_action` için özel işlemlerin süzülmesini sağlar.

change_action:METİN

Değişiklik işlemiyle ilgili süzgeçler, `change_time` ile birlikte kullanışlı olur. Değişiklik işleminin İngilizce adını, boşluklar ile tırnak arasında ya da boşluklar yerine tire ile küçük harflerle yazabilirsiniz. Örnekler için *Değişiklik arama* bölümüne bakabilirsiniz.

check:METİN

Denetim hatası olan dizgelerde arama. Denetim tanımlayıcıları için *Denetimler ve düzeltmeler* bölümüne bakabilirsiniz.

dismissed_check:METİN

Denetim hatası yok sayılan dizgelerde arama. Denetim tanımlayıcıları için *Denetimler ve düzeltmeler* bölümüne bakabilirsiniz.

comment:METİN

Kullanıcı yorumlarında arama.

resolved_comment:METİN

Çözümlemiş yorumlarda arama.

comment_author:METİN

Yorum yazarına göre süzme.

suggestion:METİN

Önerilerde arama.

suggestion_author:METİN

Öneri yazarına göre süzme.

explanation:METİN

Açıklamalarda arama.

label:METİN

Etiketlerde arama.

screenshot:METİN

Ekran görüntülerinde arama.

1.7.3 Boole işlemleri

Karmaşık sorgular oluşturmak için aramaları AND, OR, NOT ve parantez kullanarak birleştirebilirsiniz. Örneğin:
`state:translated AND (source:hello OR source:bar)`

1.7.4 Alan işlemleri

Tarih veya sayısal aramalar için işlemler, aralıklar veya parçalı aramalar belirtebilirsiniz:

state:> =translated

Durumu `translated` ya da daha iyisi (`approved`) olanlar.

changed:2019

2019 yılında değiştirilmişler.

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

Belirtilen iki tarih arasında değiştirilmişler.

position:[10 to 100]

Konumu 10 ile 100 arasında olan dizgeler (10 ve 100 dahil).

1.7.5 Kesin işlemler

= işlemini kullanarak farklı dizge alanlarında tam eşleşme sorgusu yapabilirsiniz. Örneğin, merhaba dünya ile tam olarak eşleşen tüm kaynak dizgeleri aramak için şunu kullanın: `source:="hello world"`. Tek sözcüklü ifadeleri aramak için tırnak işaretlerini kullanmayabilirsiniz. Örneğin, merhaba ile eşleşen tüm kaynak dizgelerini aramak için şunu kullanabilirsiniz: `source:=hello`.

1.7.6 Değişiklik arama

4.4 sürümünde geldi.

Geçmiş kayıtları `change_action` ve `change_time` ölçütleri kullanılarak aranabilir.

Örneğin, 2018 yılında, düzenlenecek olarak imlenmiş dizgeleri aramak için `change_time:2018 AND change_action:marked"-for-edit` ya da `change_time:2018 AND change_action:``Marked for edit` kullanılabilir.

1.7.7 Kurallı ifadeler

Ayrıca metin kabul edilen her yerde, bir kurallı ifadeyi `r"kurallıifade"` olarak belirtebilirsiniz.

Örneğin, 2 ile 5 arasında basamağı olan tüm kaynak dizgeleri aramak için `source:r"[2-5]"` kullanın.

1.7.8 Hazır sorgular

Arama sayfasında hazır sorgulardan seçim yapabilirsiniz. Böylece sık yapılan aramaları hızlı bir şekilde kullanabilirsiniz:

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 Sonuçları sıralamak

Dizgeleri gerek duyduğunuz biçimde sıralayabileceğiniz birçok seçenek bulunur:

The screenshot displays the Weblate web interface. At the top, there's a navigation bar with links to Dashboard, Projects, Languages, and Checks. Below this, the breadcrumb trail shows 'WeblateOrg / Django / Czech / Translate'. A progress indicator shows 'translated 96%'. The main area features a translation form with fields for English and Czech. The English field contains the text 'The string uses three dots (...) instead of an ellipsis character (...)'. The Czech field is empty. Below the form are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. A dropdown menu is open, showing options like 'Position and priority', 'Position', 'Priority', 'Labels', 'Source string', 'Target string', 'String age', 'Number of words', 'Number of comments', 'Number of failing checks', 'Key', and 'String location'. The right sidebar contains several panels: 'Glossary' (showing no related strings), 'String information' (with fields for Screenshot context, Explanation, Labels, Flags, Source string location, String age, Source string age, and Translation file), and 'Nearby strings' (showing 16 items). At the bottom, there's a 'New comment' section with a text area and a 'Save' button.

Powered by Weblate 4.17 · [About Weblate](#) · [Legal](#) · [Contact](#) · [Documentation](#) · [Donate to Weblate](#)

1.8 Çeviri iş akışları

Weblate kullanmak, sizi çevirmenlerinize ve kullanıcılarınıza yaklaştıran bir süreçtir. Hangi özellikleri kullanmak istediğinize siz karar verirsiniz.

Aşağıda, farklı Weblate yapılandırmalarının tümü bulunmuyor. Farklı iş akışlarını burada listelenen yaygın uygulamalara göre yapılandırabilirsiniz.

1.8.1 Çeviri erişimi

Erişim denetimi, seçeneklerinin çoğu herhangi bir iş akışına uygulanabileceğinden, iş akışlarında tüm ayrıntılarına girilmemiştir. Çevirilere erişimin nasıl yönetileceği ile ilgili bilgi almak için ilgili belgelere bakın.

İzleyen bölümlerde, *tüm kullanıcılar*, çeviriye erişimi olan bir kullanıcı anlamına gelir. Proje herkese açıksa, kimliği doğrulanmış herhangi bir kullanıcı veya projede *Çevirme* izni olan bir kullanıcı olabilir.

1.8.2 Çeviri durumları

Çevrilmiş her dizge şu durumlardan birinde olabilir:

Çevrilmemiş

Çeviri yapılmamıştır ve boştur. Dosya biçimine bağlı olarak dosyada bulunabilir ya da bulunmayabilir.

Düzenlenmesi gerekli

Çevirinin düzenlenmesi gereklidir. Genellikle bir kaynak dizge değişikliğinden, belirsiz eşleşmeden veya çevirmen işleminden kaynaklanır. Çeviri, dosya biçimine bağlı olarak dosyada bulunabilir ve düzenlenmesi gerekli şekilde imlenebilir (örneğin, gettext dosyasında `fuzzy` olarak imlenir).

Onaylanmayı bekliyor

Çeviri yapılmış, ancak onaylanmamıştır. Dosyada geçerli bir çeviri olarak bulunur.

Onaylanmış

Çeviri onaylanmıştır. Artık çevirmenler tarafından değiştirilemez. Yalnızca onaylayanlar değiştirebilir. Çevirmenler yalnızca öneri ekleyebilir.

Bu durum yalnızca onaylama özelliği etkinleştirildiğinde kullanılabilir.

Öneriler

Öneriler yalnızca Weblate üzerinde bulunur. Çeviri dosyasında bulunmaz.

Durumlar, çeviri dosyalarında olabildiğince belirtilir.

İpucu: Kullandığınız dosya biçimi durumların kaydedilmesini desteklemiyorsa, değiştirilmemiş dizgeleri düzenlenmesi gerekli olarak imlemek için *Değiştirilmemiş çevirileri* “*Düzenlenmesi gerekli*” olarak *imleme* eklentisini kullanmak isteyebilirsiniz.

Ayrıca bakınız:

Çeviri türü yetenekleri, *Çeviri iş akışları*

1.8.3 Doğrudan çeviri

Bu seçenek küçük takımlar için en yaygın kullanılan kurulumdur. Herkes doğrudan çeviri yapabilir. Varsayılan Weblate kurulumudur.

- *Tüm kullanıcılar* çevirileri düzenleyebilir.
- Öneriler, çevirmenlerin çeviride bir değişiklik yapmaktan emin olmadığı durumlarda isteğe bağlı bir değişiklik önerisinde bulunmalarını sağlar.

Ayar	Değer	Not
Onaylama kullanılsın	kapalı	Proje düzeyinde yapılandırılır.
Öneriler kullanılsın	açık	Kullanıcıların emin olmadıkları durumlarda öneri yapabilmeleri açısından yararlıdır.
Öneriler oylanabilsin	kapalı	
Öneriler kendiliğinden kabul edilsin	0	
Çevirmenler grubu	<i>Kullanıcılar</i>	Ya da <i>per-project access control</i> ile <i>Çevirmek</i> .
Onaylayıcılar grubu	Kullanılmaz	Kullanılmıyor.

1.8.4 Akran değerlendirmesi

Bu iş akışında, herkes öneri ekleyebilir. Önerilerin onaylanmış olarak kabul edilmesi için başka üyelerin onaylaması gerekir.

- *Tüm kullanıcılar* öneri ekleyebilir.
- *Tüm kullanıcılar* önerilere oy verebilir.
- Önerilerin onaylanmış olarak kabul edilmesi için önceden belirlenmiş sayıda oy alması gerekir.

Ayar	Değer	Not
Onaylama kullanılsın	kapalı	Proje düzeyinde yapılandırılır.
Öneriler kullanılsın	açık	
Öneriler oylanabilsin	kapalı	
Öneriler kendiliğinden kabul edilsin	1	Daha fazla akran değerlendirmesi gerekmesi için yüksek bir değer ayarlayabilirsiniz.
Çevirmenler grubu	<i>Kullanıcılar</i>	Ya da <i>per-project access control</i> ile <i>Çevirmek</i> .
Onaylayıcılar grubu	Kullanılmaz	Kullanılmadığında, tüm çevirmenler onaylayabilir.

1.8.5 Özel onaylayanlar

Özel onaylayanlar kullanıldığında, biri çeviri yapan, diğeri de çevirilerin tutarlılığı ve kalitesini sağlamak için bunları gözden geçirip onaylayan iki kullanıcı grubu bulunur.

- *Tüm kullanıcılar* onaylanmamış çevirileri düzenleyebilir.
- *Onaylayan* dizgeleri onaylayabilir ve onayı kaldırabilir.
- *Onaylayan* tüm çevirileri düzenleyebilir (onaylanmışlar ile birlikte).
- Öneriler, onaylanmış dizgelerde değişiklik önermek amacıyla da kullanılabilir.

Ayar	Değer	Not
Onaylama kullanılsın	açık	Proje düzeyinde yapılandırılır.
Öneriler kullanılsın	kapalı	Kullanıcıların emin olmadıkları durumlarda öneri yapabilmeleri açısından yararlıdır.
Öneriler oylanabilsin	kapalı	
Öneriler kendiliğinden kabul edilsin	0	
Çevirmenler grubu	<i>Kullanıcılar</i>	Ya da <i>per-project access control</i> ile <i>Çevirmek</i> .
Onaylayıcılar grubu	<i>Onaylayanlar</i>	Ya da <i>per-project access control</i> ile <i>Onayla</i> kullanılabilir.

1.8.6 Onaylama özelliğini açmak

Onaylama, proje yapılandırmasında, proje ayarlarındaki *İş akışı* bölümünden açılabilir (*Yönetim* → *Ayarlar* yoluyla bulabilirsiniz):

Webplate / 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.

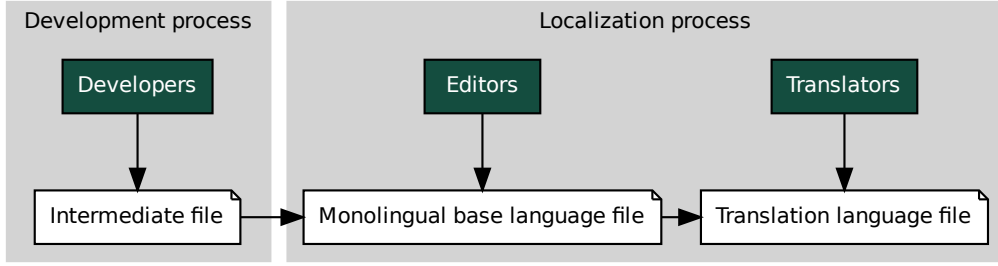
Save

Powered by Weblate 4.17 About Weblate Legal Contact Documentation Donate to Weblate

1.8.7 Kaynak dizgeleri için kalite aracı

Çoğu durumda, özgün kaynak dil dizgeleri geliştiricilerden gelir. Çünkü kodu yazma ve ilk dizgeleri sağlama işini onlar yapar. Bununla birlikte, geliştiricilerin ana dili genellikle kaynak dil değildir ve kaynak dizgeleri için istenilen kaliteyi sağlayamazlar. Ara çeviri bu sorunu çözmenize yardımcı olabilir. Geliştiriciler, çevirmenler ve kullanıcılar arasındaki dizgeleri için kullanılacak bir ek kalite aracı vardır.

Ara dil dosyası ayarlandığında, bu dosya dizgeleri için kaynak olarak kullanılır. Ancak güzelleştirmek için kaynak dile göre düzenlenir. Dizge kaynak dilde hazır olduğunda, çevirmenlerin diğer dillere çevirmesi için de kullanılabilir.

**Ayrıca bakınız:**

Ara dil dosyası, Tek dilli temel dil dosyası, İki dilli ve tek dilli biçimler

1.8.8 Kaynak dizge onayları

Kaynak onaylama kullanılsın seçeneği etkinleştirildiğinde, onaylama işlemi kaynak dizgelere de uygulanabilir. Bu seçenek etkinleştirildikten sonra, kullanıcılar kaynak dizgelerdeki sorunları bildirebilir. Gerçek süreç, iki dilli ya da tek dilli biçimleri kullanıp kullanmadığınıza göre değişir.

Tek dilli biçimlerde, kaynak dizge incelemesi *Özel onaylayanlar* ile benzer şekilde davranır. Kaynak dizge için bir sorun bildirildiğinde dizge *Düzenlenmesi gerekli* olarak imlenir.

İki dilli biçimlerde, kaynak dizgelerin doğrudan düzenlenmesine izin verilmez (bunlar genellikle doğrudan kaynak kodundan ayıklanır). Bu durumda çevirmenler tarafından bildirilen dizgelere *Kaynağın incelenmesi gerekli* etiketi eklenir. Bu dizgeleri gözden geçirerek kaynağı düzenlemeniz ya da etiketi kaldırmanız gerekir.

Ayrıca bakınız:

İki dilli ve tek dilli biçimler, Özel onaylayanlar, labels, Yorumlar

1.9 Sık sorulan sorular

1.9.1 Yapılandırma

Kendiliğinden çalışan bir iş akışı nasıl oluşturulur?

Weblate, tüm çeviri işlerini sizin için yarı otomatik olarak yapabilir. Deponuza anında erişim izni vererseniz, bazı birleştirme çakışmaları oluşmadığı sürece çeviriler etkileşim olmadan yapılabilir.

1. Git deponuzu Weblate üzerinde herhangi bir değişiklik olduğunda bildirecek şekilde ayarlayın. Bunun nasıl yapılacağı ile ilgili bilgi almak için *Bildirim kancaları* bölümüne bakabilirsiniz.
2. Weblate içindeki *Bileşen yapılandırması* üzerinde bir itme adresi ayarlayın. Böylece Weblate değişiklikleri deponuza gönderebilir.
3. Weblate içindeki *Bileşen yapılandırması* üzerinde *İşleme ile itme* seçeneğini açın. Böylece Weblate üzerinde yapılan değişiklikler deponuza itilir.

Ayrıca bakınız:

Sürekli yerelleştirme, Birleştirme çakışmalarından kaçınmak

Depolara SSH üzerinden nasıl erişilir?

SSH anahtarlarını ayarlamak ile ilgili bilgi almak için [Depolara erişmek](#) bölümüne bakabilirsiniz.

Çevirilerdeki birleştirme çakışmaları nasıl düzeltilir?

Birleştirme çakışmaları, çeviri dosyası hem Weblate üzerinde hem de yukarı akış deposunda değiştirildiğinde ortaya çıkar. Bu sorunu genellikle çeviri dosyalarında değişiklik yapmadan önce Weblate çevirilerini birleştirerek (örneğin, msgmerge çalıştırmadan önce) önleyebilirsiniz. Weblate üzerinde bekleyen tüm çevirilerin işlenmesini isteyin (Yönetim menüsündeki *Depo bakımı* bölümünden yapabilirsiniz) ve depoyu birleştirin (kendiliğinden itme açık değilse).

Zaten bir birleştirme çakışmasıyla karşılaştıysanız, makinenizdeki tüm çakışmaları yerel olarak çözmenin en kolay yolu, Weblate kurulumunu bir uzak depo olarak eklemek, yukarı akışta birleştirmek ve çakışmaları düzeltmektir. Değişiklikleri geri ittiğinizde, Weblate birleştirilmiş sürümü başka herhangi bir özel işlem yapmadan kullanabilir.

Not: Kurulumunuza bağlı olarak, Weblate deposuna erişim için kimlik doğrulaması gerektirebilir. Weblate üzerinde yerleşik *Git dışı aktarıcı* özelliğini kullanırken, kimliğinizi kullanıcı adınız ve API anahtarınızla doğrularsınız.

```
# 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
```

Weblate üzerinde birden fazla dal kullanıyorsanız, hepsine aynı şeyi yapabilirsiniz:

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

# Merge QA_4_7 branch:
git checkout QA_4_7
git merge weblate-one/QA_4_7
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
... # 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
```

Gettext PO dosyaları söz konusu olduğunda, çakışmaları yarı otomatik bir şekilde birleştirmenin bir yolu vardır:

Weblate Git deposunun yerel bir kopyasını alın ve tutun. Ayrıca, yukarı akış Git deposunun ikinci bir yeni yerel kopyasını da alın (yani, yukarı akış Git deposunun iki kopyası olmalıdır: Bozulmamış ve çalışan bir kopya):

```
# 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
```

Ayrıca bakınız:

Weblate tarafından kullanılan Git deposu nasıl dışa aktarılır?, Sürekli yerelleştirme, Birleştirme çakışmalarından kaçınmak, Weblate istemcisi

Aynı anda birkaç dalı nasıl çevirebilirim?

Weblate, çeviri değişikliklerinin tek bir *Proje yapılandırması* içine itilmesini destekler. Açık olan her *Bileşen yapılandırması* için (varsayılan davranış), yapılan değişiklik kendiliğinden diğerlerine yayılır. Böylece, dalların kendileri zaten oldukça fazla farklılaşmış olsa bile çeviriler eşitlenmiş olarak tutulur ve çeviri değişiklikleri aralarında birleştirilemez.

Weblate üzerindeki değişiklikleri birleştirdikten sonra, farklılıkları atarak bu dalları birleştirmeniz gerekebilir (geçirtilme iş akışınıza bağlı olarak):

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

Ayrıca bakınız:

Bileşenler arasında çevirilerin tutarlılığını sağlamak

Çok platformlu projeler nasıl çevrilir?

Weblate çok çeşitli dosya biçimlerini destekler (bilgi almak için: *Desteklenen dosya biçimleri*) ve en kolay yaklaşım her platformun yerel biçimini kullanmaktır.

Tüm platform çeviri dosyalarını tek bir projeye bileşen olarak ekledikten sonra (bilgi almak için: *Çeviri projelerini ve bileşenleri eklemek*), tüm platformlar için dizgeleri aynı anda çevirmek üzere çeviri yayma özelliğini (varsayılan olarak açıktır ve *Bileşen yapılandırması* içinden kapatılabilir) kullanabilirsiniz.

Ayrıca bakınız:

Bileşenler arasında çevirilerin tutarlılığını sağlamak

Weblate tarafından kullanılan Git deposu nasıl dışa aktarılır?

Depo ile ilgili özel bir şey yoktur, `DATA_DIR` klasörü altında bulunur ve `vcs/<project>/<component>` olarak adlandırılır. Bu makineye SSH erişiminiz varsa, depoyu doğrudan kullanabilirsiniz.

Anonim erişim için bir Git sunucusu işletmek ve depoyu dış dünyaya sunmasına izin vermek isteyebilirsiniz.

Alternatif olarak, bu işlemin kendiliğinden yapılabilmesi için Weblate içindeki *Git dışa aktarıcı* komutunu kullanabilirsiniz.

Değişiklikleri yukarı akışa geri itme seçenekleri nelerdir?

Bunlar büyük ölçüde kurulumunuza bağlıdır. Weblate bu alanda oldukça esnek. Burada Weblate ile kullanılan bazı iş akışı örneklerini bulabilirsiniz:

- Weblate, değişiklikleri kendiliğinden gönderir ve birleştirir (bilgi almak için: *Kendiliğinden çalışan bir iş akışı nasıl oluşturulur?*).
- Eli ile Weblate tarafından gönderilmesini söylersiniz (yukarı akış deposuna itme erişimi olmalıdır).
- Birisi Weblate Git deposundaki değişiklikleri yukarı akış deposuyla el ile birleştirir.
- Birisi Weblate tarafından üretilen geçmişi yeniden yazar (örneğin, birleştirme işlemlerini ortadan kaldırarak), değişiklikleri birleştirir ve Weblate'e yukarı akış deposundaki içeriği sıfırlamasını söyler.

Tabii ki tüm bunları istediğiniz gibi bir arada kullanabilirsiniz.

Weblate erişimini, kaynak kodunu açığa çıkarmadan nasıl yalnızca çevirilerle sınırlandırabilirim?

Çevirileri sürüm denetimi altında tutarken kaynak kodundan ayırmak için `git submodule` kullanabilirsiniz.

1. Çeviri dosyalarınızla bir depo oluşturun.
2. Bunu kodunuza bir alt modül olarak ekleyin:

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

3. Weblate ile bu depoyu bağlayın. Artık Weblate tarafından kaynak kodunuzu içeren depoya erişilmesi gerekmez.
4. Ana depoyu Weblate üzerindeki çevirilerle güncelleyebilirsiniz:

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

Ayrıntılı bilgi almak için `git submodule` belgelerine bakabilirsiniz.

Weblate kurulumunun düzgün olup olmadığını nasıl denetleyebilirim?

Weblate içinde, yönetim arayüzünde görebileceğiniz bir dizi yapılandırma denetimi bulunur. Yönetim arayüzündeki *Başarım raporu* bağlantısını izleyin ya da doğrudan `/manage/performance/` adresini açın.

Ayrıca bakınız:

Weblate uygulamasını izlemek, Celery durumunu izlemek

Neden tüm işlemler Weblate <noreply@weblate.org> tarafından yapılıyor?

`DEFAULT_COMMITTER_EMAIL` ve `DEFAULT_COMMITTER_NAME` seçenekleri ile yapılandırılan varsayılan gönderici adı.

Her işleminin yazarı (kullanılan sürüm denetimi sistemi destekliorsa), çeviriyi yapan kullanıcı olarak doğru kaydedilir.

Yazarı bilinmeyen işlemler için (anonim öneriler ya da makine çevirisi sonuçları gibi), yazarın hesabı anonim kullanıcı olarak tutulur (bilgi almak için: `ANONYMOUS_USER_NAME`). Ad ve e-posta bilgilerini yönetim arayüzünden değiştirebilirsiniz.

Ayrıca bakınız:

Bileşen yapılandırması

Weblate üzerindeki geçmiş kaybetmeden depodaki dosyalar nasıl taşınır?

Dosya konumunu değiştirdikten sonra dizgelere bağlı geçmiş, yorumları ya da ekran görüntülerini tutmak için, bu dizgelerin Weblate üzerinde hiçbir zaman silinmeyeceğinden emin olmanız gerekir. Bu kaldırma işlemleri, Weblate deposunun güncellenmesi durumunda gerçekleşebilir. Ancak bileşen yapılandırması hala eski dosyaları gösterir. Bu durumda, Weblate tüm çevirileri silmesi gerektiğini varsayar.

Bunun çözümü, işlemi Weblate ile eşitleyerek yapmaktır:

1. Weblate üzerinde etkilenen bileşeni kilitleyin.
2. Bekleyen değişiklikleri işleyin ve depo yukarı akışına birleştirin.
3. İnternet kancalarını almayı devre dışı bırakın: `ref:project`. Böylece Weblate depodaki değişiklikleri hemen görmez.
4. Depoda gerekli değişiklikleri yapın (örneğin `git mv` kullanarak) ve bunları yukarı akış deposuna itin.
5. *Bileşen yapılandırması* ögesini yeni kurulumla eşleştirecek şekilde değiştirin. Yapılandırmayı değiştirdikten sonra, Weblate güncellenmiş depoyu alır ve var olan dizgeleri korurken değiştirilen konumları fark eder.
6. Bileşenin kilidini açın ve proje yapılandırmasındaki kancaları yeniden etkinleştirin.

1.9.2 Kullanım

Başkalarının çevirilerini nasıl onaylarım?

- Weblate üzerinde kullanılabilen birkaç onaylama tabanlı iş akışı bulunur. Bilgi almak için: *Çeviri iş akışları*.
- *Bildirimler* içinde yapılan herhangi bir değişikliğe abone olabilir ve ardından e-posta ile gelen diğer katkıları gözden geçirebilirsiniz.
- Çeviri görünümünün alt bölümünde, belirli bir tarihten sonra başkaları tarafından yapılan çevirileri görebileceğiniz bir onaylama aracı bulunur.

Ayrıca bakınız:

Çeviri iş akışları

Bir kaynak dizge hakkında nasıl geri bildirim sağlayabilirim?

Çevirinin altındaki bağlam bölümünde, bir kaynak dizge hakkında geri bildirim vermek ya da diğer çevirmenlerle tartışmak için *Yorumlar* sekmesini kullanabilirsiniz.

Ayrıca bakınız:

report-source, *Yorumlar*

Çeviri yaparken var olan çevirileri nasıl kullanabilirim?

- Paylaşılan çeviri belleği ile, Weblate içindeki tüm çeviriler kullanılabilir.
- Var olan çeviri belleği dosyalarını Weblate üzerine aktarabilirsiniz.
- Özetleri çeviriler, öneriler ya da düzenlenmesi gereken çeviriler olarak yüklemek için içe aktarma özelliğini kullanın. Bu, bir özet ya da benzer bir çeviri veri tabanı kullanan tek seferlik bir çeviri için en iyi yaklaşımdır.
- Kullandığınız tüm veri tabanlarına *tmserver* kurabilir ve Weblate tarafından kullanmasını sağlayabilirsiniz. Bu yöntem, çeviri sırasında birkaç kez kullanmak istediğinizde daha iyidir.
- Başka bir yöntem, ilgili tüm projeleri tek bir Weblate kopyasında çevirmektir. Bu da diğer projelerdeki çevirilerin de kendiliğinden alınmasını sağlar.

Ayrıca bakınız:

Kendiliğinden önerileri yapılandırmak, Kendiliğinden öneriler, Çeviri belleği

Weblate, çevirilerin yanında çeviri dosyalarını da günceller mi?

Weblate, çeviri dosyalarındaki değişiklikleri en aza indirmeye çalışır. Ne yazık ki bazı dosya biçimlerini yeniden biçimlendirebilir. Dosyanın biçimlendirmesini istediğiniz gibi korumak istiyorsanız, bir işleme öncesi kancası kullanın.

Ayrıca bakınız:

updating-target-files

Dil tanımları nereden gelir ve kendi dil tanımımı nasıl ekleyebilirim?

Temel dil tanımları kümesi Weblate ve Translate-toolkit içine eklenmiştir. Bu tanımlarda 150 üzerinde dil bulunur ve çoğul biçimler ya da metin yazma yönü hakkında bilgileri içerir.

Yönetim arayüzünde kendi dillerinizi tanımlamakta özgürsünüz. Yalnızca bu konuda bilgi vermeniz gerekiyor.

Ayrıca bakınız:

Dil tanımları

Weblate, belirsiz bir dizgedeki değişiklikleri vurgulayabilir mi?

Weblate bunu destekler, ancak farkı göstermek için verilere gerek duyar.

Gettext PO dosyaları için, PO dosyalarını güncellerken **msgmerge** komutuna `—previous` parametresini eklemeniz gerekir. Örneğin:

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

Tek dilli çeviriler için, Weblate önceki dizgeyi kimliğine göre bulabilir. Böylece farklılıkları kendiliğinden gösterir.

Kalıbı güncellediğimde Weblate neden hala eski çeviri dizgelerini gösteriyor?

Weblate, çeviri dosyalarını, çevirmenlerin çeviri yapmasına izin vermek dışında hiçbir şekilde değiştirmeye çalışmaz. Bu nedenle, kalıp ya da kaynak kodu değiştirildiğinde çevrilebilir dosyaları da güncellemez. Bunu el ile yapmanız ve değişiklikleri depoya göndermeniz yeterlidir. Daha sonra Weblate değişiklikleri kendiliğinden alır.

Not: Çeviri dosyalarını güncellemeden önce Weblate üzerinde yapılan değişiklikleri birleştirmek genellikle iyi bir fikirdir. Yoksa genellikle bazı birleştirilmesi gereken bazı çakışmalarla karşılaşsınız.

Örneğin, Gettext PO dosyalarıyla, **msgmerge** komutunu kullanarak çeviri dosyalarını güncelleyebilirsiniz:

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

Güncellenmenin kendiliğinden yapılmasını isterseniz, eklentiye kurabilirsiniz: `ref:addon-weblate.gettext.msgmerge`.

Ayrıca bakınız:

updating-target-files

Çeviri dosyaları nasıl yeniden adlandırılır?

Depodaki dosyaları yeniden adlandırırken, Weblate bunu dosyaların kaldırılması ve eklenmesi olarak görür. Bu işlem, dizge geçmişinin, yorumların ve önerilerin kaybedilmesine neden olabilir.

Bundan kaçınmak için, yeniden adlandırmayı şu şekilde yapın:

1. manage-vcs içinden çeviri bileşenini kilitleyin.
2. manage-vcs içinde bekleyen değişiklikleri gönderin.
3. Weblate değişikliklerini yukarı akış deposuna aktarın.
4. *Kancalar kullanılsın* ile kancalar üzerinden güncellemelerin alınmasını devre dışı bırakın.
5. Depodaki dosyaları yeniden adlandırın.
6. Bileşen yapılandırmasını yeni dosya adlarıyla eşleştirecek şekilde güncelleyin.
7. Güncelleme kancalarını etkinleştirin ve bileşenin kilidini açın.

1.9.3 Sorun çözme

İstekler bazen ``çok fazla dosya açık`` hatasıyla başarısız olur

Bu bazen Git deponuz çok fazla büyüdüğünde ve bu depolardan çok sayıda olduğunda olur. Git depolarının sıkıştırılması bu durumu iyileştirir.

Bunu yapmanın en kolay yolu şunu çalıştırmaktır:

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

Ayrıca bakınız:

DATA_DIR

Siteye erişirken “İstek hatalı (400)” hatasıyla karşılaşıyorum

Bunun nedeni büyük olasılıkla yanlış yapılandırılmış bir `ALLOWED_HOSTS` seçeneğidir. Weblate kopyanızdan erişmek istediğiniz tüm sunucu adlarının yazılmış olması gerekir. Örneğin:

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

Ayrıca bakınız:

Allowed hosts kurulumu

“Tek dil (en) için daha fazla dosya var” ne anlama geliyor?

Bu genellikle kaynak dil için çeviri dosyanız olduğunda olur. Weblate, kaynak dizgeleri izler ve bunun için kaynak dili ayırır. Aynı dil için ek dosya işlenmez.

- Kaynak dil için çeviri isteniyorsa, lütfen bileşen ayarlarındaki *Kaynak dil* seçeneğini değiştirin. Kaynak dili *İngilizce (Geliştirici)* olarak seçmek ya da *Kaynak dizgeleri için kalite aracı* kullanmak isteyebilirsiniz.
- Kaynak dil için çeviri dosyasına gerek yoksa, lütfen depodan kaldırın.
- Kaynak dil için çeviri dosyasına gerek duyulduğu, ancak Weblate tarafından yok sayılmasının gerektiği durumlarda, katılmaması için *Dil süzgeci* seçeneğini ayarlayın.

İpucu: Benzer hata iletisini diğer diller için de görebilirsiniz. Bu durumda en olası neden, birkaç dosyanın Weblate üzerinde tek bir dil ile eşleştirilmesidir.

Bu durum, eski dil kodlarının yenisiyle birlikte kullanılmasından (Japonca için `ja` ve `jp`) ya da hem ülkeye özgü hem de genel kodların (`fr` ve `fr_FR`) kullanılmasından kaynaklanabilir. Bilgi almak için: *Dil kodlarını işlemek*.

1.9.4 Özellikler

Weblate, Git ve Mercurial dışındaki diğer sürüm denetimi sistemlerini destekliyor mu?

Weblate şu anda *Git* (*GitHub çekme istekleri*, *Gerrit* ve *Subversion* için genişletilmiş destekle) ve *Mercurial* dışında hiçbir şey için yerleşik destek sağlamaz. Ancak diğer sürüm denetimi sistemleri için arka uçlar yazılabilir.

Diğer sürüm denetimi sistemlerine erişmek için Git üzerinde *Git uzak yardımcıları* komutunu da kullanabilirsiniz.

Weblate ayrıca sürüm denetimi sistemi olmadan çalışmayı da destekler. Bilgi almak için: *Yerel dosyalar*.

Not: Diğer sürüm denetimi sistemlerinin yerleşik desteği için, Weblate dağıtılmış sürüm denetimi sistemi kullanmalıdır ve büyük olasılıkla Git ve Mercurial dışındaki herhangi bir şeyle çalışacak şekilde ayarlanabilir. Ancak birisinin bu desteği eklemesi gerekir.

Ayrıca bakınız:

Sürüm denetimi bütünleştirilmesi

Weblate çevirmenlerin hesabını nasıl tutar?

Weblate üzerinde yapılan her değişiklik, çevirmenin adıyla sürüm denetimi sistemine işlenir. Böylece, her bir değişikliğin bir yazarı olur ve kod geliştirmek için kullandığınız standart sürüm denetimi sistemi araçlarını kullanarak bunu izleyebilirsiniz.

Ek olarak, çeviri dosyası biçimi tarafından desteklendiğinde, dosya üst bilgilerine çevirmenin adı eklenir.

Ayrıca bakınız:

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

Weblate neden tüm PO dosyalarını tek bir ağaçta göstermeyi dayatıyor?

Weblate, her PO dosyasının tek bir bileşen olarak görüntüleneceği şekilde tasarlanmıştır. Bu özellik, çevirmenler için yararlıdır. Böylece gerçekte neyi çevirdiklerini bilebilirler.

4.2 sürümünde değişti: Çevirmenler, bir projenin tüm bileşenlerini bir bütün olarak belirli bir dile çevirebilir.

Weblate neden sr_Latn ya da zh_Hant gibi dil kodlarını kullanıyor?

Bunlar, daha önce yanlış kullanılan değiştiriciler (@latin çeşitleri için) ya da ülke kodları (Çince için) yerine gerçekten farklı diller olduklarını daha iyi belirtmek için [RFC 5646](#) tarafından tanımlanmış dil kodlarıdır.

Weblate eski dil kodlarını anlamayı sürdürüyor ve bunları var olanlarla eşleştiriyor. Örneğin `sr@latin sr_Latn` ya da `zh@CN zh_Hans` olarak işlenir.

Not: Weblate varsayılan olarak alt çizgili POSIX biçemi dil kodlarını kullanır. Bilgi almak için: [Dil tanımları](#).

Ayrıca bakınız:

[Dil tanımları](#), [Dil kodu biçemi](#), `adding-translation`

1.10 Desteklenen dosya biçimleri

Weblate, çok çeşitli çeviri biçimlerini destekler. Her biçimin farklı yetenekler sağlayan farklı özellikleri vardır.

İpucu: Uygulamanız için bir dosya biçimi seçerken, kullandığınız araç takımında/platformda iyi çalışan bir biçime bağlı kalmak daha iyidir. Böylece, çevirmenleriniz alışkın oldukları araçları kullanabilir ve projenize katkıda bulunulma olasılığı yükselir.

1.10.1 Android dizge kaynakları

Uygulamaları çevirmek için özel Android dosya biçimi.

Android dizge kaynakları tek dillidir, *Tek dilli temel dil dosyası* diğer dosyalardan farklı bir konumda tutulur – `res/values/strings.xml`.

Not: Android *string-array* yapıları şu anda desteklenmiyor. Bu soruna çözmek için, dizge dizilerinizi birbirinden ayırabilirsiniz:

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

şuna dönüştür:

```
<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>
```

String bileşenlerini gösteren *string-array* farklı bir dosyada tutulmalı ve çevrilecek olarak sunulmamalıdır.

Bu betik, var olan dizgeler.xml dosyalarınızı ve çevirilerinizi önceden işlemenize yardımcı olabilir: <https://gist.github.com/paour/11291062>

İpucu: Bazı dizgelerin çevrilmesini önlemek için, bunlar çevrilemez olarak imlenebilir. Bu seçenek özellikle dizge başvuruları için kullanışlı olabilir:

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

Ayrıca bakınız:

Android dizge kaynakları belgeleri, Android string resources

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	res/values-*/strings.xml
Tek dilli temel dil dosyası	res/values/strings.xml
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Android dizgesi kaynağı</i>

1.10.2 Apple iOS dizgeleri

Dosya biçimi genellikle Apple:index:iOS <pair: iOS; translation> uygulamalarını çevirmek için kullanılır. Ancak PWG 5100.13 ile standartlaştırılır ve NeXTSTEP/OpenSTEP üzerinde kullanılır.

Apple iOS dizgeleri genellikle tek dilli çeviriler olarak kullanılır.

Ayrıca bakınız:

Stringsdict biçimi, Apple “dizge dosyaları” belgeleri, `PWG 5100.13 İleti Kataloğu Dosya Biçimi <<http://ftp.pwg.org/pub/pwg/candidates/cs-ippjobprinterext3v10-20120727-5100.13.pdf#page=66>>`, Mac OSX strings

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	Resources/*.lproj/Localizable.strings
Tek dilli temel dil dosyası	Resources/en.lproj/Localizable.strings ya da Resources/Base.lproj/Localizable.strings
Yeni çeviriler kalıbı	Boş
Dosya biçimi	iOS dizgeleri (UTF-8)

1.10.3 Uygulama mağazası üst veri dosyaları

3.5 sürümünde geldi.

Çeşitli uygulama mağazalarında uygulama yayınlamak için kullanılan üst veriler çevrilebilir. Şu anda şu araçlar uyumludur:

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

Üst veriler, Weblate tarafından çevrilmek üzere ayrı dizgeler olarak sunulacak birkaç metin dosyasından oluşur.

İpucu: Belirli dizgeleri (örneğin değişiklik günlükleri) çevirmek istemiyorsanız, bunları salt okunur olarak işaretleyin (bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)). Bu işlem, [Toplu düzenleme](#) ile kendiliğinden yapılabilir.

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	fastlane/android/metadata/*
Tek dilli temel dil dosyası	fastlane/android/metadata/en-US
Yeni çeviriler kalıbı	fastlane/android/metadata/en-US
Dosya biçimi	App store üst veri dosyaları

1.10.4 ARB dosyası

4.1 sürümünde geldi.

ARB çevirileri tek dillidir. Bu nedenle (en sık kullanılan) İngilizce dizgeleri bulunan bir temel dosya belirtmeniz önerilir.

Ayrıca bakınız:

[JSON](#), [Uygulama kaynaklarını paketleme teknik özellikleri](#), [Flutter uygulamalarını yerelleştirme](#), [updating-target-files](#), [JSON çıktısını özelleştirme](#), [Çeviri dosyalarını temizleme](#)

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	lib/l10n/intl_*.arb
Tek dilli temel dil dosyası	lib/l10n/intl_en.arb
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>ARB dosyası</i>

1.10.5 CSV dosyaları

CSV dosyaları basit bir kaynak ve çeviri listesi içerebilir. Weblate şu dosyaları destekler:

- Üst bilgi olarak tanımlayıcı alanları bulunan dosyalar (`location`, `source`, `target`, `ID`, `fuzzy`, `context`, `translator_comments`, `developer_comments`). En az hataya eğilimli olduğundan bu yaklaşım önerilir. Dosya biçimi olarak *CSV dosyası* seçin.
- İki alanı olan dosyalar: Kaynak ve çeviri (bu sırayla). Dosya biçimi olarak `:guilabel:` Basit CSV dosyası seçin.
- [Translate-toolkit](#) tarafından sıralaması şöyle tanımlanmış alanlarla üst bilgileri olmayan dosyalar: `location`, `source`, `target`, `ID`, `fuzzy`, `context`, `translator_comments`, `developer_comments`. Dosya biçimi olarak *CSV dosyası* seçin.
- Dosyalarınız tek dilli olduğunda *Tek dilli temel dil dosyası* tanımlamayı unutmayın (bilgi almak için: *İki dilli ve tek dilli biçimler*).

İpucu: Varsayılan olarak, CSV biçimi dosya kodlamasının kendiliğinden algılanmasını sağlar. Bu davranış, bazı uç durumlarda güvenilir olamaz ve başarımın düşmesine yol açabilir. Bunu önlemek için lütfen kodlamalı dosya biçimi çeşidini seçin (*CSV dosyası (UTF-8)* gibi).

Uyarı: CSV biçimi şu anda CSV dosyasının lehçesini otomatik olarak algılar. Bazı durumlarda kendiliğinden algılama yapılamayabilir ve karışık sonuçlar alırsınız. Bu, özellikle değerlerde yeni satırlar bulunan CSV dosyaları için geçerlidir. Geçici bir çözüm olarak, alıntı karakterlerini atlamanız önerilir.

Ayrıca bakınız:

CSV

Çok değerli CSV dosyası

4.13 sürümünde geldi.

CSV dosyalarının bu çeşidi, bir dizge için birden çok çevirinin tutulmasını sağlar.

Örnek dosyalar

Örnek dosya:

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

Weblate yapılandırması

İki dilli CSV için tipik Weblate <i>Bileşen yapılandırması</i> dosyası	
Dosya maskesi	locale/*.csv
Tek dilli temel dil dosyası	<i>Boş</i>
Yeni çeviriler kalıbı	locale/en.csv
Dosya biçimi	<i>CSV dosyası</i>

Tek dilli CSV için tipik Weblate <i>Bileşen yapılandırması</i> dosyası	
Dosya maskesi	locale/*.csv
Tek dilli temel dil dosyası	locale/en.csv
Yeni çeviriler kalıbı	locale/en.csv
Dosya biçimi	<i>Basit CSV dosyası</i>

1.10.6 DTD dosyaları

Ayrıca bakınız:

[Mozilla DTD format](#)

Örnek DTD dosyası:

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

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	locale/*.dtd
Tek dilli temel dil dosyası	locale/en.dtd
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>DTD dosyası</i>

1.10.7 Excel Open XML

3.2 sürümünde geldi.

Excel Açık XML (.xlsx) dosyaları içe ve dışa aktarılabilir.

XLSX dosyalarını çeviri için yüklerken, yalnızca etkin çalışma sayfasının dikkate alındığını ve en azından `source` adlı bir sütun (kaynak dizgenin bulunduğu) ve `target` adlı bir sütun (çevirinin bulunduğu) olması gerektiğini unutmayın. Ek olarak, `context` adlı bir sütun (çeviri dizgesinin bağlam yolunun bulunduğu) olmalıdır. Çevirileri bir Excel çalışma kitabı olarak dışa aktarmak için XLSX indirmesini kullanırsanız, doğru biçimde bir dosya alırsınız.

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	path/*.xlsx
Tek dilli temel dil dosyası	path/en.xlsx
Yeni çeviriler kalıbı	path/en.xlsx
Dosya biçimi	<i>Excel Open XML</i>

1.10.8 Düz XML dosyaları

3.9 sürümünde geldi.

Ayrıca bakınız:

[Flat XML](#)

Düz XML dosyası örneği:

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

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	locale/*.xml
Tek dilli temel dil dosyası	locale/en.xml
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Düz XML dosyası</i>

1.10.9 Fluent biçimi

4.8 sürümünde geldi.

Not: Bu biçimin desteği şu anda geliştirme aşamasında. Bazı özellikler henüz kullanılamıyor olabilir ve davranışları sürümler arasında değişiklik gösterebilir. Denemelerinizle ilgili geri bildirimlerinizi bekliyoruz.

Fluent, asimetrik yerelleştirmeye odaklanmış tek dilli bir metin biçimidir. Bir dildeki basit bir dizge, başka bir dildeki karmaşık bir çok değişkenli çeviriyle eşleştirilebilir.

Ayrıca bakınız:

[Fluent projesi sitesi](#)

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	locales/*/messages.ftl
Tek dilli temel dil dosyası	locales/en/messages.ftl
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Fluent dosyası</i>

1.10.10 GNU gettext

Özgür yazılımları çevirmek için en yaygın olarak kullanılan biçim.

Dosyada depolanan bağlamsal bilgiler, üst bilgileri ayarlanarak veya ilgili kaynak dosyalarla bağlantı kurularak sağlanabilir.

Ayrıca bakınız:

../devel/gettext, ../devel/sphinx, [Wikipedi üzerinde Gettext, PO Files](#), “*configure*” dosyasındaki *ALL_LINGUAS* değişkeni güncelleme, *Gettext çıktısını özelleştirme*, *LINGUAS* dosyasını güncelleme, *MO* dosyaları üretme, *PO* dosyalarını *POT* ile eşleştirerek güncelleme (*msgmerge*)

Tek dilli gettext

Bazı projeler tek dilli biçimler için gettext kullanmaya karar verir. Yalnızca kaynak kodlarındaki kimlikleri kodlarlar ve dizgenin İngilizce ile birlikte tüm dillere çevrilmesi gerekir. Bu yöntem desteklenir, ancak bileşenleri Weblate içine aktarırken açıkça bu dosya biçimini seçmeniz gerekir.

Örnek dosyalar

İki dilli Gettext PO dosyası genellikle şöyle görünür:

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

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

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

Tek dilli Gettext PO dosyası genellikle şöyle görünür:

```
#: 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"
```

Temel dil dosyası şöyle olacaktır:

```
#: 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"
```

Weblate yapılandırması

İki dilli gettext için tipik Weblate <i>Bileşen yapılandırması</i> dosyası	
Dosya maskesi	po/* .po
Tek dilli temel dil dosyası	<i>Boş</i>
Yeni çeviriler kalıbı	po/messages .pot
Dosya biçimi	<i>Gettext PO dosyası</i>

Tek dilli gettext için tipik Weblate <i>Bileşen yapılandırması</i> dosyası	
Dosya maskesi	po/* .po
Tek dilli temel dil dosyası	po/en .po
Yeni çeviriler kalıbı	po/messages .pot
Dosya biçimi	<i>Gettext PO dosyası (tek dilli)</i>

1.10.11 go-i18n JSON dosyaları

4.1 sürümünde geldi.

4.16 sürümünde değişti: Bu biçimin 2. sürümü için destek eklendi.

go-i18n çevirileri tek dillidir. Bu nedenle (en sık kullanılan) İngilizce dizgeleri bulunan bir temel dosya belirtmeniz önerilir.

Not: Weblate, go-i18n JSON 1. ve 2. sürümlerini destekler. Lütfen ortamınıza uyan doğru dosya biçimini seçin.

Ayrıca bakınız:

JSON, go-i18n, updating-target-files, *JSON çıktısını özelleştirme*, *Çeviri dosyalarını temizleme*,

Weblate yapılandırması

V1 için tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	langs/* .json
Tek dilli temel dil dosyası	langs/en .json
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>go-i18n v1 JSON dosyası</i>

V2 için tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	langs/* .json
Tek dilli temel dil dosyası	langs/en .json
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>go-i18n v2 JSON dosyası</i>

1.10.12 gotext JSON dosyaları

4.15.1 sürümünde geldi.

gotext çevirileri tek dillidir. Bu nedenle (en sık kullanılan) İngilizce dizgeleri bulunan bir temel dosya belirtmeniz önerilir.

Ayrıca bakınız:

[JSON](#), [Go ile I18n: Çeviri yönetimi](#), [updating-target-files](#), [JSON çıktısını özelleştirme](#), [Çeviri dosyalarını temizleme](#),

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	internal/translations/locales/*/messages.gotext.json
Tek dilli temel dil dosyası	internal/translations/locales/en-GB/messages.gotext.json
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>gotext JSON dosyası</i>

1.10.13 GWT özellikleri

Çeviriler için yerel GWT biçimi.

GWT özellikleri genellikle tek dilli çeviriler olarak kullanılır.

Ayrıca bakınız:

[GWT yerelleştirme rehberi](#), [GWT yerelleştirme eğitimi](#), [Mozilla and Java properties files](#), [updating-target-files](#), [Java özellikleri dosyasını biçimlendirme](#), [Çeviri dosyalarını temizleme](#)

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	src/app/Bundle_*.properties
Tek dilli temel dil dosyası	src/app/Bundle.properties
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>GWT özellikleri</i>

1.10.14 HTML dosyaları

4.1 sürümünde geldi.

Çevrilebilir içerik HTML dosyalarından ayıklanır ve çevrilmek üzere sunulur.

Ayrıca bakınız:

[HTML](#)

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	path/*.html
Tek dilli temel dil dosyası	path/en.html
Yeni çeviriler kalıbı	path/en.html
Dosya biçimi	HTML dosyası

1.10.15 i18next JSON dosyaları

4.15.1 sürümünde değişti: Bu biçimin 4. sürümü için destek eklendi.

İpucu: Çoğul biçimleri kullanıyorsanız, CLDR ile uyumlu olarak işlenmesi için 4. sürümü kullanmanız önerilir. Eski sürümlerde, bazı diller için doğru olmayan farklı çoğul kuralları var.

i18next, JavaScript ile ve JavaScript için yazılmış bir yerelleştirme çatısıdır. Weblate, bu yerelleştirme dosyalarını çoğul biçimler gibi özelliklerle destekler.

i18next çevirileri tek dillidir. Bu nedenle (en sık kullanılan) İngilizce dizgeleri olan bir temel dosya belirtmeniz önerilir.

Not: Weblate, *i18next* JSON 3. ve 4. sürümlerini destekler. Lütfen ortamınıza uyan doğru dosya biçimini seçin.

The v2 and v1 variants are mostly compatible with v3, with exception of how plurals are handled.

Ayrıca bakınız:

[JSON](#), [i18next JSON biçimi](#), [updating-target-files](#), [JSON çıktısını özelleştirme](#), [Çeviri dosyalarını temizleme](#)

Örnek dosya:

```
{
  "hello": "Hello",
  "apple": "I have an apple",
  "apple_plural": "I have {{count}} apples",
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
"apple_negative": "I have no apples"
}
```

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	langs/*.json
Tek dilli temel dil dosyası	langs/en.json
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	i18next JSON dosyası v3

1.10.16 IDML biçimi

4.1 sürümünde geldi.

Çevrilebilir içerik, Adobe InDesign markup dili dosyalarından ayıklanır ve çevrilmek üzere sunulur.

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	path/*.idml
Tek dilli temel dil dosyası	path/en.idml
Yeni çeviriler kalıbı	path/en.idml
Dosya biçimi	<i>IDML dosyası</i>

1.10.17 INI çevirileri

4.1 sürümünde geldi.

Çeviriler için INI dosya biçimi. Weblate şu anda JSON çevirilerinin birkaç çeşidini destekler:

- Tek dilli INI dosyaları
- *Joomla! çevirileri*
- *Inno Setup INI çevirileri*

INI çevirileri genellikle tek dilli çeviriler olarak kullanılır.

Not: Weblate, yalnızca bir INI dosyasındaki bölümlerden anahtarları ayıklar. INI dosyanızda bölümler yoksa, bunun yerine *Joomla! çevirileri* ya da *Java özellikleri* kullanmak isteyebilirsiniz.

Ayrıca bakınız:

INI Files, *Java özellikleri*, *Joomla! çevirileri*, *Inno Setup INI çevirileri*

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	language/*.ini
Tek dilli temel dil dosyası	language/en.ini
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>INI dosyası</i>

1.10.18 Inno Setup INI çevirileri

4.1 sürümünde geldi.

Çeviriler için Inno Setup INI dosya biçimi.

Inno Setup INI çevirileri genellikle tek dilli çeviriler olarak kullanılır.

Not: *INI çevirileri* ile arasındaki dikkate değer tek fark, satır sonu için %n ve sekme için %t kodlarının desteklenmesidir.

Not: Şu anda yalnızca Unicode dosyaları (.isl) destekleniyor. ANSI çeşidi (.isl) şu anda desteklenmiyor.

Ayrıca bakınız:

[INI Files](#), [Joomla! çevirileri](#), [INI çevirileri](#)

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	language/*.isl
Tek dilli temel dil dosyası	language/en.isl
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Inno Setup INI dosyası</i>

1.10.19 Java özellikleri

Çeviriler için doğal Java biçimi.

Java özellikleri genellikle tek dilli çeviriler olarak kullanılır.

Weblate, bu biçimin ISO-8859-1, UTF-8 ve UTF-16 çeşitlerini destekler. Hepsi tüm Unicode karakterlerinin tutulmasını destekler, yalnızca farklı şekilde kodlanmıştır. ISO-8859-1 içinde, Unicode kaçış dizileri kullanılır (zkou\u0161ka gibi), diğerlerinde karakterler doğrudan UTF-8 ya da UTF-16 olarak kodlanır.

Not: Kaçış dizilerinin yüklenmesi UTF-8 kipinde de çalışır. Bu nedenle lütfen uygulama gereksinimlerinize uygun doğru kodlama kümesini seçerken dikkatli olun.

Ayrıca bakınız:

[Vikipedi üzerinde Java özellikleri](#), [Mozilla and Java properties files](#), [mi18n lang dosyaları](#), [GWT özellikleri](#), [updating-target-files](#), [Java özellikleri dosyasını biçimlendirme](#), [Çeviri dosyalarını temizleme](#)

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	src/app/Bundle_*.properties
Tek dilli temel dil dosyası	src/app/Bundle.properties
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Java Properties (ISO-8859-1)</i>

1.10.20 Joomla! çevirileri

Çeviriler için doğal Joomla! biçimi.

Joomla çevirileri genellikle tek dilli çeviriler olarak kullanılır.

Ayrıca bakınız:

Mozilla and Java properties files, *INI çevirileri*, *Inno Setup INI çevirileri*

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	language/*/com_foobar.ini
Tek dilli temel dil dosyası	language/en-GB/com_foobar.ini
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Joomla dil dosyası</i>

1.10.21 JSON dosyaları

4.3 sürümünde değişti: JSON dosyasının yapısı, önceki sürümlerde bozulmaya yol açan karmaşık durumlar için bile uygun şekilde korunur.

JSON biçimi çoğunlukla JavaScript içine eklenmiş uygulamaları çevirmek için kullanılır.

Weblate şu anda JSON çevirilerinin birkaç çeşidini destekler:

- *vue-i18n* ya da *react-intl* gibi basit anahtar / değer dosyaları tarafından kullanılır.
- İç içe anahtarları olan dosyalar.
- *i18next JSON dosyaları*
- *go-i18n JSON dosyaları*
- *gotext JSON dosyaları*
- *WebExtension JSON*
- *ARB dosyası*

JSON çevirileri genellikle tek dillidir. Bu nedenle (en sık kullanılan) İngilizce dizgeleri bulunan bir temel dosya belirtmeniz önerilir.

İpucu: *JSON dosyası* ve *JSON iç içe geçmiş yapı dosyası* aynı tür dosyaları işleyebilir. İkisi de çeviri yaparken var olan JSON yapısını korur.

Aralarındaki tek fark, Weblate kullanarak yeni dizgeler eklerken. İç içe geçmiş yapı biçimi yeni eklenen anahtarı ayrıştırır ve yeni dizgeyi eşleşen yapıya ekler. Örneğin `app.name` anahtarı şöyle eklenir:

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

Ayrıca bakınız:

JSON, updating-target-files, JSON çıktısını özelleştirme, Çeviri dosyalarını temizleme,

Örnek dosyalar

Örnek dosya:

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

İç içe geçmiş dosyalar da desteklenir (gereksinimler için yukarıya bakın). Böyle bir dosya şöyle görünebilir:

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

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	langs/translation-*.json
Tek dilli temel dil dosyası	langs/translation-en.json
Yeni çeviriler kalıbı	Boş
Dosya biçimi	JSON iç içe geçmiş yapı dosyası

1.10.22 Laravel PHP dizgeleri

4.1 sürümünde değişti.

Laravel PHP yerelleştirme dosyaları çoğul biçimler ile desteklenir:

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

Ayrıca bakınız:

PHP, Laravel ile yerelleştirme

1.10.23 Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	lang/*/texts.php
Tek dilli temel dil dosyası	lang/en/texts.php
Yeni çeviriler kalıbı	lang/en/texts.php
Dosya biçimi	Laravel PHP dizgeleri

1.10.24 mi18n lang dosyaları

4.7 sürümünde geldi.

JavaScript yerelleştirmesi için kullanılan dosya biçimi *mi18n*. Sözdizimsel olarak *Java özellikleri* ile eşleşir.

Ayrıca bakınız:

mi18n, Mozilla and Java properties files, *Java özellikleri*, updating-target-files, *Java özellikleri dosyasını biçimlendirme*, *Çeviri dosyalarını temizleme*

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	*.lang
Tek dilli temel dil dosyası	en-US.lang
Yeni çeviriler kalıbı	Boş
Dosya biçimi	<i>mi18n lang dosyası</i>

1.10.25 OpenDocument biçimi

4.1 sürümünde geldi.

Çevrilebilir içerik OpenDocument dosyalarından ayıklanır ve çevrilmek üzere sunulur.

Ayrıca bakınız:

OpenDocument Format

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	path/*.odt
Tek dilli temel dil dosyası	path/en.odt
Yeni çeviriler kalıbı	path/en.odt
Dosya biçimi	<i>OpenDocument dosyası</i>

1.10.26 PHP dizgeleri

PHP çevirileri genellikle tek dillidir. Bu nedenle (en sık kullanılan) İngilizce dizgeleri bulunan bir temel dosya belirtmeniz önerilir.

Weblate şu anda PHP çevirilerinin birkaç çeşidini destekler:

- Çeşitli söz dizimlerinde tek dilli PHP dizgeleri
- *Laravel PHP dizgeleri*

Örnek dosya:

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

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	lang/*/texts.php
Tek dilli temel dil dosyası	lang/en/texts.php
Yeni çeviriler kalıbı	lang/en/texts.php
Dosya biçimi	<i>PHP dizgeleri</i>

1.10.27 Qt Linguist .ts

Qt temelli uygulamalarda kullanılan çeviri biçimi.

Qt Linguist dosyaları iki dilde ve tek dilde çeviriler için kullanılabilir.

Ayrıca bakınız:

Qt Linguist rehberi, Qt .ts, *İki dilli ve tek dilli biçimler*

Weblate yapılandırması

İki dilli olarak kullanıldığında tipik Weblate bileşeni	
Dosya maskesi	i18n/app.*.ts
Tek dilli temel dil dosyası	<i>Boş</i>
Yeni çeviriler kalıbı	i18n/app.de.ts
Dosya biçimi	<i>Qt Linguist çeviri dosyası</i>

Tek dilli olarak kullanıldığında tipik Weblate bileşeni	
Dosya maskesi	i18n/app.*.ts
Tek dilli temel dil dosyası	i18n/app.en.ts
Yeni çeviriler kalıbı	i18n/app.en.ts
Dosya biçimi	<i>Qt Linguist çeviri dosyası</i>

1.10.28 ResourceDictionary dosyaları

4.13 sürümünde geldi.

ResourceDictionary, Windows Presentation Foundation (WPF) uygulamaları için yerelleştirilebilir dizge kaynaklarını paketlemek için kullanılan tek dilli bir XML dosya biçimidir.

Ayrıca bakınız:

Flat XML, *Düz XML dosyaları*, updating-target-files, *Çeviri dosyalarını temizleme*

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	Languages/* .xaml
Tek dilli temel dil dosyası	Language/en .xaml
Yeni çeviriler kalıbı	Boş
Dosya biçimi	ResourceDictionary dosyası

1.10.29 RESX .NET kaynak dosyaları

.XML kaynağı (.resx) dosyası, Microsoft .NET uygulamalarında kullanılan tek dilli XML dosya biçimini kullanır. `.resx` <<https://lingohub.com/developers/resource-files/resw-resx-localization>> ile aynı söz dizimini kullanırken `.resw` ile değiştirilebilir`_.

Ayrıca bakınız:

.NET Resource files (.resx), updating-target-files, *Çeviri dosyalarını temizleme*

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	Resources/Language.* .resx
Tek dilli temel dil dosyası	Resources/Language .resx
Yeni çeviriler kalıbı	Boş
Dosya biçimi	.NET kaynak dosyası

1.10.30 Ruby YAML dosyaları

Kök düğüm olarak dil ile Ruby i18n YAML dosyaları.

Ayrıca bakınız:

YAML, *YAML dosyaları*

Ruby i18n YAML dosyası örneği:

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

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	translations/messages.*.yaml
Tek dilli temel dil dosyası	translations/messages.en.yaml
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Ruby YAML dosyası</i>

1.10.31 Stringsdict biçimi

4.8 sürümünde geldi.

Apple tarafından kullanılan ve bir dizgenin çoğul biçimlerini tutabilen XML tabanlı biçim.

Ayrıca bakınız:

Apple iOS dizgeleri, Stringsdict dosya biçimi

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	Resources/*.lproj/Localizable.stringsdict
Tek dilli temel dil dosyası	Resources/en.lproj/Localizable.stringsdict ya da Resources/Base.lproj/Localizable.stringsdict
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Stringsdict dosyası</i>

1.10.32 Alt yazı dosyaları

3.7 sürümünde geldi.

Weblate çeşitli alt yazı dosyalarını çevirebilir:

- SubRip alt yazı dosyası (*.srt)
- MicroDVD alt yazı dosyası (*.sub)
- Advanced Substation Alpha alt yazı dosyası (*.ass)
- Substation Alpha alt yazı dosyası (*.ssa)

Ayrıca bakınız:

Subtitles

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	path/*.srt
Tek dilli temel dil dosyası	path/en.srt
Yeni çeviriler kalıbı	path/en.srt
Dosya biçimi	<i>SubRip alt yazı dosyası</i>

1.10.33 TermBase eXchange biçimi

4.5 sürümünde geldi.

TBX, terminoloji verilerinin alış verişinde kullanılan bir XML biçimidir.

Ayrıca bakınız:

[Vikipedi üzerinde TBX](#), [TBX](#), [Sözlük](#)

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	tbx/*.tbx
Tek dilli temel dil dosyası	<i>Boş</i>
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>TermBase eXchange dosyası</i>

1.10.34 Metin dosyaları

4.6 sürümünde geldi.

Çevrilebilir içerik düz metin dosyalarından ayıklanır ve çevrilmek üzere sunulur. Her paragraf ayrı bir dizge olarak çevrilir.

Bu biçimin üç çeşidi vardır:

- Düz metin dosyası
- DokuWiki metin dosyası
- MediaWiki metin dosyası

Ayrıca bakınız:

[Simple Text Documents](#)

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	path/*.txt
Tek dilli temel dil dosyası	path/en.txt
Yeni çeviriler kalıbı	path/en.txt
Dosya biçimi	<i>Düz metin dosyası</i>

1.10.35 WebExtension JSON

Mozilla Firefox ya da Google Chromium uzantılarını çevirirken kullanılan dosya biçimi.

Not: Bu biçim JSON olarak adlandırılrsa da, teknik özellikleri JSON teknik özelliklerinin dışında kalan yorumların kullanılmasını da sağlar. Weblate şu anda yorumların bulunduğu dosya türünü desteklemiyor.

Ayrıca bakınız:

JSON, Google chrome.i18n, Mozilla Eklenti Yerelleştirmesi

Örnek dosya:

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

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	<code>_locales/*/messages.json</code>
Tek dilli temel dil dosyası	<code>_locales/en/messages.json</code>
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>WebExtension JSON dosyası</i>

1.10.36 Windows RC dosyaları

4.1 sürümünde değişti: Windows RC dosyaları desteği yeniden yazıldı.

Not: Bu biçimin desteği şu anda geliştirme aşamasında. Bazı özellikler henüz kullanılamıyor olabilir ve davranışları sürümler arasında değişiklik gösterebilir. Denemelerinizle ilgili geri bildirimlerinizi bekliyoruz.

RC dosyaları, Windows uygulamaları için çevrilebilir metinleri, iletişim pencerelerini ve menüleri yerelleştirmek için kullanılan dil dosyalarıdır.

Ayrıca bakınız:

Windows RC files

Örnek dosyalar

Windows RC dosyası örneği:

```
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
```

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	lang/*.rc
Tek dilli temel dil dosyası	lang/en-US.rc
Yeni çeviriler kalıbı	lang/en-US.rc
Dosya biçimi	RC dosyası

1.10.37 XLIFF

XML temelli biçim, çeviri dosyalarını standartlaştırmak için oluşturulmuştur. Ancak sonuçta bu alandaki <https://xkcd.com/927/> birçok standarttan biridir.

XML yerelleştirme değişim dosyası biçimi (XLIFF) genellikle iki dilli olarak kullanılır. Ancak Weblate bunun tek dilli olarak kullanılmasını da destekler.

Weblate, birkaç çeşitte XLIFF desteği sunar:

XLIFF çeviri dosyası

Bileşenlerin içeriğinin düz metin olarak tutulduğu basit XLIFF dosyası (tüm XML bileşenlerinden kaçış).

Yerleştirilebilir destekli XLIFF

Yerleştirilebilir öğeleri ve diğer XML öğelerini destekleyen standart XLIFF.

gettext uzantılı XLIFF

XLIFF, çoğul biçimleri desteklemek için Gettext PO için XLIFF 1.2 Gösterim Rehberi ile zenginleştirilmiştir.

Ayrıca bakınız:

XML yerelleştirme değişim dosyası biçimi (XLIFF) teknik özellikleri, Gettext PO için XLIFF 1.2 gösterim rehberi, Vikipedi üzerinde XLIFF, XLIFF,

Çeviri durumları

3.3 sürümünde değişti: Weblate, 3.3 sürümünden önce `state` özniteliğini yok sayıyordu.

Dosyadaki `state` özniteliği kısmen işlenir ve Weblate üzerindeki `Düzenlenmesi` gerekli durumuyla eşleştirilir (şu durumlar, bir hedef varsa dizgeyi düzenlenmesi gerekli olarak imlemek için kullanılır: `new`, `needs-translation`, `needs-adaptation`, `needs-l10n`). `state` özniteliği eksikse, bir dizge bir `<target>` bileşeni var olur olmaz çevrilmiş olarak değerlendirilir.

Çeviri dizgesinde `approved = "yes"` varsa, Onaylanmış olarak Weblate içine aktarılır. Diğer her şey *Onaylanmayı bekliyor* (XLIFF teknik özellikleriyle eşleşen) olarak içe aktarılır.

Kaydedilirken, Weblate gerekli olmadıkça bu öznitelikleri eklemeyiz:

- `state` özniteliği yalnızca dizgenin düzenlenmesi gerekli olarak imlenmesi durumunda eklenir.
- Onaylanmış özniteliği yalnızca dizgenin onaylanması durumunda eklenir.
- Diğer durumlarda öznitelikler eklenmez. Ancak var olmaları durumunda güncellenirler.

XLIFF biçimini kullanırken, dizgelerin onaylanmış durumunu görmek ve değiştirmek için Weblate onaylama sürecini etkinleştirmeniz önemle önerilir.

Benzer şekilde, bu tür dosyaları içe aktarırken (yükleme formunda), *Düzenlenmesi gerekli dizgelere yapılacak işlem* olarak *Çeviri olarak eklensin* seçeneğini seçmelisiniz.

Ayrıca bakınız:

Özel onaylayanlar

XLIFF dosyasında boşluk ve yeni satırlar

XML biçimleri arasında boşluk türleri ya da miktarları genellikle ayırt edilmez. Korumak istiyorsanız, dizgeye `xml:space = "preserve"` imini eklemeniz gerekir.

Örnek:

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

Çeviri imlerini belirtmek

`weblate-flags` özniteliğini kullanarak ek çeviri imleri belirtebilirsiniz (bilgi almak için: *Parametreleri kullanarak davranışı özelleştirmek*). Weblate ayrıca XLIFF Teknik özelliklerindeki `maxwidth` ve `font` özniteliklerini de anlar:

```
<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>
```

font özniteliği yazı tipi ailesi, boyutu ve ağırlığı için işlenir. Yukarıdaki örnekte bunların tümünü görebilirsiniz. Ancak yalnızca yazı tipi ailesi gereklidir. Yazı tipi ailesindeki herhangi bir boşluk alt çizgiye dönüştürülür. Bu nedenle Source Sans Pro, Source_Sans_Pro olur. Lütfen yazı tipi grubunu adlandırırken bunu aklınızda bulundurun (bilgi almak için: [Yazı tiplerini yönetmek](#)).

Ayrıca bakınız:

[XLIFF 1.2 font özniteliği](#), [XLIFF 1.2 maxwidth özniteliği](#)

Dizge anahtarları

Weblate, XLIFF dosyasındaki birimleri, var olması durumunda `resname` özniteliğiyle tanımlar ve `id` kullanmaya döner (varsa `file` kod imi ile birlikte).

`resname` özniteliğinin, birimin insan dostu tanımlayıcısı olması ve Weblate üzerinde `id` yerine görüntülenmesini daha uygun biçime getirmesi beklenir. `resname`, tüm XLIFF dosyasında benzersiz olmalıdır. Bu durum Weblate için gereklidir ve XLIFF standardı kapsamında değildir. Öznitelige herhangi bir benzersizlik kısıtlaması getirmez.

Weblate yapılandırması

İki dilli XLIFF için tipik Weblate bileşeni	
Dosya maskesi	<code>localizations/*.xliff</code>
Tek dilli temel dil dosyası	<i>Boş</i>
Yeni çeviriler kalıbı	<code>localizations/en-US.xliff</code>
Dosya biçimi	<i>XLIFF çeviri dosyası</i>

Tek dilli XLIFF için tipik Weblate bileşeni	
Dosya maskesi	<code>localizations/*.xliff</code>
Tek dilli temel dil dosyası	<code>localizations/en-US.xliff</code>
Yeni çeviriler kalıbı	<code>localizations/en-US.xliff</code>
Dosya biçimi	<i>XLIFF çeviri dosyası</i>

1.10.38 YAML dosyaları

Dizge anahtarları ve değerleri bulunan düz YAML dosyaları. Weblate ayrıca listelerden ya da sözlüklerden dizgeleri ayıklar.

Weblate şu anda YAML çevirilerinin birkaç çeşidini destekler:

- İç içe anahtarları olan dosyalar.
- *Ruby YAML dosyaları*

Ayrıca bakınız:

[YAML](#), [Ruby YAML dosyaları](#)

YAML dosyası örneği:

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

Weblate yapılandırması

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	translations/messages.*.yml
Tek dilli temel dil dosyası	translations/messages.en.yml
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>YAML dosyası</i>

Ayrıca bakınız:

Translation Related File Formats

1.10.39 Kendiliğinden algılama

Weblate, *Çeviri projelerini ve bileşenleri eklemek* sırasında dosya biçimini algılamaya çalışır. Aynı serileştirme biçiminin (JSON, YAML, özellikler) ya da dosya kodlamasının farklı çeşitleri yanlış algılanabilir. Bu nedenle lütfen bileşeni oluşturmadan önce *Dosya biçimi* değerinin doğru olduğundan emin olun.

1.10.40 Çeviri türü yetenekleri

Tablo 1: Desteklenen tüm biçimlerdeki yetenekler

Bişim	Dil bilgisi ^{Sayfa 102, 1}	Çoğullar ^{Sayfa 102, 2}	Açıklamalar ^{Sayfa 102, 3}	Bağlam ^{Sayfa 102, 4}	Ko-num ^{Sayfa 102, 5}	İmler ^{Sayfa 102, 8}	Ek durumlar ^{Sayfa 102, 6}
<i>GNU gettext</i>	iki dilde	evet	evet	evet	evet	evet ⁹	düzenlenmesi gerekli
<i>Tek dilli gettext</i>	tek	evet	evet	evet	evet	evet ⁹	düzenlenmesi gerekli
<i>XLIFF</i>	ikisi de	evet	evet	evet	evet	evet	düzenlenmesi gerekli, onaylanmış
<i>Java özellikleri</i>	ikisi de	hayır	evet	hayır	hayır	hayır	
<i>mi18n lang dosyaları</i>	tek	hayır	evet	hayır	hayır	hayır	
<i>GWT özellikleri</i>	tek	evet	evet	hayır	hayır	hayır	
<i>Joomla! çevirileri</i>	tek	hayır	evet	hayır	evet	hayır	
<i>Qt Linguist .ts</i>	ikisi de	evet	evet	hayır	evet	evet	düzenlenmesi gerekli
<i>Android dizge kaynakları</i>	tek	evet	evet ⁷	hayır	hayır	evet	
<i>Apple iOS dizgeleri</i>	ikisi de	hayır	evet	hayır	hayır	hayır	
<i>PHP dizgeleri</i>	tek	hayır ¹⁰	evet	hayır	hayır	hayır	

sonraki sayfaya devam

Tablo 1 – önceki sayfadan devam

Bişim	Dil bilgi- si ^{Sayfa 102, 1}	Çoğullar ²	Açıklama- lar ³	Bağlam ⁴	Konum ⁵	İmler ⁸	Ek durum- lar ⁶
<i>JSON dosyaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>i18next JSON dosyaları</i>	tek	evet	hayır	hayır	hayır	hayır	
<i>go-i18n JSON dosyaları</i>	tek	evet	evet	hayır	hayır	hayır	
<i>gotext JSON dosyaları</i>	tek	evet	evet	hayır	evet	hayır	
<i>ARB dosyası</i>	tek	evet	evet	hayır	hayır	hayır	
<i>WebExtension JSON</i>	tek	evet	evet	hayır	hayır	hayır	
<i>RESX .NET kaynak dosyaları</i>	tek	hayır	evet	hayır	hayır	evet	
<i>Resource-Dictionary dosyaları</i>	tek	hayır	hayır	hayır	hayır	evet	
<i>CSV dosyaları</i>	ikisi de	hayır	evet	evet	evet	hayır	düzen- lenmesi gerekli
<i>YAML dosyaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>Ruby YAML dosyaları</i>	tek	evet	hayır	hayır	hayır	hayır	
<i>DTD dosyaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>Düz XML dosyaları</i>	tek	hayır	hayır	hayır	hayır	evet	
<i>Windows RC dosyaları</i>	tek	hayır	evet	hayır	hayır	hayır	
<i>Excel Open XML</i>	tek	hayır	evet	evet	evet	hayır	düzen- lenmesi gerekli
<i>Uygulama mağazası üst veri dosyaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>Alt yazı dosyaları</i>	tek	hayır	hayır	hayır	evet	hayır	
<i>HTML dosyaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>OpenDocument biçimi</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>IDML biçimi</i>	tek	hayır	hayır	hayır	hayır	hayır	

sonraki sayfaya devam

Tablo 1 – önceki sayfadan devam

B biçim	Dil bilgi- si ¹	Çoğullar ²	Açıklama- lar ³	Bağlam ⁴	Konum ⁵	İmler ⁸	Ek durum- lar ⁶
<i>INI çeviri- leri</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>Inno Setup INI çeviri- leri</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>TermBase eXchange biçimi</i>	iki dilde	hayır	evet	hayır	hayır	evet	
<i>Metin dos- yaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>Stringsdict biçimi</i>	tek	evet	evet	hayır	hayır	hayır	
<i>Fluent biç- mi</i>	tek	hayır ¹¹	evet	hayır	hayır	hayır	

İki dilli ve tek dilli biçimler

Hem tek dilli hem de iki dilli biçimler desteklenir. İki dilli biçimler iki dili tek bir dosyada tutar. Kaynak ve çeviri (tipik örnekler *GNU gettext*, *XLIFF* ya da *Apple iOS dizgeleri*). Diğer taraftan, tek dilli biçimler dizgeyi kimliğine göre tanımlar ve her dil dosyasında yalnızca bunların belirli bir dille eşleşmesi bulunur (genellikle *Android dizge kaynakları*). Bazı dosya biçimleri her iki çeşit için de kullanılır. Aşağıda ayrıntılı açıklamayı bulabilirsiniz.

Tek dilli dosyaların doğru kullanımı için Weblate, kaynaklarıyla çevrilecek dizgelerin tam listesinin bulunduğu bir dosyaya erişmelidir. Bu dosyaya Weblate içinde *Tek dilli temel dil dosyası* adı verilir. Ancak adlandırma yapınızda bu ad değişebilir.

Ayrıca, bu iş akışı, geliştiriciler tarafından sağlanan dizgeleri kapsayacak şekilde *Ara dil dosyası* kullanılarak genişletilebilir. Ancak son dizgelerde olduğu gibi kullanılmamalıdır.

Çeviri imleri

Çeviri imleri, Weblate davranışının özelleştirilmesini sağlar. Bazı biçimler, imlerin çeviri dosyasında tanımlanmasını destekler (Weblate arayüzünden her zaman tanımlayabilirsiniz. Bilgi almak için: `:ref: custom-checks`).

Bu özellik, *GNU gettext* içindeki imlerde modellenmiştir.

İmler, tüm XML tabanlı biçimler için standart olmayan `weblate-flags` özniteliğinden ayıklanır. Ek olarak, `max-length:N`, XLIFF standardında tanımlandığı gibi `maxwidth` özniteliği ile desteklenir. Bilgi almak için: *Çeviri imlerini belirtmek*.

Ayrıca bakınız:

Parametreleri kullanarak davranışı özelleştirmek, *PO dosyaları belgeleri*

¹ Bilgi almak için: *İki dilli ve tek dilli biçimler*

² Bilgi almak için: *Çoğul biçimli dizgeler*

³ Kaynak dizge açıklamaları, çevrilecek dizge ile ilgili ek bilgiler vermek için kullanılabilir.

⁴ Bağlam, farklı kapsamlarda kullanılan özdeş dizgeleri ayırt etmek için kullanılır (örneğin, İngilizce *Sun*, “Pazar” gününün kısaltılmış biçimi olarak ya da bize en yakın yıldızın adı olarak kullanılabilir).

⁵ Bir dizgenin kaynak kodundaki konumu, yetkin çevirmenlerin dizgenin nasıl kullanıldığını anlamalarına yardımcı olabilir.

⁸ Bilgi almak için: *Çeviri imleri*.

⁶ “Çevrilmemiş” ve “Çevrilmiş” durumlarına ek olarak dosya biçimi tarafından desteklenen ek durumlar.

⁹ Gettext türü yorumlar im olarak kullanılır.

⁷ `<string>` bileşeninden önce yerleştirilmiş XML açıklaması, kaynak dizge açıklaması olarak işlenir.

¹⁰ Çoğul biçimler yalnızca bunları tanımlamak için dizge söz diziminde kullanan Laravel için desteklenir. Bilgi almak için: `Laravel ile yerelleştirme`.

¹¹ Çoğul biçimler, dizgelerin söz diziminde işlenir ve Weblate üzerinde çoğul olarak gösterilmez.

Çoğul biçimli dizgeler

Çoğul biçimler, değişken sayısı olan dizgeleri doğru yerelleştirmek için gereklidir. Kurallar hedef dile bağlıdır ve birçok biçim bunun için CLDR teknik özelliklerini kullanılır.

İpucu: Çoğul biçimli dizgeler uygulama çatısı tarafından da desteklenmelidir. Platformunuza uygun *GNU gettext*, `:ref:'aresource'` ya da `:ref:'stringsdict'` yerel biçimini seçin.

Salt okunur dizgeler

3.10 sürümünde geldi.

Çeviri dosyalarındaki salt okunur dizgeler de alınır. Ancak Weblate üzerinde düzenlenemez. Bu özellik yerel olarak birkaç biçim (*XLIFF* ve *Android dizge kaynakları*) tarafından desteklenir. Ancak salt okunur imi eklenerek diğerlerinde taklit edilebilir. Bilgi almak için: *Parametreleri kullanarak davranışı özelleştirmek*.

1.10.41 Diğer biçimlerin desteği

Serileştirmeyi destekleyen *translate-toolkit* tarafından desteklenen çoğu biçim kolayca desteklenebilir. Ancak (henüz) herhangi bir deneme yapılmamıştır. Çoğu durumda, farklı *translate-toolkit* depolarının davranış farklılıklarını gizlemek için Weblate üzerinde bazı ince katmanlar gereklidir.

Yeni bir biçim desteği eklemek için yeğlenen yaklaşım, önce *translate-toolkit* içine bu biçim için desteği eklemektir.

Ayrıca bakınız:

Translation Related File Formats

1.11 Sürüm denetimi bütünleştirmesi

Weblate şu anda *Git* (gelişmiş destek ile *GitHub çekme istekleri*, *GitLab birleştirme istekleri*, *Gitea çekme isteği*, *Gerrit*, *Subversion* ve *Bitbucket sunucusu çekme isteği*) ve *Mercurial* sürüm denetimi sistemlerini destekliyor.

1.11.1 Depolara erişmek

Kullanmak istediğiniz sürüm denetimi deposuna Weblate üzerinden erişilebiliyor olması gerekir. Herkese açık olan bir depoda doğru adresi yazmanız yeterlidir (<https://github.com/WeblateOrg/weblate.git> gibi). Ancak gizli depolar ya da itme adresleri için kurulum daha karmaşıktır ve kimlik doğrulaması gerekir.

Hosted Weblate üzerinden depolara erişmek

Hosted Weblate için GitHub, Bitbucket, Codeberg ve GitLab üzerinde kayıtlı özel bir itme kullanıcısı bulunur (kullanıcı adı *weblate*, e-posta adresi hosted@weblate.org ve adı *Weblate push user* olan). Bu kullanıcıyı katılımcı olarak eklemeniz ve deponuza uygun izni vermeniz gerekir (kopyalama için salt okunur izni uygundur, itme için yazma izni gereklidir). Hizmete ve kuruluş ayarlarınıza bağlı olarak, bu işlem hemen yapılır ya da Weblate tarafında onay gerekir.

GitHub üzerindeki *weblate* kullanıcısı davetleri beş dakika içinde kendiliğinden kabul eder. Diğer hizmetlerde işlemi el ile yapmak gerekebilir, bu nedenle lütfen sabırlı olun.

weblate kullanıcısı eklendikten sonra, SSH iletişim kuralını kullanarak *Kaynak kod deposu* ve *Depo itme adresi* değerlerini yapılandırabilirsiniz (örneğin `git@github.com:WeblateOrg/weblate.git`).

SSH depoları

Gizli depolara erişmek için en sık kullanılan yöntem SSH kullanmaktır. Herkese açık Weblate SSH anahtarını (bilgi almak için: [Weblate SSH anahtarı](#)) yukarı akış deposuna bu şekilde erişmesi için yetkilendirin.

Uyarı: GitHub üzerinde her anahtar yalnızca bir kez kullanılabilir. Bilgi almak için: [GitHub depoları](#) ve [Hosted Weblate üzerinden depolara erişmek](#).

Weblate ayrıca ilk bağlantıda sunucu anahtarının parmak izini saklar ve daha sonra değişmesi durumunda sunucu ile bağlantı kuramaz (bilgi almak için: [SSH sunucu anahtarlarını doğrulamak](#)).

Ayarlama yapılması gerektiğinde, bunu Weblate yönetim arayüzünden yapın:

Public RSA 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
AAAAB3NzaC1yc2EAAAADAQABAAQDg1iQ7xGwqmOPyvPZhLIDtwM/5jP4PZEqsPU5Da1u+SY6hnT+MJRqAyf9EXctGqHvKjHW7MHdPu+JliTZDKIhQ9A6RebGqEF
Weblate
```

[Download RSA private key](#)

Generate Ed25519 SSH key

You don't seem to have existing Ed25519 SSH key, by pressing button below Weblate will generate one for you.

[Generate new Ed25519 SSH key](#)

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-ed25519	+DIY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeiOtrVc98/R1BUFWu3/LiyKgUfQM
github.com	ssh-rsa	uNiVztksCsDhcc0u9e8BujQXVUpKZIDTMczCvj3tD2s

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

Weblate SSH anahtarı

4.17 sürümünde değişti: Weblate artık hem RSA hem de Ed25519 SSH anahtarları üretebiliyor. Yeni kurulumlar için Ed25519 kullanılması önerilir.

Tüm kullanıcılar herkese açık Weblate anahtarını, *Hakkında* sayfasında görebilir.

Yöneticiler, yönetim arayüzü açılış sayfasında (*SSH anahtarları* bölümünden) Weblate tarafından bağlantıda kullanılmakta olan herkese açık anahtarı oluşturabilir veya görüntüleyebilir.

Not: İlgili kişisel SSH anahtarının şu anda bir parolası olamaz. Bu nedenle iyi korunduğundan emin olun.

İpucu: Oluşturulan kişisel Weblate SSH anahtarının yedeğini alın.

SSH sunucu anahtarlarını doğrulamak

Weblate, SSH sunucu anahtarlarını ilk kez eriştiğinde kendiliğinden depolar ve sonraki kullanımlar için hatırlar.

Depoya bağlanmadan önce anahtar parmak izini doğrulamak isterseniz, erişeceğiniz sunucuların SSH sunucu anahtarlarını yönetim arayüzünün aynı bölümünden *Sunucu anahtarı ekle* ile ekleyin. Erişeceğiniz sunucu adını yazın (`gitlab.com` gibi) ve *Gönder* üzerine basın. Parmak izinin eklediğiniz sunucuyla eşleştiğini doğrulayın.

Parmak izli eklenen anahtarlar onay iletilisinde görüntülenir:

Weblate
 Dashboard Projects Languages Checks

Manage / SSH keys

Added host key for github.com with fingerprint uNiVztkCsDhcc0u9e8BujQXVUpKZIDTMczCvj3tD2s (ssh-rsa), please verify that it is correct.

Added host key for github.com with fingerprint p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM (ecdsa-sha2-nistp256), please verify that it is correct.

Added host key for github.com with fingerprint +DiY3wvV6TuJJhbpZisF/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 RSA 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
AAAAB3NzaC1yc2EAAAADAQABAAQDg1iQ7xGwqmOPyvPZhLIDtwM/5jP4PZEqsPU5Da1u+SY6hnT+MjRqAyf9EXctGqHvKjHW7MHdPu+JliTZDK1hQ9A6RebGqEf
Weblate
```

Download RSA private key

Generate Ed25519 SSH key

You don't seem to have existing Ed25519 SSH key, by pressing button below Weblate will generate one for you.

Generate new Ed25519 SSH key

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-ed25519	+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM
github.com	ssh-rsa	uNiVztkCsDhcc0u9e8BujQXVUpKZIDTMczCvj3tD2s

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

GitHub depoları

SSH üzerinden erişilebilir (bilgi almak için: [SSH depoları](#)). Ancak birden fazla depoya erişmeniz gerekirse, izin verilen SSH anahtarı kullanımında uyarısı ile bir GitHub sınırlamasıyla karşılaşsınız (her anahtar yalnızca bir kez kullanılabilir).

İme işleminin yapılacağı dal ayarlanmamışsa, proje dallandır ve değişiklikler bir daldan itilir. Ayarlanmış ise, değişiklikler yukarı akış deposuna ve seçilen dala itilir.

Daha küçük dağıtımlar için, kişisel erişim belirteci ve GitHub hesabınızla HTTPS kimlik doğrulaması kullanın. Bilgi almak için: [Komut satırı kullanımı için erişim belirteci oluşturmak](#).

Daha büyük kurulumlar için, Weblate için özel bir kullanıcı oluşturmak, Weblate üzerinde oluşturulan genel SSH anahtarını atamak (bilgi almak için: [Weblate SSH anahtarı](#)) ve çevirmek istediğiniz tüm depolara erişim izni vermek genellikle daha iyidir. Bu yaklaşım Hosted Weblate için de kullanılır. Bunun için özel bir *weblate* kullanıcısı vardır.

Ayrıca bakınız:

[Hosted Weblate üzerinden depolara erişmek](#)

İç Weblate adresleri

Bir depo kurulumunu, diğer (bağlantılı) bileşenlerde `weblate://project/component` olarak konumlandırılmasına atıfta bulunarak farklı bileşenler arasında paylaşın. Bu şekilde bağlantılı bileşenler, ana (başvurulan) bileşenin sürüm denetimi sistemi deposu yapılandırmasını kullanır.

Uyarı: Ana bileşenin kaldırılması, bağlantılı bileşenleri de kaldırır.

Weblate, eşleşen bir depo kurulumu olan bir bileşen bulursa, bileşen oluştururken depo adresini kendiliğinden ayarlar. Bileşen yapılandırmasının son adımında bunu değiştirebilirsiniz.

Bunun kullanılma nedenleri:

- Sunucuda daha az disk alanı kullanır, depo yalnızca bir kez kaydedilir.
- Güncellemelerin daha hızlı yapılmasını sağlar, yalnızca bir depo güncellenir.
- Weblate çevirilerinin bulunduğu yalnızca tek bir depo dışa aktarılır (bilgi almak için: [Git dışa aktarıcı](#)).
- Bazı eklentiler bir depoyu paylaşan birden fazla bileşen üzerinde çalışabilir. Örneğin [Git işlemlerini bir araya toplar](#).

HTTPS depoları

Korunmuş HTTPS depolarına erişmek için adrese kullanıcı adını ve parolayı ekleyin. Endişelenmeyin, Weblate, adresi kullanıcılara görüntülerken (depo adresinin görülmesine izin verilse bile) bu bilgileri çıkarır.

Örneğin, kimlik doğrulaması eklenmiş GitHub adresi şöyle görünebilir: `https://kullanıcı:erişim_kodunuz@github.com/WeblateOrg/weblate.git`.

Not: Kullanıcı adınızda ya da parolanızda özel karakterler varsa, bunların URL olarak kodlanması gerekir. Örneğin `https://kullanıcı%40örnek.com:%24parola%23@bitbucket.org/...`.

Vekil sunucu kullanmak

Bir vekil sunucu kullanarak HTTP/HTTPS sürüm denetimi sistemi depolarına erişmeniz gerekiyorsa, sürüm denetimi sistemini bunu kullanacak şekilde yapılandırın.

Bunun için, `http_proxy`, `https_proxy` ve `all_proxy` ortam değişkenlerini kullanın (bilgi almak için: [cURL belgeleri](#)) ya da sürüm denetimi sistemi yapılandırmasından dayatın. Örneğin:

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

Not: Vekil sunucu yapılandırmasının Weblate çalıştıran kullanıcı ile (ayrıca bilgi almak için: [Dosya sistemi izinleri](#)) ve `HOME = $DATA_DIR/home` (bilgi almak için: [DATA_DIR](#)) yolunda yapılması gerekir. Yoksa Weblate tarafından yürütülen Git bunu kullanmaz.

Ayrıca bakınız:

[cURL rehberi](#), [Git yapılandırma belgeleri](#)

1.11.2 Git

İpucu: Weblate için Git 2.12 ya da daha yeni sürümü gereklidir.

Ayrıca bakınız:

Farklı türde depolara nasıl erişileceği ile ilgili bilgi almak için: [Depolara erişmek](#).

Git (itme dayatması ile)

Bu tam olarak Git gibi davranır. Tek fark her zaman itmenin dayatılmasıdır. Yalnızca çeviriler için ayrı bir depo kullanılması durumunda seçilmesi amaçlanmıştır.

Uyarı: Dikkatli kullanın, çünkü kolayca yukarı akış deponuzda eksik işlemlere yol açar.

Git yapılandırmasını özelleştirmek

Weblate, tüm sürüm denetimi sistemi komutlarını `HOME = $DATA_DIR/home` ile çağırır (bilgi almak için [DATA_DIR](#)). Bu nedenle kullanıcı yapılandırması `DATA_DIR/home/.git` içinde düzenlenmelidir.

Git uzak yardımcıları

Ek olarak diğer sürüm denetim sistemlerini desteklemek için Git [remote helpers](#) kullanabilirsiniz. Ancak bunun yol açabileceği sorunları çözmeye hazır olmalısınız.

Şu anda, Bazaar ve Mercurial yardımcıları GitHub üzerindeki ayrı depolarda bulunabilir: [git-remote-hg](#) ve [git-remote-bzr](#). Bunları el ile indirin ve arama yolunuzda bir yere koyun (`~/bin` gibi). İlgili sürüm denetim sistemlerinin kurulmuş olduğundan emin olun.

Kurduktan sonra, bu uzak yardımcıları Weblate üzerinde bir depo belirtmek için kullanılabilir.

Bazaar kullanarak Launchpad üzerindeki `gnuselam` projesini kopyalamak için:

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

Mercurial kullanarak `selenic.com` üzerindeki `selam` deposu için:


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

Uyarı: Git uzak yardımcılarını kullanmanın zorluğu, örneğin Mercurial üzerinde, uzak yardımcının değişiklikleri geri iterken yeni bir ipucu oluşturmastır.

1.11.3 GitHub çekme istekleri

Bu yöntem, *Git* üzerine çeviri değişikliklerini doğrudan depoya itmek yerine *GitHub API* ile çekme istekleri olarak itmeyi sağlayan ince bir katman ekler.

Git değişiklikleri doğrudan bir depoya iterken, *GitHub çekme istekleri* çekme istekleri oluşturur. İkincisi, yalnızca *Git* depolarına erişmek için gerekmez.

Bunun çalışması için Weblate ayarlarında API kimlik doğrulama bilgilerini (*GITHUB_CREDENTIALS*) yapılandırmanız gerekir. Yapılandırdığınızda, *Sürüm denetimi sistemi* seçerken *GitHub* seçeneğini göreceksiniz.

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri itmek, *GITHUB_CREDENTIALS*

1.11.4 GitLab birleştirme istekleri

3.9 sürümünde geldi.

Bu yöntem, *Git* üzerine çeviri değişikliklerini doğrudan depoya itmek yerine, *GitLab API* ile birleştirme istekleri olarak itmeyi sağlayan ince bir katman ekler.

Git depolarına erişmek için bunu kullanmak gerekmez. Sıradan *Git* aynı şekilde çalışır. Tek fark bir depoya itme işleminin nasıl yapıldığıdır. *Git* ile değişiklikler doğrudan depoya itilirken, *GitLab birleştirme istekleri* birleştirme isteği oluşturur.

Bunun çalışması için Weblate ayarlarından API kimlik doğrulama bilgilerini (*GITLAB_CREDENTIALS*) yapılandırmanız gerekir. Yapılandırdığınızda sonra, *Sürüm denetimi sistemi* seçerken *GitLab* seçeneğini göreceksiniz.

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri itmek, *GITLAB_CREDENTIALS*

1.11.5 Gitea çekme isteği

4.12 sürümünde geldi.

Bu yöntem, *Git* üzerine çeviri değişikliklerini doğrudan depoya itmek yerine *Gitea API* ile çekme istekleri olarak itmeyi sağlayan ince bir katman ekler.

Git depolarına erişmek için bunu kullanmak gerekmez. Sıradan *Git* aynı şekilde çalışır. Tek fark bir depoya itme işleminin nasıl yapıldığıdır. *Git* ile değişiklikler doğrudan depoya itilirken, *Gitea çekme isteği* birleştirme isteği oluşturur.

Bunun çalışması için Weblate ayarlarında API kimlik doğrulama bilgilerini (*GITEA_CREDENTIALS*) yapılandırmanız gerekir. Yapılandırdığınızda, *Sürüm denetimi sistemi* seçerken *Gitea* seçeneğini göreceksiniz.

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri itmek, *GITEA_CREDENTIALS*

1.11.6 Bitbucket sunucusu çekme isteği

4.16 sürümünde geldi.

Bu yöntem, [Git](#) üzerine çeviri değişikliklerini doğrudan depoya itmek yerine [Bitbucket sunucu API](#) ile çekme istekleri olarak itmeyi sağlayan ince bir katman ekler.

Uyarı: Bu yöntemde, Bitbucket Cloud API desteği yoktur.

Git depolarına erişmek için bunu kullanmak gerekmez. Sıradan [Git](#) aynı şekilde çalışır. Tek fark bir depoya itme işleminin nasıl yapıldığıdır. [Git](#) ile değişiklikler doğrudan depoya itilirken, [Bitbucket sunucusu çekme isteği](#) birleştirme isteği oluşturur.

Bunun çalışması için Weblate ayarlarında API kimlik doğrulama bilgilerini ([BITBUCKETSERVER_CREDENTIALS](#)) yapılandırmanız gerekir. Yapılandırdığınızda, [Sürüm denetimi sistemi](#) seçerken [Bitbucket sunucusu](#) seçeneğini göreceksiniz.

Ayrıca bakınız:

[Weblate üzerindeki değişiklikleri itmek](#), [BITBUCKETSERVER_CREDENTIALS](#)

1.11.7 Pagure birleştirme istekleri

4.3.2 sürümünde geldi.

Bu yöntem, [Git](#) üzerine çeviri değişikliklerini doğrudan depoya itmek yerine, [Pagure API](#) ile birleştirme istekleri olarak itmeyi sağlayan ince bir katman ekler.

Git depolarına erişmek için bunu kullanmak gerekmez. Sıradan [Git](#) aynı şekilde çalışır. Tek fark bir depoya itme işleminin nasıl yapıldığıdır. [Git](#) ile değişiklikler doğrudan depoya itilirken, [Pagure birleştirme istekleri](#) birleştirme isteği oluşturur.

Bunun çalışması için Weblate ayarlarında API kimlik doğrulama bilgilerini ([PAGURE_CREDENTIALS](#)) yapılandırmanız gerekir. Yapılandırdığınızda, [Sürüm denetimi sistemi](#) seçerken [Pagure](#) seçeneğini göreceksiniz.

Ayrıca bakınız:

[Weblate üzerindeki değişiklikleri itmek](#), [PAGURE_CREDENTIALS](#)

1.11.8 Gerrit

Bu yöntem [git-review](#) aracını kullanarak, [Git](#) üzerine çeviri değişikliklerini doğrudan depoya itmek yerine Gerrit onaylama istekleri olarak itmeyi sağlayan ince bir katman ekler.

Gerrit belgelerinde, bu tür depoları kurmak için gerekli yapılandırma bilgilerini bulabilirsiniz.

1.11.9 Mercurial

Mercurial, doğrudan Weblate içinden kullanabileceğiniz başka bir sürüm denetimi sistemidir.

Not: Herhangi bir Mercurial sürümüyle çalışmalıdır. Ancak bazen komut satırı arayüzünde Weblate bütünleştirme-sini bozan uyumsuz değişiklikler olabilir.

Ayrıca bakınız:

Farklı türde depolara nasıl erişileceği ile ilgili bilgi almak için: [Depolara erişmek](#).

1.11.10 Subversion

Weblate, [subversion](#) depolarıyla etkileşim kurmak için [git-svn](#) kullanır. Subversion depolarının bir Git istemcisi tarafından kullanılması ve kullanıcıların iç deponun tam bir kopyasını alarak yerelde işleme yapabilmesi bir Perl betiği tarafından sağlanır.

Not: Weblate, Subversion depo düzenini kendiliğinden algılamayı dener. Hem dal için doğrudan adresleri hem de standart düzendeki depoları (branches/ , tags/ ve trunk/) destekler. Bu konuda bilgi almak için [git-svn belgeleri](#) bölümüne bakabilirsiniz. Deponuzun düzeni standart değilse ve hatalarla karşılaşırsanız, dal adını depo adresine eklemeyi ve dalı boş bırakmayı deneyin.

Subversion kimlik doğrulama bilgileri

Weblate, sertifikayı (ve gerekirse kimlik doğrulama bilgilerinizi) önceden onaylamış olmanızı bekler. Bunları `DATA_DIR` klasörüne eklemeye çalışır. `$HOME` ortam değişkeninin `DATA_DIR` olarak ayarlanmış olduğunu denetleyin ve `svn` kullanarak sertifikayı onaylayın:

```
# 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
```

Ayrıca bakınız:

`DATA_DIR`

1.11.11 Yerel dosyalar

1.11.12 Git

İpucu: Bunun altında [Git](#) kullanılır. Git kurulu olmalıdır. Tüm geçmişleriyle birlikte çevirileriniz için doğal biçimde Git kullanmaya geçmenizi sağlar.

3.8 sürümünde geldi.

Weblate, uzak sürüm denetimi sistemi olmadan da çalışabilir. İlk çeviriler yüklenerek içe aktarılır. Daha sonra tek tek dosyalar dosya yükleme ile değiştirebilir ya da doğrudan Weblate üzerinden çeviri dizgeleri eklenebilir (şu anda yalnızca tek dilli çeviriler için kullanılabilir).

Arka planda Weblate sizin için bir Git deposu oluşturur ve tüm değişiklikler izlenir. Daha sonra çevirileri depolamak için bir sürüm denetimi sistemi kullanmaya karar verirsiniz, Bütünleştirmenizde kullanabileceğiniz deponuz Weblate içinde hazır olur.

1.12 Weblate REST API uygulaması

API erişimi için `/api/` adresi kullanılabilir. API [Django REST framework](#) üzerine geliştirilmiştir. Doğrudan ya da `:ref:`wlc`` ile kullanabilirsiniz.

1.12.1 Kimlik doğrulama ve genel parametreler

Herkese açık proje API uygulaması kimlik doğrulaması olmadan kullanılabilir. Ancak kimliği doğrulanmamış istekler yoğun şekilde kısıtlanır (varsayılan olarak günde 100 istek). Bu nedenle kimlik doğrulamasının kullanılması önerilir. Kimlik doğrulamasında, profilinizden alabileceğiniz bir kod kullanılır. Bunu Yetkilendirme üst bilgisinde kullanın:

ANY /

API için genel istek davranışı, buradaki üst bilgiler, durum kodları ve parametreler tüm uç noktalara da uygulanır.

Sorgu parametreleri

- **format** – Yanıt biçimi (**Accept** değiştirir). Olası değerler REST çatısının kurulumuna bağlıdır. Varsayılan olarak `json` ve `api` desteklenir. İkincisi, API için tarayıcı arayüzünü sağlar.
- **page** – Sayfalanmış sonuçların belirtilen sayfasını görüntüler (gezinmeyi otomatikleştirmek için yanıtta *next* ve *previous* alanlarını kullanın).

İstek üst bilgileri

- **Accept** – Yanıt içerik türü **Accept** üst bilgisine bağlıdır
- **Authorization** – Kimlik doğrulaması için **Authorization: Token BELİRTECİNİZ** olarak isteğe bağlı belirteç

Yanıt üst bilgileri

- **Content-Type** – bu parametre, isteğin **Accept** üst bilgisine bağlıdır
- **Allow** – Nesne için izin verilen HTTP yöntemlerinin listesi

Yanıt JSON nesnesi

- **detail** (*string*) – Sonucun ayrıntılı açıklaması (**200 OK** dışındaki HTTP durum kodları için)
- **count** (*int*) – Nesne listelerindeki toplam öge sayısı
- **next** (*string*) – Nesne listeleri için sonraki sayfa adresi
- **previous** (*string*) – Nesne listeleri için önceki sayfa adresi
- **results** (*array*) – Nesne listelerinin sonuçları
- **url** (*string*) – API ile bu kaynağa erişme adresi
- **web_url** (*string*) – Tarayıcı ile bu kaynağa erişme adresi

Durum kodları

- **200 OK** – istek düzgün olarak işlendiğinde
- **201 Created** – yeni bir nesne sorunsuz oluşturulduğunda
- **204 No Content** – bir nesne sorunsuz silindiğinde
- **400 Bad Request** – form parametreleri eksik olduğunda
- **403 Forbidden** – erişim reddedildiğinde
- **429 Too Many Requests** – hız sınırlaması uygulandığında

Kimlik doğrulama kodları

4.10 sürümünde değişti: Proje kapsamlı kodlar 4.10 sürümünde tanıtıldı.

Her kullanıcı için, kullanıcı profilinden öğrenilebilecek bir kendi kişisel erişim kodu bulunur. Yeni oluşturulan kullanıcı kodları `wlu_` ön ekini taşır.

Yalnızca belirli bir projeye API erişimi sağlamak için proje kapsamlı kodlar oluşturulabilir. Bu belirteçler `wlp_` ön ekiyle anlaşılabilir.

Kimlik doğrulama örnekleri

Örnek istek:

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

Örnek yanıt:

```
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/"
}
```

CURL örneği:

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

Parametre aktarma örnekleri

POST yöntemi için parametreler form gönderimi (*application/x-www-form-urlencoded*) ya da JSON (*application/json*) olarak belirtilebilir.

Form isteği örneği:

```
POST /api/projects/hello/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Authorization: Token TOKEN

operation =pull
```

JSON isteği örneği:

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

{"operation": "pull"}
```

CURL örneği:

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

CURL JSON örneği:

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

API hız sınırlaması

API isteklerinin hızı sınırlanmıştır; Varsayılan yapılandırmada anonim kullanıcılar için günde 100 istek ve kimliği doğrulanmış kullanıcılar için saatte 5000 istek sınırı bulunur.

Hız sınırlaması `settings.py` dosyasından ayarlanabilir. Nasıl yapılandırılacağı ile ilgili ayrıntılı bilgi almak için [Throttling in Django REST framework documentation](#) bölümüne bakabilirsiniz.

Docker kapsayıcısında hız sınırlaması `WEBLATE_API_RATELIMIT_ANON` ve `WEBLATE_API_RATELIMIT_USER` seçenekleri kullanılarak yapılandırılabilir.

Hız sınırlamasının durumu şu üst bilgiler ile bildirilir:

X-RateLimit-Limit	Yapılacak isteklerin hız sınırlama sınırı
X-RateLimit-Remaining	Kalan istek sınırı
X-RateLimit-Reset	Saniye olarak hız sınırlama penceresinin sıfırlanma süresi

4.1 sürümünde değişti: Eklenen hız sınırlaması durumu üst bilgileri.

Ayrıca bakınız:

[Hızı sınırlamak](#), [Hızı sınırlamak](#), `WEBLATE_API_RATELIMIT_ANON`, `WEBLATE_API_RATELIMIT_USER`

1.12.2 API giriş noktası

GET /api/

API kök giriş noktası.

Örnek istek:

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

Örnek yanıt:

```

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 Kullanıcılar

4.0 sürümünde geldi.

GET /api/users/

Kullanıcıları yönetme izniniz varsa kullanıcı listesini görüntüler. Yoksa, yalnızca kendi bilgilerinizi görüntüler.

Ayrıca bakınız:

Kullanıcı nesnesi öznitelikleri `GET /api/users/(str:username)/` bölümünde bulunabilir.

POST /api/users/

Yeni bir kullanıcı oluşturur.

Parametreler

- **username** (*string*) – Kullanıcı adı
- **full_name** (*string*) – Kullanıcının tam adı
- **email** (*string*) – Kullanıcının e-posta adresi
- **is_superuser** (*boolean*) – Kullanıcı süper kullanıcı mı? (isteğe bağlı)
- **is_active** (*boolean*) – Kullanıcı etkin mi? (isteğe bağlı)
- **is_bot** (*boolean*) – Kullanıcı bot mu? (isteğe bağlı) (proje kapsamındaki belirteçler için kullanılır)

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

Kullanıcılar hakkındaki bilgileri görüntüler.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı

Yanıt JSON nesnesi

- **username** (*string*) – bir kullanıcının kullanıcı adı
- **full_name** (*string*) – bir kullanıcının tam adı
- **email** (*string*) – bir kullanıcının e-posta adresi
- **is_superuser** (*boolean*) – kullanıcının süper kullanıcı olup olmadığı
- **is_active** (*boolean*) – kullanıcının etkin olup olmadığı
- **is_bot** (*boolean*) – kullanıcının bot olup olmadığı (proje kapsamındaki belirteçler için kullanılır)

- **date_joined** (*string*) – kullanıcının oluşturulduğu tarih
- **groups** (*array*) – ilişkili grup bağlantıları. Bilgi almak için: `GET /api/groups/(int:id)/`

Örnek JSON verileri:

```
{
  "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) /

Kullanıcı parametrelerini değiştirir.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı

Yanıt JSON nesnesi

- **username** (*string*) – bir kullanıcının kullanıcı adı
- **full_name** (*string*) – bir kullanıcının tam adı
- **email** (*string*) – bir kullanıcının e-posta adresi
- **is_superuser** (*boolean*) – kullanıcının süper kullanıcı olup olmadığı
- **is_active** (*boolean*) – kullanıcının etkin olup olmadığı
- **is_bot** (*boolean*) – kullanıcının bot olup olmadığı (proje kapsamındaki belirteçler için kullanılır)
- **date_joined** (*string*) – kullanıcının oluşturulduğu tarih

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

Kullanıcı parametrelerini değiştirir.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı

Yanıt JSON nesnesi

- **username** (*string*) – bir kullanıcının kullanıcı adı
- **full_name** (*string*) – bir kullanıcının tam adı
- **email** (*string*) – bir kullanıcının e-posta adresi
- **is_superuser** (*boolean*) – kullanıcının süper kullanıcı olup olmadığı
- **is_active** (*boolean*) – kullanıcının etkin olup olmadığı
- **is_bot** (*boolean*) – kullanıcının bot olup olmadığı (proje kapsamındaki belirteçler için kullanılır)
- **date_joined** (*string*) – kullanıcının oluşturulduğu tarih

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

Tüm kullanıcı bilgilerini siler ve kullanıcıyı etkin değil olarak imler.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı

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

Grupları bir kullanıcı ile ilişkilendirir.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı

Form parametreleri

- **string group_id** – Benzersiz grup kimliği

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

4.13.1 sürümünde geldi.

Kullanıcıyı bir gruptan kaldır.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı

Form parametreleri

- **string group_id** – Benzersiz grup kimliği

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

Bir kullanıcının istatistiklerini görüntüle.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı

Yanıt JSON nesnesi

- **translated** (int) – Kullanıcının yaptığı çeviri sayısı
- **suggested** (int) – Kullanıcının yaptığı öneri sayısı
- **uploaded** (int) – Kullanıcının yaptığı yükleme sayısı
- **commented** (int) – Kullanıcının yaptığı yorum sayısı
- **languages** (int) – Kullanıcının çevirebileceği dil sayısı

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

Bir kullanıcının aboneliklerini listeler.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı

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

Abonelikleri bir kullanıcıyla ilişkilendirir.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı

JSON nesnesi ister

- **notification** (string) – Kayıtlı bildirimin adı
- **scope** (int) – Var olan seçeneklerden bildirim kapsamı
- **frequency** (int) – Bildirimler için sıklık seçenekleri

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

Bir kullanıcıyla ilişkilendirilmiş bir aboneliği alır.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı
- **subscription_id** (*int*) – Kayıtlı bildirim kimliği

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

Bir kullanıcıyla ilişkili aboneliği düzenler.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı
- **subscription_id** (*int*) – Kayıtlı bildirim kimliği

JSON nesnesi ister

- **notification** (*string*) – Kayıtlı bildirimin adı
- **scope** (*int*) – Var olan seçeneklerden bildirim kapsamı
- **frequency** (*int*) – Bildirimler için sıklık seçenekleri

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

Bir kullanıcıyla ilişkili aboneliği düzenler.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı
- **subscription_id** (*int*) – Kayıtlı bildirim kimliği

JSON nesnesi ister

- **notification** (*string*) – Kayıtlı bildirimin adı
- **scope** (*int*) – Var olan seçeneklerden bildirim kapsamı
- **frequency** (*int*) – Bildirimler için sıklık seçenekleri

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

Bir kullanıcıyla ilişkili aboneliği siler.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı
- **subscription_id** – Kayıtlı bildirimin adı
- **subscription_id** – tamsayı

1.12.4 Gruplar

4.0 sürümünde geldi.

GET /api/groups/

Grupları yönetme izniniz varsa grupların listesini görüntüler. Yoksa, yalnızca kullanıcının parçası olduğu grupları görüntüler.

Ayrıca bakınız:

Grup nesnesi öz nitelikleri [GET /api/groups/ \(int:id\) /](#) bölümünde bulunabilir.

POST /api/groups/

Yeni bir grup oluşturur.

Parametreler

- **name** (*string*) – Grup adı
- **project_selection** (*int*) – Belirtilen seçeneklerden proje seçimi grubu
- **language_selection** (*int*) – Belirtilen seçeneklerden seçilmiş dil grubu
- **defining_project** (*str*) – *Her proje için erişim denetimi yönetimi* için kullanılan tanımlayıcı proje bağlantısı. Bilgi almak için: `GET /api/projects/(string:project)/`

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

Grup hakkındaki bilgileri görüntüler.

Parametreler

- **id** (*int*) – Grubun kimliği

Yanıt JSON nesnesi

- **name** (*string*) – bir grubun adı
- **project_selection** (*int*) – proje grubuna karşılık gelen tamsayı
- **language_selection** (*int*) – dil grubuna karşılık gelen tamsayı
- **roles** (*array*) – ilişkili rollerin bağlantısı. Bilgi almak için: `GET /api/roles/(int:id)/`
- **projects** (*array*) – ilişkili projelerin bağlantısı. Bilgi almak için: `GET /api/projects/(string:project)/`
- **components** (*array*) – ilişkili bileşenlerin bağlantısı. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/`
- **componentlists** (*array*) – ilişkili bileşen listelerinin bağlantısı. Bilgi almak için: `GET /api/component-lists/(str:slug)/`
- **defining_project** (*str*) – *Her proje için erişim denetimi yönetimi* için kullanılan tanımlayıcı proje bağlantısı. Bilgi almak için: `GET /api/projects/(string:project)/`

Örnek JSON verileri:

```
{
  "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": [
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
    "http://example.com/api/components/demo/weblate/"
  ]
}
```

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

Grup parametrelerini değiştirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Yanıt JSON nesnesi

- **name** (*string*) – bir grubun adı
- **project_selection** (*int*) – proje grubuna karşılık gelen tamsayı
- **language_selection** (*int*) – dil grubuna karşılık gelen tamsayı

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

Grup parametrelerini değiştirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Yanıt JSON nesnesi

- **name** (*string*) – bir grubun adı
- **project_selection** (*int*) – proje grubuna karşılık gelen tamsayı
- **language_selection** (*int*) – dil grubuna karşılık gelen tamsayı

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

Grubu siler.

Parametreler

- **id** (*int*) – Grubun kimliği

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

Rolleri bir grupta ilişkilendirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Form parametreleri

- **string role_id** – Benzersiz rol kimliği

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

Bileşenleri bir grupta ilişkilendirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Form parametreleri

- **string component_id** – Benzersiz bileşen kimliği

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

Bileşeni bir gruptan siler.

Parametreler

- **id** (*int*) – Grubun kimliği

- **component_id** (*int*) – Benzersiz bileşen kimliği

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

Projeleri bir grupta ilişkilendirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Form parametreleri

- **string project_id** – Benzersiz proje kimliği

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

int: *project_id*

Projeyi bir gruptan siler.

Parametreler

- **id** (*int*) – Grubun kimliği
- **project_id** (*int*) – Benzersiz proje kimliği

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

Dilleri bir grupta ilişkilendirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Form parametreleri

- **string language_code** – Benzersiz dil kodu

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

string: *language_code*

Dili bir gruptan siler.

Parametreler

- **id** (*int*) – Grubun kimliği
- **language_code** (*string*) – Benzersiz dil kodu

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

Bileşen listelerini bir grupta ilişkilendirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Form parametreleri

- **string component_list_id** – Benzersiz bileşen listesi kimliği

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

int: *component_list_id*

Bileşen listesini bir gruptan siler.

Parametreler

- **id** (*int*) – Grubun kimliği
- **component_list_id** (*int*) – Benzersiz bileşen listesi kimliği

1.12.5 Roller

GET /api/roles/

Kullanıcıyla ilişkili tüm rollerin listesini görüntüler. Kullanıcı süper kullanıcıysa, var olan tüm rollerin listesini görüntüler.

Ayrıca bakınız:

Roller nesnesi öznitelikleri [GET /api/roles/\(int:id\)/](#) bölümünde bulunabilir.

POST /api/roles/

Yeni bir rol oluşturur.

Parametreler

- **name** (*string*) – Rol adı
- **permissions** (*array*) – İzinlerin kod adı listesi

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

Bir rol hakkındaki bilgileri görüntüler.

Parametreler

- **id** (*int*) – Rol kimliği

Yanıt JSON nesnesi

- **name** (*string*) – Rol adı
- **permissions** (*array*) – izinlerin kod adı listesi

Örnek JSON verileri:

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

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

Rol parametrelerini değiştirir.

Parametreler

- **id** (*int*) – Rolün kimliği

Yanıt JSON nesnesi

- **name** (*string*) – Rol adı
- **permissions** (*array*) – izinlerin kod adı listesi

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

Rol parametrelerini değiştirir.

Parametreler

- **id** (*int*) – Rolün kimliği

Yanıt JSON nesnesi

- **name** (*string*) – Rol adı
- **permissions** (*array*) – izinlerin kod adı listesi

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

Rolü siler.

Parametreler

- **id** (*int*) – Rolün kimliği

1.12.6 Diller

GET `/api/languages/`

Tüm dillerin listesini görüntüler.

Ayrıca bakınız:

Dil nesnesi öznitelikleri `GET /api/languages/(string: language) /` bölümünde bulunabilir.

POST `/api/languages/`

Yeni bir dil oluşturur.

Parametreler

- **code** (*string*) – Dil adı
- **name** (*string*) – Dil adı
- **direction** (*string*) – Metin yönü
- **population** (*int*) – Konuşan sayısı
- **plural** (*object*) – Dil çoğul formülü ve sayısı

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

Bir dil hakkındaki bilgileri görüntüler.

Parametreler

- **language** (*string*) – Dil kodu

Yanıt JSON nesnesi

- **code** (*string*) – Dil kodu
- **direction** (*string*) – Metin yönü
- **plural** (*object*) – Dil çoğul bilgisinin nesnesi
- **aliases** (*array*) – Dil için takma adlar dizisi

JSON nesnesi ister

- **population** (*int*) – Konuşan sayısı

Örnek JSON verileri:

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

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    "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) /

Dil parametrelerini değiştirir.

Parametreler

- **language** (string) – Dilin kodu

JSON nesnesi ister

- **name** (string) – Dil adı
- **direction** (string) – Metin yönü
- **population** (int) – Konuşan sayısı
- **plural** (object) – Dil çoğul ayrıntıları

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

Dil parametrelerini değiştirir.

Parametreler

- **language** (string) – Dilin kodu

JSON nesnesi ister

- **name** (string) – Dil adı
- **direction** (string) – Metin yönü
- **population** (int) – Konuşan sayısı
- **plural** (object) – Dil çoğul ayrıntıları

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

Dili siler.

Parametreler

- **language** (string) – Dilin kodu

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

Bir dilin istatistiklerini görüntüler.

Parametreler

- **language** (string) – Dil kodu

Yanıt JSON nesnesi

- **total** (int) – toplam dizge sayısı
- **total_words** (int) – toplam sözcük sayısı
- **last_change** (timestamp) – dildeki son değişiklikler
- **recent_changes** (int) – toplam değişiklik sayısı
- **translated** (int) – çevrilmiş dizge sayısı

- **translated_percent** (*float*) – çevrilmiş dizge yüzdesi
- **translated_words** (*int*) – çevrilmiş sözcük sayısı
- **translated_words_percent** (*int*) – çevrilmiş sözcük yüzdesi
- **translated_chars** (*int*) – çevrilmiş karakter sayısı
- **translated_chars_percent** (*int*) – çevrilmiş karakter yüzdesi
- **total_chars** (*int*) – toplam karakter sayısı
- **fuzzy** (*int*) – belirsiz (düzenlenecek olarak imlenmiş) dizge sayısı
- **fuzzy_percent** (*int*) – belirsiz (düzenlenecek olarak imlenmiş) dizge yüzdesi
- **failing** (*int*) – denetim hatası olan dizge sayısı
- **failing** – denetim hatası olan dizge yüzdesi

1.12.7 Projeler

GET /api/projects/

Tüm projelerin listesini görüntüler.

Ayrıca bakınız:

Proje nesnesi öznitelikleri [GET /api/projects/\(string:project\)/](#) bölümünde bulunabilir.

POST /api/projects/

3.9 sürümünde geldi.

Yeni bir proje oluşturur.

Parametreler

- **name** (*string*) – Proje adı
- **slug** (*string*) – Proje kısaltması
- **web** (*string*) – Proje sitesi

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

Bir proje hakkındaki bilgileri görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **name** (*string*) – proje adı
- **slug** (*string*) – proje kısaltması
- **web** (*string*) – proje sitesi
- **components_list_url** (*string*) – Bileşen listesi adresi. Bilgi almak için: [GET /api/projects/\(string:project\)/components/](#)
- **repository_url** (*string*) – Depo durumu adresi. Bilgi almak için: [GET /api/projects/\(string:project\)/repository/](#)
- **changes_list_url** (*string*) – Değişiklikler listesi adresi. Bilgi almak için: [GET /api/projects/\(string:project\)/changes/](#)
- **translation_review** (*boolean*) – *Onaylama kullanılsın*
- **source_review** (*boolean*) – *Kaynak onaylama kullanılsın*
- **set_language_team** (*boolean*) – *“Language-Team” üst bilgisi ayarlansın*

- **enable_hooks** (*boolean*) – *Kancalar kullanılsın*
- **instructions** (*string*) – *Çeviri yönergeleri*
- **language_aliases** (*string*) – *Dil kod adları*

Örnek JSON verileri:

```
{
  "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*) /

4.3 sürümünde geldi.

Bir projeyi **PATCH** isteğiyle düzenler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

PUT /api/projects/ (*string: project*) /

4.3 sürümünde geldi.

Bir projeyi **PUT** isteğiyle düzenler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

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

3.9 sürümünde geldi.

Bir projeyi siler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

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

Proje değişikliklerinin listesini görüntüler. Temelde **GET** /api/changes/ kapsamıyla aynı parametreleri kabul eden bir projedir.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – bileşen nesnelerinin dizisi. Bilgi almak için: **GET** /api/changes/(*int:id*)/

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

Sürüm denetimi sistemi deposunun durumu ile ilgili bilgileri görüntüler. Bu noktada, projenin tüm depoları için yalnızca genel bir özet bulunur. Daha ayrıntılı durum bilgileri almak için **GET** /api/components/(*string:project*)/(*string:component*)/repository/ kullanın.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **needs_commit** (*boolean*) – işlenebilecek bekleyen değişikliklerin olup olmadığı

- **needs_merge** (*boolean*) – aktarılabilecek herhangi bir yukarı akış değişikliğinin olup olmadığı
- **needs_push** (*boolean*) – itilecek herhangi bir yerel değişikliğin olup olmadığı

Örnek JSON verileri:

```
{
  "needs_commit": true,
  "needs_merge": false,
  "needs_push": true
}
```

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

Sürüm denetimi sistemi deposunda belirtilen işlemi gerçekleştirir.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

JSON nesnesi ister

- **operation** (*string*) – İşlem: push, pull, commit, reset, cleanup ya da file-sync olabilir

Yanıt JSON nesnesi

- **result** (*boolean*) – işlemin sonucu

CURL örneği:

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

JSON isteği örneği:

```
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"}
```

JSON yanıtı örneği:

```
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/

Belirtilen projedeki çeviri bileşenlerinin listesini görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – bileşen nesneleri dizisi. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/`

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

3.9 sürümünde geldi.

4.3 sürümünde değişti: Sürüm denetimi sistemi kullanmayan bileşenler için de `zipfile` ve `docfile` parametreleri kabul edilir. Bilgi almak için: *Yerel dosyalar*.

4.6 sürümünde değişti: Kopyalanan depolar *İç Weblate adresleri* kullanılarak kendiliğinden bir proje içinde paylaşılabilir. Bu özelliği kapatmak için `disable_autoshare` seçeneğini kullanın.

Belirtilen projede çeviri bileşenleri oluşturur.

İpucu: Tek bir sürüm denetimi sistemi deposundan birden fazla bileşen oluştururken *İç Weblate adresleri* seçeneğini kullanın.

Not: Bileşen oluşturma işleminin çoğu arka planda yapılır. Oluşturulan bileşenin `task_url` özniteliğini denetleyin ve oradaki ilerlemeyi izleyin.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Form parametreleri

- **file zipfile** – Çevirilerin başlatılması için Weblate üzerine yüklenecek ZIP dosyası
- **file docfile** – Çevrilecek belge
- **boolean disable_autoshare** – *İç Weblate adresleri* seçeneği ile kendiliğinden depo paylaşımını devre dışı bırakır.

JSON nesnesi ister

- **object** – Bileşen parametreleri. Ayrıntılı bilgi almak için `GET /api/components/(string:project)/(string:component)/`

Yanıt JSON nesnesi

- **result** (*object*) – Oluşturulan bileşen nesnesi. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/`

Dosyalar `zipfile` ve `docfile` parametreleri kullanılarak yüklenirken JSON kullanılamaz. Veriler `multipart/form-data` olarak yüklenmelidir.

CURL form isteği örneği:

```
curl \
  --form docfile=@strings.html \
  --form name=Weblate \
  --form slug=weblate \
  --form file_format=html \
  --form new_lang=add \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

CURL JSON isteği örneği:

```
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/
```

Git üzerinden yeni bir bileşen oluşturan JSON isteği:

```
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"
}
```

Başka bir bileşenden yeni bir bileşen oluşturan JSON isteği:

```
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"
}
```

JSON yanıtı örneği:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

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/`

Bir projedeki tüm diller için sayılanmış istatistikleri görüntüler.

3.8 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – çeviri istatistiği nesneleri dizisi
- **language** (*string*) – dil adı
- **code** (*string*) – dil kodu
- **total** (*int*) – toplam dizge sayısı
- **translated** (*int*) – çevrilmiş dizge sayısı
- **translated_percent** (*float*) – çevrilmiş dizge yüzdesi
- **total_words** (*int*) – toplam sözcük sayısı
- **translated_words** (*int*) – çevrilmiş sözcük sayısı

- **words_percent** (*float*) – çevrilmiş sözcük yüzdesi

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

Bir projenin istatistiklerini görüntüler.

3.8 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **total** (*int*) – toplam dizge sayısı
- **translated** (*int*) – çevrilmiş dizge sayısı
- **translated_percent** (*float*) – çevrilmiş dizge yüzdesi
- **total_words** (*int*) – toplam sözcük sayısı
- **translated_words** (*int*) – çevrilmiş sözcük sayısı
- **words_percent** (*float*) – çevrilmiş sözcük yüzdesi

1.12.8 Bileşenler

İpucu: Yeni bileşenler oluşturmak için `POST /api/projects/(string:project)/components/` kullanın.

GET `/api/components/`

Çeviri bileşenlerinin listesini görüntüler.

Ayrıca bakınız:

Bileşen nesnesi özenitelikleri `GET /api/components/(string:project)/(string:component)/` bölümünde bulunabilir.

GET `/api/components/(string:project)/string:component/`

Çeviri bileşeni hakkındaki bilgileri görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **project** (*object*) – çeviri projesi. Bilgi almak için `GET /api/projects/(string:project)/`
- **name** (*string*) – *Bileşen adı*
- **slug** (*string*) – *Bileşen kısaltması*
- **vcs** (*string*) – *Sürüm denetimi sistemi*
- **repo** (*string*) – *Kaynak kod deposu*
- **git_export** (*string*) – *Dışa aktarılmış depo adresi*
- **branch** (*string*) – *Depo dalı*
- **push_branch** (*string*) – *İtme işleminin yapılacağı dal*
- **filemask** (*string*) – *Dosya maskesi*

- **template** (*string*) – Tek dilli temel dil dosyası
- **edit_template** (*string*) – Temel dosyayı düzenle
- **intermediate** (*string*) – Ara dil dosyası
- **new_base** (*string*) – Yeni çeviriler kalıbı
- **file_format** (*string*) – Dosya biçimi
- **license** (*string*) – Çeviri lisansı
- **agreement** (*string*) – Katılımcı anlaşması
- **new_lang** (*string*) – Yeni çeviri ekleme
- **language_code_style** (*string*) – Dil kodu biçimi
- **source_language** (*object*) – kaynak dil nesnesi. Ayrıntılı bilgi alın `GET /api/languages/(string:language)/`
- **push** (*string*) – Depo itme adresi
- **check_flags** (*string*) – Çeviri imleri
- **priority** (*string*) – Öncelik
- **enforced_checks** (*string*) – Zorunlu denetimler
- **restricted** (*string*) – Kısıtlı erişim
- **repoweb** (*string*) – Depo tarayıcısı
- **report_source_bugs** (*string*) – Kaynak dizge hata bildirme adresi
- **merge_style** (*string*) – Birleştirme biçimi
- **commit_message** (*string*) – İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri
- **add_message** (*string*) – İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri
- **delete_message** (*string*) – İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri
- **merge_message** (*string*) – İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri
- **addon_message** (*string*) – İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri
- **pull_message** (*string*) – İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri
- **allow_translation_propagation** (*string*) – Çevirilerin yayılmasını sağlamak
- **enable_suggestions** (*string*) – Öneriler kullanılsın
- **suggestion_voting** (*string*) – Öneriler oylanabilsin
- **suggestion_autoaccept** (*string*) – Öneriler kendiliğinden kabul edilsin
- **push_on_commit** (*string*) – İşleme ile itme
- **commit_pending_age** (*string*) – İşlenecek değişikliklerin yaşı
- **auto_lock_error** (*string*) – Hata durumunda kilitlenme
- **language_regex** (*string*) – Dil süzgeci
- **variant_regex** (*string*) – Çeşitler kurallı ifadesi

- **repository_url** (*string*) – Depo durumu adresi. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/repository/`
- **translations_url** (*string*) – Çeviri listesi adresi. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/translations/`
- **lock_url** (*string*) – Kilit durumu adresi. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/lock/`
- **changes_list_url** (*string*) – Değişiklik listesi adresi. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/changes/`
- **task_url** (*string*) – Arka plan görevi (varsa) adresi. Bilgi almak için: `GET /api/tasks/(str:uuid)/`

Örnek JSON verileri:

```
{
  "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/

Bir bileşeni **PATCH** isteği ile düzenleme.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **source_language** (*string*) – Proje kaynak dili kodu (isteğe bağlı)

JSON nesnesi ister

- **name** (*string*) – bileşen adı
- **slug** (*string*) – bileşen kısaltması
- **repo** (*string*) – Sürüm denetimi sistemi deposunun adresi

CURL örneği:

```
curl \
  --data-binary '{"name": "new name"}' \
  -H "Content-Type: application/json" \
  -H "Authorization: Token TOKEN" \
  PATCH http://example.com/api/projects/hello/components/
```

JSON isteği örneği:

```
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"
}
```

JSON yanıtı örneği:

```
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/"
    }
  }
}
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
  },
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "url": "http://example.com/api/components/hello/weblate/",
  "vcs": "git",
  "web_url": "http://example.com/projects/hello/weblate/"
}

```

PUT `/api/components/(string: project) /`
string: `component/`

Bir bileşeni **PUT** isteği ile düzenleme.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

JSON nesnesi ister

- **branch** (*string*) – Sürüm denetimi sistemi deposunun dalı
- **file_format** (*string*) – çevirilerin dosya biçimi
- **filemask** (*string*) – depodaki çeviri dosyalarının modeli
- **name** (*string*) – bileşen adı
- **slug** (*string*) – bileşen kısaltması
- **repo** (*string*) – Sürüm denetimi sistemi deposunun adresi
- **template** (*string*) – tek dilli çeviriler için temel dosya
- **new_base** (*string*) – yeni çeviriler eklemek için temel dosya
- **vcs** (*string*) – sürüm denetimi sistemi

DELETE `/api/components/(string: project) /`
string: `component/`

3.9 sürümünde geldi.

Bir bileşeni siler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

GET `/api/components/(string: project) /`
string: `component/changes/`

Bileşen değişikliklerinin listesini görüntüler. Temelde `GET /api/changes/` kapsamında bir bileşendir ve aynı parametreleri kabul eder.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – bileşen nesnelerinin dizisi. Bilgi almak için: `GET /api/changes/(int:id)/`

GET /api/components/(string: project) /string: component/file/

4.9 sürümünde geldi.

Bileşenle ilişkili tüm kullanılabilir çevirileri istenilen biçimdeki arşiv dosyası olarak indirir.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Sorgu parametreleri

- **format** (*string*) – Kullanılacak arşiv biçimi belirtilmemişse varsayılan olarak zip biçimi kullanılır. Desteklenen biçimler: zip

GET /api/components/(string: project) /string: component/screenshots/

Bileşen ekran görüntülerinin listesini görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – bileşen ekran görüntüleri dizisi. Bilgi almak için: `GET /api/screenshots/(int:id)/`

GET /api/components/(string: project) /string: component/lock/

Bileşenin kilit durumunu görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **locked** (*boolean*) – bileşenin güncellemeler için kilitlenip kilitlenmediği

Örnek JSON verileri:

```
{
  "locked": false
}
```

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

Bileşenin kilit durumunu ayarlar.

Yanıt `GET /api/components/(string:project)/(string:component)/lock/` ile aynıdır.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

JSON nesnesi ister

- **lock** – Kilitli olduğunu ya da olmadığını belirten ikili değer.

CURL örneği:

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

JSON isteği örneği:

```
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}
```

JSON yanıtı örneği:

```
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/`

Sürüm denetimi sistemi deposunun durumu hakkındaki bilgileri görüntüler.

Yanıt `GET /api/projects/(string:project)/repository/` ile aynıdır.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **needs_commit** (*boolean*) – işlenebilecek bekleyen değişikliklerin olup olmadığı
- **needs_merge** (*boolean*) – aktarılabilecek herhangi bir yukarı akış değişikliğinin olup olmadığı
- **needs_push** (*boolean*) – itilecek herhangi bir yerel değişikliğin olup olmadığı
- **remote_commit** (*string*) – Uzaktan işleme bilgileri
- **status** (*string*) – Sürüm denetimi sistemi tarafından bildirilen sürüm denetimi sistemi deposunun durumu
- **merge_failure** – Varsa aktarım hatasını açıklayan metin, yoksa null

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

Sürüm denetimi sistemi deposunda belirtilen işlemi gerçekleştirir.

Belgeler için `POST /api/projects/(string:project)/repository/` adresine bakabilirsiniz.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

JSON nesnesi ister

- **operation** (*string*) – Yapılacak işlem: push, pull, commit, reset ya da cleanup

Yanıt JSON nesnesi

- **result** (*boolean*) – işlemin sonucu

CURL örneği:

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

JSON isteği örneği:

```
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"}
```

JSON yanıtı örneği:

```
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/*

Tek dilli çeviriler için temel dosyayı indirir.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

GET /api/components/(string: *project*) /
string: *component/new_template/*

Yeni çeviriler için kalıp dosyasını indirir.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

GET /api/components/(string: *project*) /
string: *component/translations/*

Belirtilen bileşendeki çeviri nesnelerinin listesini görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – çeviri nesneleri dizisi. Bilgi almak için: `GET /api/translations/(string:project)/(string:component)/(string:language)/`

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

Belirtilen bileşende yeni çeviri oluşturur.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

JSON nesnesi ister

- **language_code** (*string*) – çeviri dili kodu. Bilgi almak için: `GET /api/languages/(string:language)/`

Yanıt JSON nesnesi

- **result** (*object*) – oluşturulan yeni çeviri nesnesi

CURL örneği:

```
curl \
  -d language_code=cs \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

JSON isteği örneği:

```
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"}
```

JSON yanıtı örneği:

```
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,
```

(sonraki sayfaya devam)

```

    "fuzzy_percent": 0.0,
    "fuzzy_words": 0,
    "have_comment": 0,
    "have_suggestion": 0,
    "is_template": false,
    "is_source": false,
    "language": {
      "code": "cs",
      "direction": "ltr",
      "population": 1303174280
      "name": "Czech",
      "url": "http://example.com/api/languages/cs/",
      "web_url": "http://example.com/languages/cs/"
    },
    "language_code": "cs",
    "id": 125,
    "last_author": null,
    "last_change": null,
    "share_url": "http://example.com/engage/hello/cs/",
    "total": 4,
    "total_words": 15,
    "translate_url": "http://example.com/translate/hello/weblate/cs/",
    "translated": 0,
    "translated_percent": 0.0,
    "translated_words": 0,
    "url": "http://example.com/api/translations/hello/weblate/cs/",
    "web_url": "http://example.com/projects/hello/weblate/cs/"
  }
}

```

GET `/api/components/(string: project) /`
string: `component/statistics/`

Bileşen içindeki tüm çevirilerin sayfalandırılmış istatistiklerini görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – çeviri istatistikleri nesneleri dizisi. Bilgi almak için: `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/`

GET `/api/components/(string: project) /`
string: `component/links/`

Bir bileşenle ilişkili projeleri görüntüler.

4.5 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **projects** (*array*) – ilişkili projeler. Bilgi almak için: `GET /api/projects/(string:project)/`

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

Projeyi bir bileşenle ilişkilendirir.

4.5 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Form parametreleri

- **string project_slug** – Proje kısaltması

DELETE /api/components/ (**string:** *project*) /
string: *component/links/string:* *project_slug/*

Bir projenin bir bileşenle ilişkisini kaldırır.

4.5 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **project_slug** (*string*) – Kaldırılacak projenin kısaltması

1.12.9 Çeviriler

GET /api/translations/

Çevirilerin listesini görüntüler.

Ayrıca bakınız:

Çeviri nesnesi öznitelikleri **GET** /api/translations/ (**string:** *project*) /
(**string:** *component*) / (**string:** *language*) / bölümünde bulunabilir.

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

Bir çeviriyle ilgili bilgileri görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

Yanıt JSON nesnesi

- **component** (*object*) – bileşen nesnesi. Bilgi almak için: **GET** /api/components/ (**string:** *project*) / (**string:** *component*) /
- **failing_checks** (*int*) – denetim hatası olan dizge sayısı
- **failing_checks_percent** (*float*) – denetim hatası olan dizge yüzdesi
- **failing_checks_words** (*int*) – denetim hatası olan sözcük sayısı
- **filename** (*string*) – çeviri dosyası adı
- **fuzzy** (*int*) – belirsiz (düzenlenecek olarak imlenmiş) dizge sayısı
- **fuzzy_percent** (*float*) – belirsiz (düzenlenecek olarak imlenmiş) dizge yüzdesi
- **fuzzy_words** (*int*) – belirsiz (düzenlenecek olarak imlenmiş) dizge sayısı
- **have_comment** (*int*) – belirsiz (düzenlenecek olarak imlenmiş) dizge yüzdesi
- **have_suggestion** (*int*) – önerileri olan dizge sayısı

- **is_template** (*boolean*) – çeviri tabanının tek dilli olup olmadığı
- **language** (*object*) – kaynak dil nesnesi. Ayrıntılı bilgi alın `GET /api/languages/(string:language)/`
- **language_code** (*string*) – depoda kullanılan dil kodu. Dil nesnesindeki dil kodundan farklı olabilir
- **last_author** (*string*) – son yazarın adı
- **last_change** (*timestamp*) – son değişiklik zaman damgası
- **revision** (*string*) – dosyanın sürüm karması
- **share_url** (*string*) – Etkileşim sayfasına yönlendiren paylaşım adresi
- **total** (*int*) – toplam dizge sayısı
- **total_words** (*int*) – toplam sözcük sayısı
- **translate_url** (*string*) – Çeviri adresi
- **translated** (*int*) – çevrilmiş dizge sayısı
- **translated_percent** (*float*) – çevrilmiş dizge yüzdesi
- **translated_words** (*int*) – çevrilmiş sözcük sayısı
- **repository_url** (*string*) – Depo durumu adresi: Bilgi almak için: `GET /api/translations/(string:project)/(string:component)/(string:language)/repository/`
- **file_url** (*string*) – Dosya nesnesi adresi. Bilgi almak için: `GET /api/translations/(string:project)/(string:component)/(string:language)/file/`
- **changes_list_url** (*string*) – Değişiklik listesi adresi. Bilgi almak için: `GET /api/translations/(string:project)/(string:component)/(string:language)/changes/`
- **units_list_url** (*string*) – Dizge listesi adresi. Bilgi almak için: `GET /api/translations/(string:project)/(string:component)/(string:language)/units/`

Örnek JSON verileri:

```
{
  "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/"
    }
  }
}
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

        "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/

3.9 sürümünde geldi.

Bir çeviriyi siler.

Parametreler

- **project** (string) – Proje adresi kısaltması
- **component** (string) – Bileşen adresinin kısaltması
- **language** (string) – Çeviri dili kodu

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

Çeviri değişikliklerinin listesini görüntüler. Temelde aynı parametreleri kabul eden bir `GET /api/changes/` çeviri kapsamıdır.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

Yanıt JSON nesnesi

- **results** (*array*) – bileşen nesnelerinin dizisi. Bilgi almak için: `GET /api/changes/(int:id)/`

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

Çeviri birimlerinin listesini görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu
- **q** (*string*) – Arama sorgusu dizgesi *Arama* (isteğe bağlı)

Yanıt JSON nesnesi

- **results** (*array*) – bileşen nesneleri dizisi. Bilgi almak için: `GET /api/units/(int:id)/`

POST /api/translations/(string: project) /
string: component/string: language/units/

Yeni birim ekle.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

JSON nesnesi ister

- **key** (*string*) – Çeviri biriminin adı (anahtar veya bağlam olarak kullanılır)
- **value** (*array*) – Kaynak dizgeler (çoğul oluşturumuyorsanız tekil dizge kullanın)
- **state** (*int*) – Dizge durumu. Bilgi almak için: `GET /api/units/(int:id)/`

Yanıt JSON nesnesi

- **unit** (*object*) – yeni oluşturulmuş birim. Bilgi almak için: `GET /api/units/(int:id)/`

Ayrıca bakınız:

Dizge yönetimi, *adding-new-strings*

POST /api/translations/(string: project) /
string: component/string: language/autotranslate/

Kendiliğinden çeviriyi tetikler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

JSON nesnesi ister

- **mode** (*string*) – Kendiliğinden çeviri kipi
- **filter_type** (*string*) – Kendiliğinden çeviri süzgeci türü
- **auto_source** (*string*) – Kendiliğinden çeviri kaynağı - mt ya da others
- **component** (*string*) – Ek bileşenlere erişebilmesi için projenin paylaşılan çeviri bel-
leğine katkıda bulunmayı etkinleştirir.
- **engines** (*array*) – Makine çevirisi hizmetleri
- **threshold** (*string*) – Skor eşiği

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

Geçerli çeviri dosyasını sürüm denetimi sisteminde tutulduğu biçimde (*format* parametresini kullanmadan) ya da başka bir biçime dönüştürülmüş olarak indirir (bilgi almak için: [Çevirileri indirmek](#)).

Not: Bu API noktası, veriler yerine tüm dosya üzerinde çalıştığından çıktı için geri kalan API işleyişinden farklı bir mantık kullanır. Kabul edilen *format* parametre kümesi farklıdır ve böyle bir parametre belirtilmediğinde sürüm denetimi sisteminde depolanmış çeviri dosyasını alır.

Sorgu parametreleri

- **format** – Kullanılacak dosya biçimi. Belirtilmemişse biçim dönüşümü yapılmaz. Desteklenen biçimlerle ilgili bilgi almak için: :ref:`download`
- **q** (*string*) – İndirilen dizgeleri süzer. Bilgi almak için: search. Yalnızca dönüşüm yapıldığında geçerlidir (*format* belirtildiğinde).

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

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

Çevirileri içeren yeni dosya yükler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

Form parametreleri

- **string conflicts** – Çakışmaların nasıl işleneceği (ignore, replace-translated ya da replace-approved)
- **file file** – Yüklenen dosya
- **string email** – İlgilinin e-posta adresi
- **string author** – Katkıda bulunanın adı
- **string method** – Yükleme yöntemi (translate, approve, suggest, fuzzy, replace, source, add). Bilgi almak için: [İçe aktarma yöntemleri](#)
- **string fuzzy** – Belirsiz (düzenlenecek olarak imlenmiş) dizgeleri işleme (empty, process, approve)

CURL örneği:

```
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/`

Sürüm denetimi sistemi deposunun durumu hakkındaki bilgileri görüntüler.

Yanıt `GET /api/components/(string:project)/(string:component)/repository/` ile aynıdır.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

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

Sürüm denetimi sistemi deposunda belirtilen işlemi gerçekleştirir.

Belgeler için `POST /api/projects/(string:project)/repository/` adresine bakabilirsiniz.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

JSON nesnesi ister

- **operation** (*string*) – Yapılacak işlem: push, pull, commit, reset ya da cleanup

Yanıt JSON nesnesi

- **result** (*boolean*) – işlemin sonucu

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

Ayrıntılı çeviri istatistiklerini görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

Yanıt JSON nesnesi

- **code** (*string*) – dil kodu
- **failing** (*int*) – denetim hataları sayısı
- **failing_percent** (*float*) – denetim hataları yüzdesi
- **fuzzy** (*int*) – belirsiz (düzenlenecek olarak imlenmiş) dizge sayısı
- **fuzzy_percent** (*float*) – belirsiz (düzenlenecek olarak imlenmiş) dizge yüzdesi
- **total_words** (*int*) – toplam sözcük sayısı
- **translated_words** (*int*) – çevrilmiş sözcük sayısı
- **last_author** (*string*) – son yazarın adı

- **last_change** (*timestamp*) – son değişiklik tarihi
- **name** (*string*) – dil adı
- **total** (*int*) – toplam dizge sayısı
- **translated** (*int*) – çevrilmiş dizge sayısı
- **translated_percent** (*float*) – çevrilmiş dizge yüzdesi
- **url** (*string*) – Çeviriye erişme adresi (etkileşim adresi)
- **url_translate** (*string*) – Çeviriye erişme adresi (gerçek çeviri adresi)

1.12.10 Bellek

4.14 sürümünde geldi.

GET /api/memory/

Bileşen listelerinin listesini görüntüler.

DELETE /api/memory/ (*int: memory_object_id*) /

Bir çeviri belleği nesnesini siler

Parametreler

- **memory_object_id** – Çeviri belleği nesnesi kimliği

1.12.11 Birimler

Birim, bir kaynak dizgeyi karşılık gelen çevrilmiş bir dizgeyle eşleştiren ve ayrıca bazı ilgili üst verileri içeren tek bir çeviri parçasıdır. Terim, [Translate Toolkit](#) ve XLIFF kapsamından türetilmiştir.

GET /api/units/

Çeviri birimlerinin listesini görüntüler.

Parametreler

- **q** (*string*) – Arama sorgusu dizgesi *Arama* (isteğe bağlı)

Ayrıca bakınız:

Birim nesnesi öznitelikleri [GET /api/units/\(int:id\)/](#) bölümünde bulunabilir.

GET /api/units/ (*int: id*) /

4.3 sürümünde değişti: Hedef ve kaynak çoğul dizgeleri doğru işlemek için diziye dönüştürülmüştür.

Çeviri birimi hakkındaki bilgileri görüntüler.

Parametreler

- **id** (*int*) – Birim kimliği

Yanıt JSON nesnesi

- **translation** (*string*) – İlişkili çeviri nesnesinin adresi
- **source** (*array*) – kaynak dizge
- **previous_source** (*string*) – belirsiz eşleştirmesi için kullanılan önceki kaynak dizge
- **target** (*array*) – hedef dizge
- **id_hash** (*string*) – birimin benzersiz kimliği
- **content_hash** (*string*) – kaynak dizgenin benzersiz kimliği
- **location** (*string*) – birimin kaynak kodundaki konumu

- **context** (*string*) – çeviri birimi bağlamı
- **note** (*string*) – çeviri birimi notu
- **flags** (*string*) – çeviri birimi imleri
- **labels** (*array*) – çeviri birimi etiketleri, kaynak birimlerde kullanılabilir
- **state** (*int*) – birim durumu, 0 - Çevrilmemiş, 10 - Düzenlenmesi gerekli, 20 - Çevrilmiş, 30 - Onaylanmış, 100 - Salt okunur
- **fuzzy** (*boolean*) – birimin belirsiz ya da onaylanmayı bekliyor olarak imlenmiş olup olmadığı
- **translated** (*boolean*) – birimin çevrilmiş olup olmadığı
- **approved** (*boolean*) – çevirinin onaylanmış olup olmadığı
- **position** (*int*) – birimin çeviri dosyasındaki konumu
- **has_suggestion** (*boolean*) – birimin önerileri olup olmadığı
- **has_comment** (*boolean*) – birimin yorumları olup olmadığı
- **has_failing_check** (*boolean*) – birimde denetim hatası olup olmadığı
- **num_words** (*int*) – kaynak sözcük sayısı
- **priority** (*int*) – çeviri önceliği. Varsayılan değer: 100
- **id** (*int*) – birim kimliği
- **explanation** (*string*) – Dizge açıklaması. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Kaynak dizgeler ile ilgili ek bilgiler](#)
- **extra_flags** (*string*) – Ek dizge imleri. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)
- **web_url** (*string*) – Birimin düzenlenebileceği adres
- **source_unit** (*string*) – Kaynak birim bağlantısı. Bilgi almak için: `GET /api/units/(int:id)/`
- **pending** (*boolean*) – birimin belirsiz ya da onaylanmış olarak imlenmiş olup olmadığı
- **timestamp** (*timestamp*) – dizge yaşı

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

4.3 sürümünde geldi.

Çeviri birimi üzerinde bazı güncellemeler yapar.

Parametreler

- **id** (*int*) – Birim kimliği

JSON nesnesi ister

- **state** (*int*) – birim durumu, 0 - Çevrilmemiş, 10 - Düzenlenmesi gerekli, 20 - Çevrilmiş, 30 - Onaylanmış (gözden geçirme iş akışı etkinleştirilmelidir. Bilgi almak için: `:ref: reviews`)
- **target** (*array*) – hedef dizge
- **explanation** (*string*) – Dizge açıklaması. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Kaynak dizgeler ile ilgili ek bilgiler](#)
- **extra_flags** (*string*) – Ek dizge imleri. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)

Yanıt JSON nesnesi

- **labels** (*array*) – etiketler, kaynak birimlerde kullanılabilir

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

4.3 sürümünde geldi.

Çeviri birimi üzerinde tüm güncellemeleri yapar.

Parametreler

- **id** (*int*) – Birim kimliği

JSON nesnesi ister

- **state** (*int*) – birim durumu, 0 - Çevrilmemiş, 10 - Düzenlenmesi gerekli, 20 - Çevrilmiş, 30 - Onaylanmış (gözden geçirme iş akışı etkinleştirilmelidir. Bilgi almak için: `:ref: reviews`)
- **target** (*array*) – hedef dizge
- **explanation** (*string*) – Dizge açıklaması. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Kaynak dizgeler ile ilgili ek bilgiler](#)
- **extra_flags** (*string*) – Ek dizge imleri. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)

Yanıt JSON nesnesi

- **labels** (*array*) – etiketler, kaynak birimlerde kullanılabilir

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

4.3 sürümünde geldi.

Bir çeviri birimini siler.

Parametreler

- **id** (*int*) – Birim kimliği

1.12.12 Değişiklikler

GET `/api/changes/`

4.1 sürümünde değişti: Değişiklikleri süzme özelliği 4.1 sürümünde sunuldu.

Çeviri değişikliklerinin listesini görüntüler.

Ayrıca bakınız:

Değişiklik nesnesi öznitelikleri `GET /api/changes/(int:id) /` bölümünde bulunabilir.

Sorgu parametreleri

- **user** (*string*) – Süzülecek kullanıcının kullanıcı adı
- **action** (*int*) – Süzülecek işlem, birkaç kez kullanılabilir
- **timestamp_after** (*timestamp*) – Sonraki değişiklikleri listelemek için ISO 8601 biçiminde zaman damgası
- **timestamp_before** (*timestamp*) – Önceki değişiklikleri listelemek için ISO 8601 biçiminde zaman damgası

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

Çeviri değişikliği hakkındaki bilgileri görüntüler.

Parametreler

- **id** (*int*) – Değişiklik kimliği

Yanıt JSON nesnesi

- **unit** (*string*) – İlişkili birim nesnesinin adresi

- **translation** (*string*) – İlişkili çeviri nesnesinin adresi
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **user** (*string*) – İlişkili kullanıcı nesnesinin adresi
- **author** (*string*) – İlişkili yazar nesnesinin adresi
- **timestamp** (*timestamp*) – etkinlik zaman damgası
- **action** (*int*) – işlemin sayısal kimliği
- **action_name** (*string*) – işlemin metin açıklaması
- **target** (*string*) – metin ya da ayrıntı değişikliği etkinliği
- **id** (*int*) – değişiklik kimliği

1.12.13 Ekran görüntüleri

GET /api/screenshots/

Ekran görüntüsü dizgesi bilgilerinin listesini görüntüler.

Ayrıca bakınız:

Ekran görüntüsü nesnesi öznitelikleri [GET /api/screenshots/\(int:id\)/](#) bölümünde bulunabilir.

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

Ekran görüntüsü bilgileriyle ilgili bilgileri görüntüler.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

Yanıt JSON nesnesi

- **name** (*string*) – ekran görüntüsünün adı
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **file_url** (*string*) – Bir dosyayı indirme bağlantısı. Bilgi almak için: [GET /api/screenshots/\(int:id\)/file/](#)
- **units** (*array*) – ilişkili kaynak dizge bilgileri bağlantısı. Bilgi almak için: [GET /api/units/\(int:id\)/](#)

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

Ekran görüntüsünü indir.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

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

Ekran görüntüsünü değiştir.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

Form parametreleri

- **file image** – Yüklenen dosya

CURL örneği:

```
curl -X POST \
-F image=@image.png \
-H "Authorization: Token TOKEN" \
http://example.com/api/screenshots/1/file/
```

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

Kaynak dizgeyi ekran görüntüsüyle ilişkilendirir.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

Form parametreleri

- **string unit_id** – Birim kimliği

Yanıt JSON nesnesi

- **name** (*string*) – ekran görüntüsünün adı
- **translation** (*string*) – İlişkili çeviri nesnesinin adresi
- **file_url** (*string*) – Bir dosyayı indirme bağlantısı. Bilgi almak için: [GET /api/screenshots/\(int:id\)/file/](#)
- **units** (*array*) – ilişkili kaynak dizge bilgileri bağlantısı. Bilgi almak için: [GET /api/units/\(int:id\)/](#)

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

Kaynak dizge ile ekran görüntüsünün ilişkilendirmesini kaldırır.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği
- **unit_id** – Kaynak dizge birimi kimliği

POST /api/screenshots/

Yeni bir ekran görüntüsü oluşturur.

Form parametreleri

- **file image** – Yüklenen dosya
- **string name** – Ekran görüntüsü adı
- **string project_slug** – Proje kısaltması
- **string component_slug** – Bileşen kısaltması
- **string language_code** – Dil kodu

Yanıt JSON nesnesi

- **name** (*string*) – ekran görüntüsünün adı
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **file_url** (*string*) – Bir dosyayı indirme bağlantısı. Bilgi almak için: [GET /api/screenshots/\(int:id\)/file/](#)
- **units** (*array*) – ilişkili kaynak dizge bilgileri bağlantısı. Bilgi almak için: [GET /api/units/\(int:id\)/](#)

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

Ekran görüntüsü ile ilgili bazı bilgileri düzenle.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

Yanıt JSON nesnesi

- **name** (*string*) – ekran görüntüsünün adı
- **component** (*string*) – İlişkili bileşen nesnesinin adresi

- **file_url** (*string*) – Bir dosyayı indirme bağlantısı. Bilgi almak için: *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ilişkili kaynak dizge bilgileri bağlantısı. Bilgi almak için: *GET /api/units/(int:id)/*

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

Ekran görüntüsü ile ilgili tüm bilgileri düzenle.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

Yanıt JSON nesnesi

- **name** (*string*) – ekran görüntüsünün adı
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **file_url** (*string*) – Bir dosyayı indirme bağlantısı. Bilgi almak için: *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ilişkili kaynak dizge bilgileri bağlantısı. Bilgi almak için: *GET /api/units/(int:id)/*

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

Ekran görüntüsünü sil.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

1.12.14 Eklentiler

4.4.1 sürümünde geldi.

GET /api/addons/

Eklentilerin listesini görüntüler.

Ayrıca bakınız:

Eklenti nesnesi öznitelikleri *GET /api/addons/(int:id) /* bölümünde bulunabilir.

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

Eklentiyle ilgili bilgileri görüntüler.

Parametreler

- **id** (*int*) – Eklenti kimliği

Yanıt JSON nesnesi

- **name** (*string*) – bir eklentinin adı
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **configuration** (*object*) – İsteğe bağlı eklenti yapılandırması

Ayrıca bakınız:

Eklentiler

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

Yeni bir eklenti oluşturur.

Parametreler

- **project_slug** (*string*) – Proje kısaltması

- **component_slug** (*string*) – Bileşen kısaltması

JSON nesnesi ister

- **name** (*string*) – bir eklentinin adı
- **configuration** (*object*) – İsteğe bağlı eklenti yapılandırması

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

Eklenti ile ilgili bazı bilgileri düzenle.

Parametreler

- **id** (*int*) – Eklenti kimliği

Yanıt JSON nesnesi

- **configuration** (*object*) – İsteğe bağlı eklenti yapılandırması

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

Eklenti ile ilgili tüm bilgileri düzenle.

Parametreler

- **id** (*int*) – Eklenti kimliği

Yanıt JSON nesnesi

- **configuration** (*object*) – İsteğe bağlı eklenti yapılandırması

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

Eklentiği sil.

Parametreler

- **id** (*int*) – Eklenti kimliği

1.12.15 Bileşen listeleri

4.0 sürümünde geldi.

GET /api/component-lists/

Bileşen listelerinin listesini görüntüler.

Ayrıca bakınız:

Bileşen listesi nesnesi öznitelikleri [GET /api/component-lists/\(str:slug\)/](#) bölümünde bulunabilir.

GET /api/component-lists/ (str: slug) /

Bileşen listesi hakkındaki bilgileri görüntüler.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması

Yanıt JSON nesnesi

- **name** (*string*) – bir bileşen listesinin adı
- **slug** (*string*) – bir bileşen listesinin kısaltması
- **show_dashboard** (*boolean*) – bir panoda görüntülenip görüntülenmeyeceği
- **components** (*array*) – ilişkili bileşenlerin bağlantısı. Bilgi almak için: [GET /api/components/\(string:project\)/\(string:component\)/](#)
- **auto_assign** (*array*) – kendiliğinden atama kuralları

PUT `/api/component-lists/ (str: slug) /`

Bileşen listesi parametrelerini değiştirir.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması

JSON nesnesi ister

- **name** (*string*) – bir bileşen listesinin adı
- **slug** (*string*) – bir bileşen listesinin kısaltması
- **show_dashboard** (*boolean*) – bir panoda görüntülenip görüntülenmeyeceği

PATCH `/api/component-lists/ (str: slug) /`

Bileşen listesi parametrelerini değiştirir.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması

JSON nesnesi ister

- **name** (*string*) – bir bileşen listesinin adı
- **slug** (*string*) – bir bileşen listesinin kısaltması
- **show_dashboard** (*boolean*) – bir panoda görüntülenip görüntülenmeyeceği

DELETE `/api/component-lists/ (str: slug) /`

Bileşen listesini siler.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması

POST `/api/component-lists/ (str: slug) /components/`

Bileşeni bir bileşen listesiyle ilişkilendirir.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması

Form parametreleri

- **string component_id** – Bileşen kimliği

DELETE `/api/component-lists/ (str: slug) /components/
str: component_slug`

Bileşen listesinden bir bileşenin ilişkisini kaldırır.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması
- **component_slug** (*string*) – Bileşen kısaltması

1.12.16 Sözlük

4.5 sürümünde değişti: Sözlükler artık normal bileşenler, çeviriler ve dizgeler olarak kaydediliyor. Lütfen bunun yerine ilgili API uygulamasını kullanın.

1.12.17 Görevler

4.4 sürümünde geldi.

GET `/api/tasks/`

Görevlerin listesi şu anda alınamıyor.

GET `/api/tasks/(str: uuid) /`

Bir görev hakkındaki bilgileri görüntüler

Parametreler

- **uuid** (*string*) – Görev UUID

Yanıt JSON nesnesi

- **completed** (*boolean*) – Görevin tamamlanıp tamamlanmadığı
- **progress** (*int*) – Yüzde olarak görev ilerlemesi
- **result** (*object*) – Görev sonucu veya ilerleme ayrıntıları
- **log** (*string*) – Görev günlüğü

1.12.18 Ölçümler

GET `/api/metrics/`

Sunucu ölçümlerini görüntüler.

Yanıt JSON nesnesi

- **units** (*int*) – Birim sayısı
- **units_translated** (*int*) – Çevrilmiş birim sayısı
- **users** (*int*) – Kullanıcı sayısı
- **changes** (*int*) – Değişiklik sayısı
- **projects** (*int*) – Proje sayısı
- **components** (*int*) – Bileşen sayısı
- **translations** (*int*) – Çeviri sayısı
- **languages** (*int*) – Kullanılan dil sayısı
- **checks** (*int*) – Tetiklenmiş kalite denetimleri sayısı
- **configuration_errors** (*int*) – Yapılandırma hataları sayısı
- **suggestions** (*int*) – Bekleyen öneri sayısı
- **celery_queues** (*object*) – Celery kuyruklarının uzunlukları. Bilgi almak için: [Celery ile arka plan görevlerini kullanmak](#)
- **name** (*string*) – Yapılandırılmış sunucu adı

1.12.19 Bildirim kancaları

Bildirim kancaları, dış uygulamaların sürüm denetimi sistemi deposunun güncellendiğini Weblate tarafına bildirmesini sağlar.

Depoları ayrı ayrı güncellemek için proje, bileşen ve çeviri depo uç noktalarını kullanabilirsiniz. Bilgi almak için `POST /api/projects/(string:project)/repository/`.

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

2.6 sürümünden beri kullanım dışı: Erişim denetimi listesi ile kısıtlanmış projeler için kimlik doğrulaması ile düzgün çalışan `POST /api/components/(string:project)/(string:component)/repository/` kullanın.

Bir bileşenin güncellenmesini tetikler (sürüm denetimi sisteminden çekme ve çeviri değişikliklerini tarama).

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

2.6 sürümünden beri kullanım dışı: Erişim denetimi listesi ile kısıtlanmış projeler için kimlik doğrulaması ile düzgün çalışan `POST /api/projects/(string:project)/repository/` kullanın.

Bir projedeki tüm bileşenlerin güncellenmesini tetikler (sürüm denetimi sisteminden çekme ve çeviri değişikliklerini tarama).

POST /hooks/github/

GitHub bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Not: GitHub, doğrudan Weblate bilgilendirmelerini destekler: Depo ayarlarından Weblate hizmet kancasını etkinleştirin ve adres olarak Weblate kurulumunuzun adresini yazın.

Ayrıca bakınız:

GitHub değişikliklerini kendiliğinden almak

GitHub bütünleştirmesini ayarlama yönergeleri için

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

GitHub internet kancaları ile ilgili genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/gitlab/

GitLab bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Ayrıca bakınız:

GitLab değişikliklerini kendiliğinden almak

GitLab bütünleştirmesini ayarlama yönergeleri için

<https://docs.gitlab.com/ee/user/project/integrations/webhooks.html>

GitLab internet kancaları ile ilgili genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/bitbucket/

Bitbucket bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Ayrıca bakınız:

Bitbucket değişikliklerini kendiliğinden almak

Bitbucket bütünleştirmesini ayarlama yönergeleri için

<https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/>

Bitbucket internet kancaları ile ilgili genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/pagure/

3.3 sürümünde geldi.

Pagure bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Ayrıca bakınız:

Pagure değişikliklerini kendiliğinden almak

Pagure bütünleştirmesini ayarlama yönergeleri için

https://docs.pagure.org/pagure/usage/using_webhooks.html

Pagure internet kancaları ile ilgili genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/azure/

3.8 sürümünde geldi.

Azure DevOps bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Not: Lütfen *Gönderilecek kaynak ayrıntıları* değerinin *Tümü* olarak ayarlandığından emin olun. Yoksa Weblate ile Azure deponuz eşleşemez.

Ayrıca bakınız:

Azure Repos değişikliklerini kendiliğinden almak

Azure bütünleştirmesini ayarlama yönergeleri için

https:

[//learn.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops](https://learn.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops)

Azure DevOps internet kancaları ile ilgili genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/gitea/

3.9 sürümünde geldi.

Gitea internet kancası bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Ayrıca bakınız:

Gitea depoları değişikliklerini kendiliğinden almak

Gitea bütünleştirmesini ayarlama yönergeleri için

<https://docs.gitea.io/en-us/webhooks/>

Gitea internet kancaları ile ilgili genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/gitee/

3.9 sürümünde geldi.

Gitee internet kancası bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Ayrıca bakınız:

Gitee depoları değişikliklerini kendiliğinden almak

Gitee bütünleştirmesini ayarlama yönergeleri için

<https://gitee.com/help/categories/40>

Gitee internet kancaları ile ilgili genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

1.12.20 Dışa aktarmak

Weblate, verileri farklı şekillerde işlemenizi sağlamak için çeşitli dışa aktarma seçenekleri sunar.

GET /exports/stats/(string: project) /
string: component/

Sorgu parametreleri

- **format** (*string*) – Çıktı biçimi: json ya da csv

2.6 sürümünden beri kullanım dışı: Lütfen bunun yerine `GET /api/components/(string:project)/(string:component)/statistics/` ve `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/` kullanın. Erişim denetim listesi kullanılan projelere de erişilmesini sağlar.

Belirli bir bileşenin istatistiklerini belirtilen biçimde alır.

Örnek istek:

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

Örnek yanıt:

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

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

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

},
{
  "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 RSS akışları

Çevirilerdeki değişiklikler RSS akışlarında dışa aktarılır.

GET /exports/rss/ (string: project) /
string: component/string: language/

Bir çevirideki son değişikliklerin RSS akışını alır.

GET /exports/rss/ (string: project) /
string: component/

Bir bileşendeki son değişikliklerin RSS akışını alır.

GET /exports/rss/ (string: project) /
 Bir projedeki son değişikliklerin RSS bildirimini alır.

GET /exports/rss/language/ (string: language) /
 Bir dildeki son değişikliklerin RSS akışını alır.

GET /exports/rss/
 Weblate kopyasındaki son değişikliklerin RSS akışını alır.

Ayrıca bakınız:

[Vikipedi üzerinde RSS](#)

1.13 Weblate istemcisi

1.13.1 Kurulum

Weblate istemcisi ayrıca alınır ve Python modülünü içerir. Şu komutları kullanabilmek için `wlc` kurmanız gerekir:

```
pip install wlc
```

1.13.2 Docker kullanımı

Weblate istemcisi, Docker kalıbı olarak da kullanılabilir.

Kalıp Docker Hub üzerinde bulunur: <https://hub.docker.com/r/weblate/wlc>

Kurulum:

```
docker pull weblate/wlc
```

Docker kapsayıcısı, varsayılan Weblate ayarlarını kullanır ve localhost ile dağıtılan API üzerine bağlanır. API URL ve API_KEY, Weblate tarafından kabul edilen bağımsız değişkenler kullanılarak yapılandırılabilir.

Kapsayıcıyı başlatma komutu şu söz dizimini kullanır:

```
docker run --rm weblate/wlc [WLC_ARGS]
```

Örnek:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

Yapılandırma dosyaları dosyanızı Docker kapsayıcısına aktarmak isteyebilirsiniz. En kolay yaklaşım geçerli klasörünüzü `/home/weblate` birimi olarak eklemektir:

```
docker run --volume $PWD:/home/weblate --rm weblate/wlc show
```

1.13.3 Başlarken

`wlc` yapılandırması `~/ .config/weblate` klasöründe bulunur (diğer konumlar ile ilgili bilgi almak için: *Yapılandırma dosyaları*). Lütfen ortamınızla uygun şekilde ayarlayın:

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

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

Ardından varsayılan sunucuda komutları yürütebilirsiniz:

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

Ayrıca bakınız:

Yapılandırma dosyaları

1.13.4 Özet

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

Komutlar aslında hangi işlemin yapılması gerektiğini belirtir.

1.13.5 Açıklama

Weblate Client, Weblate yönetiminin *Weblate REST API uygulaması* üzerinden yapılmasını sağlayan bir Python kütüphanesi ve komut satırı uygulamasıdır. Komut satırı uygulaması **wlc** komutu ile çağrılabilir ve *wlc* modülü üzerinde bulunur.

Argümanlar

Çıktı biçimini veya hangi Weblate kopyasının kullanılacağını tanımlamak için aşağıdaki bağımsız değişkenler kullanılabilir. Bunlar herhangi bir komuttan önce yazılmalıdır.

--format {csv,json,text,html}

Çıktı biçimini belirtin.

--url URL

API adresini belirtir. Yapılandırma dosyasındaki tüm değerlerin yerini alır. Bilgi almak için: *Yapılandırma dosyaları*. Adres /api/ ile bitmelidir. Örnek: `https://hosted.weblate.org/api/`.

--key KEY

Kullanılacak API kullanıcı anahtarını belirtir. Yapılandırma dosyasındaki tüm değerlerin yerini alır. Bilgi almak için: *Yapılandırma dosyaları*. Anahtarınızı Weblate profilinizde bulabilirsiniz.

--config PATH

Yapılandırma dosyası yolunu değiştirir. Bilgi almak için: *Yapılandırma dosyaları*.

--config-section SECTION

Kullanımdaki yapılandırma dosyası bölümünü değiştirir. Bilgi almak için: *Yapılandırma dosyaları*.

Komutlar

Kullanılabilecek komutlar şunlardır:

version

Geçerli sürümü yazdırır.

list-languages

Weblate üzerinde kullanılan dilleri listeler.

list-projects

Weblate üzerindeki projeleri listeler.

list-components

Weblate üzerindeki bileşenleri listeler.

list-translations

Weblate üzerindeki çevirileri listeler.

show

Weblate nesnesini (çeviri, bileşen ya da proje) görüntüler.

ls

Weblate nesnesini (çeviri, bileşen ya da proje) listeler.

commit

Bir Weblate nesnesinde (çeviri, bileşen ya da proje) yapılan değişiklikleri işler.

pull

Uzak depo değişikliklerini Weblate nesnesine (çeviri, bileşen ya da proje) çeker.

push

Weblate nesnesindeki (çeviri, bileşen ya da proje) değişiklikleri uzak depoya gönderir.

reset

0.7 sürümünde geldi: wlc 0.7 sürümünden başlayarak destekleniyor.

Uzak depoya eşleşmesi için Weblate nesnesindeki (çeviri, bileşen ya da proje) değişiklikleri sıfırlar.

cleanup

0.9 sürümünde geldi: wlc 0.9 sürümünden başlayarak destekleniyor.

Uzak depoya eşleşmesi için bir Weblate nesnesindeki (çeviri, bileşen ya da proje) izlenmeyen değişiklikleri siler.

repo

Belirli bir Weblate nesnesi (çeviri, bileşen ya da proje) için depo durumunu görüntüler.

stats

Belirli bir Weblate nesnesi (çeviri, bileşen ya da proje) için ayrıntılı istatistikleri görüntüler.

lock-status

0.5 sürümünde geldi: wlc 0.5 sürümünden başlayarak destekleniyor.

Kilit durumunu görüntüler.

lock

0.5 sürümünde geldi: wlc 0.5 sürümünden başlayarak destekleniyor.

Weblate bileşenini kilitleyerek üzerinde daha fazla çeviri yapılmasını engeller.

unlock

0.5 sürümünde geldi: wlc 0.5 sürümünden başlayarak destekleniyor.

Weblate bileşeninin kilidini açarak çeviri yapılmasını sağlar.

changes

0.7 sürümünde geldi: wlc 0.7 ve Weblate 2.10 sürümünden başlayarak destekleniyor.

Belirli bir nesnenin değişikliklerini görüntüler.

download

0.7 sürümünde geldi: wlc 0.7 sürümünden başlayarak destekleniyor.

Bir çeviri dosyasını indirir.

--convert

Dosya biçimini dönüştürür. Belirtilmemişse sunucuda herhangi bir dönüştürme yapılmaz ve dosya depodan olduğu gibi indirilir.

--output

Çıktının kaydedileceği dosyayı belirtir. Belirtilmezse stdout çıkışına yazdırılır.

upload

0.9 sürümünde geldi: wlc 0.9 sürümünden başlayarak destekleniyor.

Bir çeviri dosyasını yükler.

--overwrite

Yükleme yapılırken var olan çevirilerin üzerine yazar.

--input

İçeriği okunacak dosya. Belirtilmemişse stdın girişinden okunur.

--method

Kullanılacak yükleme yöntemi. Bilgi almak için: [İçe aktarma yöntemleri](#).

--fuzzy

Belirsiz (düzenlenecek olarak imlenmiş) dizgeleri işleme (*empty*, *process*, *approve*)

--author-name

Geçerli kullanıcı yerine kullanılacak yazar adı

--author-email

Geçerli kullanıcı yerine kullanılacak yazar e-posta adresi

İpucu: Komutların kullanımı ile ilgili ayrıntılı bilgi almak için `--help` seçeneğini kullanabilirsiniz. Örneğin: `wlc ls --help`.

1.13.6 Yapılandırma dosyaları

.weblate, .weblate.ini, weblate.ini

Projeye özel yapılandırma dosyası

C:\Users\NAME\AppData\weblate.ini

Windows üzerinde kullanıcı yapılandırma dosyası.

~/.config/weblate

Kullanıcı yapılandırma dosyası

/etc/xdg/weblate

Sistem geneli yapılandırma dosyası

Program XDG teknik özelliklerine uyar. Böylece `XDG_CONFIG_HOME` ya da `XDG_CONFIG_DIRS` ortam değişkenlerine göre yapılandırma dosyalarının konumunu ayarlayabilirsiniz. Windows üzerinde yapılandırma dosyası için yeğlenen konum `APPDATA` klasörüdür.

[weblate] bölümünde şu ayarlar yapılandırılabilir (bunu `--config-section` ile özelleştirebilirsiniz):

key

Weblate erişimi için kullanılacak API anahtarı.

url

API sunucusunun adresi. Varsayılan değer: `http://127.0.0.1:8000/api/`.

translation

Varsayılan çeviri yolu. Bileşen ya da proje.

Yapılandırma dosyası bir INI dosyasıdır. Örneğin:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

Ek API anahtarları [keys] bölümüne kaydedilebilir:

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

Bu seçenek, sürüm denetimi sistemi deposundaki `.weblate` yapılandırmasını kullanırken anahtarları kişisel ayarlarınızda tutmanızı sağlar. Böylece `wlc` hangi sunucuyla konuşması gerektiğini bilir.

1.13.7 Örnekler

Geçerli program sürümünü yazdır:

```
$ wlc version
version: 0.1
```

Tüm projeleri listele:

```
$ 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/
```

Çeviri dosyasını yükle:

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

Ayrıca wlc hangi proje üzerinde çalışacak onu da belirleyebilirsiniz:

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

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

Bu kurulumla, geçerli projede bekleyen değişiklikler kolayca işlenebilir:

```
$ wlc commit
```

1.14 Weblate Python API

1.14.1 Kurulum

Python API ayrıca alınır. Alabilmek için *Weblate istemcisi* (wlc) kurmanız gerekir.

```
pip install wlc
```


1.14.2 wlc

WeblateException

exception `wlc.WeblateException`

Tüm özel durumlar için temel sınıf.

Weblate

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

Parametreler

- **key** (*str*) – Kullanıcı anahtarı
- **url** (*str*) – API sunucusu adresi, belirtilmemişse varsayılan adres kullanılır
- **config** (`wlc.config.WeblateConfig`) – Yapılandırma nesnesi, diğer parametreleri değiştirir.

API erişim sınıfı. API anahtarını ve isteğe bağlı olarak API adresini tanımlar.

get (*path*)

Parametreler

path (*str*) – İstek yolu

Dönüş türü

object

Tek bir API GET çağrısı yapar.

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

Parametreler

path (*str*) – İstek yolu

Dönüş türü

object

Tek bir API GET çağrısı yapar.

1.14.3 wlc.config

WeblateConfig

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

Parametreler

section (*str*) – Kullanılacak yapılandırma bölümü

XDG teknik özelliklerine uygun yapılandırma dosyası işleyici.

load (*path*=None)

Parametreler

path (*str*) – Yapılandırmanın yükleneceği yol.

Yapılandırmayı bir dosyadan yükler. None olarak belirtilirse, XDG yapılandırma yolunuzda (/etc/xdg/wlc) bulunan *wlc* yapılandırma dosyasından (~/.config/wlc) yüklenir.

1.14.4 `wlc.main`

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

Parametreler

- **settings** (*list*) – Demetlerin listesi olarak geçersiz kılınacak ayarlar
- **stdout** (*object*) – stdout dosya nesnesi çıktıyı yazdırmak için varsayılan olarak `sys.stdout` kullanır
- **args** (*list*) – İşlenecek komut satırı bağımsız değişkenleri. Varsayılan olarak `sys.args` kullanır

Komut satırı arayüzü için ana giriş noktası.

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

main() tarafından kullanılan ana işleyicide *Command* sınıfını kaydeden dekoratör.

Command

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

Komutları çağıran ana sınıf.

2.1 Yapılandırma yönergeleri

2.1.1 Weblate kurulumu

Docker ile kurmak

Docker için hazırlanmış Weblate dağıtımını kullanarak kişisel Weblate bulut sunucunuzu saniyeler içinde çalışır duruma getirebilirsiniz. Tüm Weblate bağımlılıkları zaten katılmıştır. Varsayılan veri tabanı olarak PostgreSQL kurulmuştur.

Donanım gereksinimleri

Weblate, herhangi bir çağdaş donanımda sorunsuz çalışmalıdır, Tek bir sunucuda Weblate bileşenlerini (Weblate, veri tabanı ve site sunucusu) çalıştırmak için gerekli en az yapılandırma şu şekildedir:

- 3 GB RAM bellek
- 2 işlemci çekirdeği
- 1 GB depolama alanı

Bellek ne kadar çok olursa o kadar iyidir. Tüm düzeylerde (dosya sistemi, veri tabanı ve Weblate) ön bellek için kullanılır.

Eşzamanlı bulunacak kullanıcı sayısı, gerekli işlemci çekirdeği sayısını artırır. Yüzlerce bileşenden oluşan çeviriler için en az 4 GB RAM bellek bulunması önerilir.

Veri tabanı depolama alanı gereksinimi, barındırılacak 1 milyon sözcük için yaklaşık 300 MB olarak düşünebilir. Kopyalanmış depolar için gereken depolama alanı değişiklik gösterir. Ancak Weblate sığ kopyalama ile depo boyutlarını en az düzeyde tutmaya çalışır.

Not: Weblate kurulumunuz için gerçek gereksinimler, üzerinde bulunacak çevirilerin boyutuna bağlı olarak büyük ölçüde değişiklik gösterebilir.

Kurulum

Aşağıdaki örneklerde, `docker-compose` kurulumu yapılmış hazır bir Docker ortamınız olduğu varsayılmaktadır. Yönergeler için Docker belgelerine bakabilirsiniz.

1. weblate-docker deposunu kopyalayın:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Ayarlarınızı yaptığınız bir `docker-compose.override.yml` dosyası oluşturun. Ortam değişkenlerinin tam listesi için *Docker ortam değişkenleri* bölümüne bakabilirsiniz.

```
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
```

Not: `WEBLATE_ADMIN_PASSWORD` değişkeni ayarlanmamışsa, yönetici kullanıcı başta belirtilen rastgele bir parolayla oluşturulur.

Verilen örnekte, Weblate 80 numaralı bağlantı noktasını dinleyecek şekilde ayarlanmıştır. `docker-compose.override.yml` dosyasını düzenleyerek bağlantı noktasını değiştirebilirsiniz.

3. Weblate kapsayıcılarını başlatın:

```
docker-compose up
```

Weblate dağıtımınızın tadını çıkarın. `weblate` kapsayıcısına 80 numaralı bağlantı noktasından erişilebilir.

3.7.1-6 sürümünde değişti: 2019 Temmuz ayında (3.7.1-6 sürümünden başlayarak), kapsayıcılar kök kullanıcı olarak çalışmayacak biçimde değiştirildi. Açık bağlantı noktası 80 yerine 8080 olarak değiştirildi.

Ayrıca bakınız:

Yönetim komutlarını çağırmak

Docker hub etiketini seçmek

Docker hub üzerinde şu etiketleri kullanabilirsiniz. Kullanılabilecek etiketlerin tam listesi için <https://hub.docker.com/r/weblate/weblate/tags/> adresine bakabilirsiniz.

Etiket adı	Açıklama	Kullanım amacı
latest	Son etiketlenmiş sürümle eşleşen kararlı Weblate sürümü	Üretim ortamında güncellemeleri yapmak
<VERSION>-<ID>	Weblate kararlı sürümü	Üretim ortamında iyi tanımlanmış dağıtım
edge	Docker kapsayıcısındaki geliştirme değişiklikleriyle Weblate kararlı sürümü (güncelleştirilmiş bağımlılıklar gibi)	Deneme ortamında güncellemeleri yapmak
edge-<DATE>-<ID>	Docker kapsayıcısındaki geliştirme değişiklikleriyle Weblate kararlı sürümü (güncelleştirilmiş bağımlılıklar gibi)	Hazırlama ortamında iyi tanımlanmış dağıtım
bleeding	Git üzerinden Weblate geliştirme sürümü	Yakında eklenecek Weblate özelliklerini denemel için güncelleme ekleri
bleeding-<DATE>-<ID>	Git üzerinden Weblate geliştirme sürümü	Yaklaşan Weblate özelliklerini denemek için iyi tanımlanmış dağıtım

Her kalıp yayınlanmadan önce denenir. Bu nedenle *bleeding* sürümünün kullanımı bile oldukça güvenli olmalıdır.

HTTPS desteği ile Docker kapsayıcısı

Genel dağıtım yönergeleri için [Kurulum](#) bölümüne bakabilirsiniz. Bu bölümde yalnızca oradakinden farklı olan durumlar ele alınmıştır.

Kendi SSL sertifikalarınızı kullanmak

3.8-3 sürümünde geldi.

Kullanmak istediğiniz bir SSL sertifikanız varsa, dosyaları Weblate veri birimine yerleştirmeniz yeterlidir (bilgi almak için: `:ref:docker-volume`):

- `ssl/fullchain.pem` dosyasında gerekli sertifika kuruluşunun sertifikaları ile birlikte kullanılacak sertifika bulunur
- `ssl/privkey.pem` dosyasında kişisel anahtar bulunur

Bu dosyaların ikisinin de sahibi, docker kapsayıcısını başlatan kullanıcı ile aynı olmalı ve dosya izinleri 600 olarak ayarlanmış olmalıdır (yalnızca sahibi olan kullanıcı tarafından okunabilir ve yazılabilir).

Ek olarak, Weblate kapsayıcısı SSL bağlantılarını artık 4443 numaralı bağlantı noktasından karşılar. Docker compose değişikliklerine HTTPS için bağlantı noktası yönlendirmeyi de katmak istersiniz:

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

Aynı sunucuda başka siteler barındırıyorsanız, büyük olasılıkla 80 ve 443 numaralı bağlantı noktaları NGINX gibi bir ters vekil sunucu tarafından kullanılmaktadır. HTTPS bağlantısını NGINX üzerinden docker kapsayıcısına geçirmek için şu yapılandırmayı kullanabilirsiniz:

```
server {
    listen 443;
    listen [::]:443;
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

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>;
}
}

```

<SITE_URL>, <SITE> ve <EXPOSED_DOCKER_PORT> seçeneklerine ortamınızın gerçek değerlerini yazın.

Let's Encrypt ile SSL sertifikası sağlamak

Genel kurulumda kendiliğinden oluşturulan [Let's Encrypt](#) SSL sertifikalarını kullanmak istiyorsanız, ters bir HTTPS vekil sunucusu eklemeniz gerekir. Bunun için ek bir Docker kapsayıcısı olan [https-portal](#) kullanılır. Bu, `docker-compose-https.yml` dosyası kullanılarak yapılır. Ardından ayarlarınızla bir `docker-compose-https.override.yml` dosyası oluşturun:

```

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'

```

docker-compose komutunu her çağırdığınızda, her iki dosyayı da belirtmeniz ve ardından şunları yapmanız gerekir:

```

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

```

Docker kapsayıcısını yükseltmek

Genellikle yalnızca Weblate kapsayıcısını güncellemek ve PostgreSQL kapsayıcısını olduğu sürümde tutmak iyi bir fikirdir. Çünkü PostgreSQL yükseltmesi oldukça zorlu bir işlemdir ve çoğu durumda fazla bir yarar sağlamaz.

4.10-1 sürümünde değişti: Weblate 4.10-1 sürümünden bu yana, Docker kapsayıcısı PostgreSQL 10 ya da üzerindeki sürümlere gerek duyan Django 4.0 kullanır. Lütfen Weblate yükseltmesinden önce Docker sürümünü yükseltin. Bilgi almak için: [4.9 sürümünden 4.10 sürümüne yükseltmek](#) ve [PostgreSQL kapsayıcısını yükseltmek](#).

Bu işlemi, var olan `docker-compose` ögesine bağlı kalarak ve son kalıpları alıp yeniden başlatarak yapabilirsiniz:

```

# Fetch latest versions of the images
docker-compose pull
# Stop and destroy the containers

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
docker-compose down
# Spawn new containers in the background
docker-compose up -d
# Follow the logs during upgrade
docker-compose logs -f
```

Weblate veri tabanının başlangıçta kendiliğinden aktarılması ve el ile yapılacak ek işlemlere gerek kalmaması beklenir.

Not: Weblate ana sürümler arasında yükseltme işlemini desteklemez. Örneğin, 3.x serisini kullanıyor ve 4.x serisine yükseltmek istiyorsanız, önce son 4.0.x-y kalıbına (yazı hazırlandığında 4.0.4-5 sürümüydü) yükseltin. Bu kalıp önce aktarımı yapıp ardından yeni sürümlere yükseltmeyi sağlar.

`docker-compose` deposunu da güncellemek isteyebilirsiniz. Ancak çoğu durumda buna gerek olmaz. PostgreSQL sunucusunu yükseltmek için [PostgreSQL kapsayıcısını yükseltmek](#) sayfasına bakabilirsiniz.

PostgreSQL kapsayıcısını yükseltmek

PostgreSQL kapsayıcıları sürümler arasında kendiliğinden yükseltmeyi desteklemez. El ile yükseltmeniz gerekir. Aşağıdaki adımlarda, yükseltme seçeneklerinden biri gösterilmektedir.

Ayrıca bakınız:

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

1. Weblate kapsayıcısını durdurun:

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

2. Veri tabanını yedekleyin:

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

3. Veri tabanı kapsayıcısını durdurun:

```
docker-compose stop database
```

4. PostgreSQL birimini kaldırın:

```
docker-compose rm -v database
docker volume remove weblate-docker_postgres-data
```

5. `docker-compose.yml` dosyasını yeni PostgreSQL sürümünü kullanarak şekilde ayarlayın.

6. Veri tabanı kapsayıcısını başlatın:

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

7. Veri tabanını yedekten geri yükleyin:

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

8. (İsteğe bağlı) Weblate kullanıcısının parolasını güncelleyin. Parolaları depolama yöntemi değiştirildiği için PostgreSQL 14 ya da 15 sürümüne geçiş yaparken bu işlemi yapmanız gerekebilir:

```
docker-compose exec -T database psql --username weblate --dbname postgres -c
↳ "ALTER USER weblate WITH PASSWORD 'weblate'"
```

9. Kalan tüm kapsayıcıları başlatın:

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

Yönetici olarak oturum açın

Kapsayıcı kurulumundan sonra, `WEBLATE_ADMIN_PASSWORD` içinde belirtilen parolayla ya da ayarlanmamışsa başlangıçta oluşturulan rastgele bir parolayla `admin` kullanıcısı ile oturum açabilirsiniz.

`admin` parolasını sıfırlamak için `WEBLATE_ADMIN_PASSWORD` seçeneğine yeni parolayı ayarlayın ve kapsayıcıyı yeniden başlatın.

Ayrıca bakınız:

`WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

İşlem sayısı ve bellek kullanımı

Hem uWSGI hem de Celery için çalışan işlem sayısı, işlemci sayısına göre kendiliğinden belirlenir. Bu seçim, çoğu bulut sanal makinesi için iyi çalışır. Bu makinelerde genellikle az sayıda işlemci ve iyi miktarda bellek bulunur.

Çok fazla işlemci çekirdeğiniz varsa ve bellek sorunlarından kurtulursanız, işlem sayısını azaltmayı deneyin:

```
environment:
  WEBLATE_WORKERS: 2
```

Ayrıca her işleyici kategorisi için ince ayar yapabilirsiniz:

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

Ayrıca bakınız:

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

Yatay ölçeklendirme

4.6 sürümünde geldi.

Hizmeti yatay olarak ölçeklemek için birden çok Weblate kapsayıcısı çalıştırabilirsiniz. `/app/data` biriminin tüm kapsayıcılar tarafından paylaşılması gerekir. Bunun için GlusterFS gibi küme dosya sistemlerinin kullanılması önerilir. `/app/cache` birimi her kapsayıcı için ayrı olmalıdır.

Her Weblate kapsayıcısı, `WEBLATE_SERVICE` ortam değişkenini kullanarak rolü tanımlar. Bazı hizmetlerin kümede yalnızca bir kez çalışması gerektiğinden ve hizmetlerin sıralaması önemli olduğundan, belgelere dikkatle uyun.

Örnek kurulumu `docker-compose` deposunda `docker-compose-split.yml` olarak bulabilirsiniz.

Docker ortam değişkenleri

Weblate *Yapılandırma* öğelerinin çoğu, Docker kapsayıcısında aşağıda açıklanan ortam değişkenleri ile ayarlanabilir. Docker ortam değişkenlerinde bulunmayan bir ayar yapmanız gerekiyorsa bilgi almak için *Ortam değişkenlerinin yapılandırması*.

Genel ayarlar

WEBLATE_DEBUG

DEBUG kullanarak Django hata ayıklama kipini yapılandırır.

Örnek:

```
environment:
  WEBLATE_DEBUG: 1
```

Ayrıca bakınız:

Hata ayıklama kipini kapatın

WEBLATE_LOGLEVEL

Günlük kayıtlarının ayrıntı düzeyini yapılandırır.

WEBLATE_LOGLEVEL_DATABASE

Veri tabanı sorguları günlük kayıtlarının ayrıntı düzeyini yapılandırır.

WEBLATE_SITE_TITLE

Tüm sayfaların üst bilgisi olarak görüntülenecek site başlığını değiştirir.

WEBLATE_SITE_DOMAIN

Site etki alanını yapılandırır. Bu parametre gereklidir.

Ayrıca bakınız:

*Doğru site etki alanını ayarlayın, *SITE_DOMAIN**

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Site yöneticisinin adını ve e-posta adresini yapılandırır. Hem *ADMINS* ayarını hem de *admin* kullanıcısını oluşturmak için kullanılır (bilgi almak için: *WEBLATE_ADMIN_PASSWORD*).

Örnek:

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

Ayrıca bakınız:

*Yönetici olarak oturum açın, Yöneticileri düzgün şekilde yapılandırın, *ADMINS**

WEBLATE_ADMIN_PASSWORD

admin kullanıcısının parolasını ayarlar.

- Ayarlanmamışsa ve *admin* kullanıcısı yoksa, kapsayıcı ilk kez başlatıldığında görüntülenen rastgele bir parola ile oluşturulur.
- Ayarlanmamışsa ve *admin* kullanıcısı varsa, hiçbir işlem gerçekleştirilmez.
- Ayarlanmışsa, *admin* kullanıcısı her kapsayıcı başlangıcında *WEBLATE_ADMIN_PASSWORD*, *WEBLATE_ADMIN_NAME* ve *WEBLATE_ADMIN_EMAIL* seçeneklerine göre ayarlanır.

Uyarı: Parolayı yapılandırma dosyasında depolamak bir güvenlik riski oluşturabilir. Bu değişkeni yalnızca ilk kurulum için (ya da Weblate ilk kez başlatıldığında rastgele parola oluşturmaya izin verin) ya da parola kurtarma için kullanmayı değerlendirin.

Ayrıca bakınız:

Yönetici olarak oturum açın, `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_PASSWORD_FILE`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

WEBLATE_ADMIN_PASSWORD_FILE

Admin kullanıcısının parolasının bulunduğu bir dosyanın yolunu ayarlar.

Ayrıca bakınız:

`WEBLATE_ADMIN_PASSWORD`

WEBLATE_SERVER_EMAIL

Hata iletilerinin gönderileceği e-posta adresi.

Ayrıca bakınız:

*`SERVER_EMAIL`, *E-posta gönderimini yapılandırın**

WEBLATE_DEFAULT_FROM_EMAIL

Giden e-postaların adresini yapılandırır.

Ayrıca bakınız:

*`DEFAULT_FROM_EMAIL`, *E-posta gönderimini yapılandırın**

WEBLATE_CONTACT_FORM

İletişim formu davranışını yapılandırır. Bilgi almak için: *`CONTACT_FORM`*.

WEBLATE_ALLOWED_HOSTS

`ALLOWED_HOSTS` kullanarak izin verilen HTTP sunucu adlarını yapılandırır.

Varsayılan değer: *. Tüm sunucu adlarına izin verir.

Örnek:

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

Ayrıca bakınız:

*`ALLOWED_HOSTS`, *Allowed hosts kurulumu*, *Doğru site etki alanını ayarlayın**

WEBLATE_REGISTRATION_OPEN

`REGISTRATION_OPEN` seçeneğini değiştirerek hesap açma özelliğini etkinleştirir.

Örnek:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

`REGISTRATION_ALLOW_BACKENDS` seçeneği ile yeni hesap açmak için hangi kimlik doğrulama yöntemlerinin kullanılabileceğini yapılandırır.

Örnek:

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

WEBLATE_REGISTRATION_REBIND

4.16 sürümünde geldi.

`REGISTRATION_REBIND` seçeneğini yapılandırır.

WEBLATE_TIME_ZONE

Weblate üzerinde kullanılacak saat dilimini yapılandırır. Bilgi almak için: `TIME_ZONE`.

Not: Docker kapsayıcısının saat dilimini değiştirmek için TZ ortam değişkenini kullanın.

Örnek:

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

WEBLATE_ENABLE_HTTPS

Weblate uygulamasının ters HTTPS vekil sunucu arkasında çalıştığını varsaymasını sağlar. Weblate e-posta ve API bağlantılarında HTTPS kullanılmasını sağlar ya da çerezler için güvenli imlerini ayarlar.

İpucu: Olası uyarılar için `ENABLE_HTTPS` belgelerine bakabilirsiniz.

Not: Bu seçenek, Weblate kapsayıcısının HTTPS bağlantılarını kabul etmesini sağlamaz. Bunu da yapılandırmanız gerekir. Bilgi almak için: *HTTPS desteği ile Docker kapsayıcısı*.

Örnek:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Ayrıca bakınız:

`ENABLE_HTTPS` Doğru site etki alanını ayarlayın, `WEBLATE_SECURE_PROXY_SSL_HEADER`

WEBLATE_INTERLEDGER_PAYMENT_POINTERS

4.12.1 sürümünde geldi.

Weblate belgelerinin başındaki `meta[name=monetization]` alanının ayarlanmasını sağlar. Birden çok seçenek belirtilmişse, rastgele olarak biri seçilir.

Ayrıca bakınız:

`INTERLEDGER_PAYMENT_POINTERS`

WEBLATE_IP_PROXY_HEADER

IP adresinin Weblate tarafından herhangi bir HTTP üst bilgisinden alınmasını sağlar. Weblate kapsayıcısının önünde ters vekil sunucu varsa bunu kullanın.

`IP_BEHIND_REVERSE_PROXY` özelliğini etkinleştirir ve `IP_PROXY_HEADER` ögesini ayarlar.

Not: Biçim, Django beklentilerine uygun olmalıdır. Django ham HTTP üst bilgi adlarını şu biçimde **dönüştürür**:

- tüm karakterleri büyük harfe dönüştürür
- tire karakterini alt çizgi karakterine dönüştürür
- başa HTTP_ ön ekini ekler

Böylece X-Forwarded-For değeri HTTP_X_FORWARDED_FOR ile eşleştirilir.

Örnek:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

Bir isteği belirten HTTP üst bilgi/değer ikilisini temsil eden bir demet güvenlidir. Weblate, standart HTTPS üst bilgilerini geçirmeyen SSL sonlandırma işlemi yapan bir ters vekil sunucu arkasında ise bu gereklidir.

Örnek:

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

Ayrıca bakınız:

SECURE_PROXY_SSL_HEADER

WEBLATE_REQUIRE_LOGIN

Tüm Weblate üzerinde kimlik doğrulamasını zorunlu kılmak için *REQUIRE_LOGIN* seçeneğini etkinleştirir.

Örnek:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

LOGIN_REQUIRED_URLS_EXCEPTIONS seçeneğini kullanarak tüm Weblate kurulumu için gereken kimlik doğrulama adresi istisnalarını ekler.

Tüm ayarları değiştirebilir ya da ADD ve REMOVE seçeneklerini kullanarak varsayılan değeri değiştirebilirsiniz.

WEBLATE_GOOGLE_ANALYTICS_ID

GOOGLE_ANALYTICS_ID seçeneğini değiştirerek Google Analytics kimliğini yapılandırır.

WEBLATE_GITHUB_USERNAME

WEBLATE_GITHUB_TOKEN

WEBLATE_GITHUB_HOST

GITHUB_CREDENTIALS seçeneğini değiştirerek GitHub çekme istekleri bütünleştirmesini yapılandırır.

Ayrıca bakınız:

GitHub çekme istekleri

WEBLATE_GITLAB_USERNAME

WEBLATE_GITLAB_TOKEN

WEBLATE_GITLAB_HOST

GITLAB_CREDENTIALS seçeneğini değiştirerek GitLab birleştirme istekleri bütünleştirmesini yapılandırır.

Örnek:

```
WEBLATE_GITLAB_USERNAME=weblate
WEBLATE_GITLAB_HOST=gitlab.com
WEBLATE_GITLAB_TOKEN=token
```

Ayrıca bakınız:

GitLab birleştirme istekleri

WEBLATE_GITEA_USERNAME

WEBLATE_GITEA_TOKEN

WEBLATE_GITEA_HOST

GITEA_CREDENTIALS seçeneğini değiştirerek Gitea çekme istekleri bütünleştirmesini yapılandırır.

Ayrıca bakınız:

Gitea çekme isteği

WEBLATE_PAGURE_USERNAME

WEBLATE_PAGURE_TOKEN

WEBLATE_PAGURE_HOST

PAGURE_CREDENTIALS seçeneğini değiştirerek Pagure birleştirme istekleri bütünleştirmesini yapılandırır.

Ayrıca bakınız:

Pagure birleştirme istekleri

WEBLATE_BITBUCKETSERVER_USERNAME

WEBLATE_BITBUCKETSERVER_TOKEN

WEBLATE_BITBUCKETSERVER_HOST

BITBUCKETSERVER_CREDENTIALS seçeneğini değiştirerek Bitbucket sunucusu birleştirme istekleri bütünleştirmesini yapılandırır.

Ayrıca bakınız:

Bitbucket sunucusu çekme isteği

WEBLATE_DEFAULT_PULL_MESSAGE

DEFAULT_PULL_MESSAGE seçeneğini değiştirerek API ile yapılan çekme isteklerinin varsayılan başlığını ve iletiyi yapılandırır

Ayrıca bakınız:

DEFAULT_PULL_MESSAGE

WEBLATE_SIMPLIFY_LANGUAGES

Dil basitleştirme ilkesini yapılandırır. Bilgi almak için: `:setting: SIMPLIFY_LANGUAGES`.

WEBLATE_DEFAULT_ACCESS_CONTROL

Yeni projeler için varsayılan *Erişim denetimi* ögesini yapılandırır. Bilgi almak için: *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Yeni bileşenler için *Kısıtlı erişim* seçeneğinin varsayılan değerini yapılandırır. Bilgi almak için: *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Yeni bileşenler için *Çevirilerin yayılmasını sağlamak* seçeneğinin varsayılan değerini yapılandırır. Bilgi almak için: *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

DEFAULT_COMMITER_EMAIL seçeneğini yapılandırır.

WEBLATE_DEFAULT_COMMITER_NAME

DEFAULT_COMMITER_NAME seçeneğini yapılandırır.

WEBLATE_DEFAULT_SHARED_TM

DEFAULT_SHARED_TM seçeneğini yapılandırır.

WEBLATE_AKISMET_API_KEY

Akismet API anahtarını yapılandırır. Bilgi almak için: *AKISMET_API_KEY*.

WEBLATE_GPG_IDENTITY

İşlemlerin GPG imzalamasını yapılandırır. Bilgi almak için: *WEBLATE_GPG_IDENTITY*.

Ayrıca bakınız:

Git işlemlerini GnuPG ile imzalamak

WEBLATE_URL_PREFIX

Weblate ortamının çalıştığı adres ön ekini yapılandırır. Bilgi almak için: *URL_PREFIX*.

WEBLATE_SILENCED_SYSTEM_CHECKS

Görüntülenmesini istemediğiniz denetimleri yapılandırır. Bilgi almak için: *SILENCED_SYSTEM_CHECKS*.

WEBLATE_CSP_SCRIPT_SRC

WEBLATE_CSP_IMG_SRC

WEBLATE_CSP_CONNECT_SRC

WEBLATE_CSP_STYLE_SRC

WEBLATE_CSP_FONT_SRC

Content-Security-Policy HTTP üst bilgisinin özelleştirilmesini sağlar.

Ayrıca bakınız:

İçerik güvenliği ilkesi, *CSP_SCRIPT_SRC*, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*, *CSP_STYLE_SRC*, *CSP_FONT_SRC*

WEBLATE_LICENSE_FILTER

LICENSE_FILTER seçeneğini yapılandırır.

WEBLATE_LICENSE_REQUIRED

LICENSE_REQUIRED seçeneğini yapılandırır

WEBLATE_WEBSITE_REQUIRED

WEBSITE_REQUIRED seçeneğini yapılandırır

WEBLATE_HIDE_VERSION

HIDE_VERSION seçeneğini yapılandırır.

WEBLATE_BASIC_LANGUAGES

BASIC_LANGUAGES seçeneğini yapılandırır.

WEBLATE_DEFAULT_AUTO_WATCH

DEFAULT_AUTO_WATCH seçeneğini yapılandırır.

WEBLATE_RATELIMIT_ATTEMPTS

WEBLATE_RATELIMIT_LOCKOUT

WEBLATE_RATELIMIT_WINDOW

4.6 sürümünde geldi.

Hız sınırlayıcıyı yapılandırır.

İpucu: Yapılandırmayı herhangi bir hız sınırlayıcı kapsamı için ayarlayabilirsiniz. Bunu yapmak için, `:ref:rate-limit`` içinde belirtilen ayarlardan herhangi birine ``WEBLATE_ ön ekini ekleyin.

Ayrıca bakınız:

Hızı sınırlamak, *RATELIMIT_ATTEMPTS*, *RATELIMIT_WINDOW*, *RATELIMIT_LOCKOUT*

WEBLATE_API_RATELIMIT_ANON

WEBLATE_API_RATELIMIT_USER

4.11 sürümünde geldi.

API hız sınırlamasını yapılandırır. Varsayılan değer anonim kullanıcılar için 100/gün ve kimliği doğrulanmış kullanıcılar için ``5000/saat` biçimindedir.

Ayrıca bakınız:

API hız sınırlaması

WEBLATE_ENABLE_HOOKS

4.13 sürümünde geldi.

ENABLE_HOOKS seçeneğini yapılandırır.

WEBLATE_ENABLE_AVATARS

4.6.1 sürümünde geldi.

ENABLE_AVATARS seçeneğini yapılandırır.

WEBLATE_AVATAR_URL_PREFIX

4.15 sürümünde geldi.

AVATAR_URL_PREFIX seçeneğini yapılandırır.

WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

4.9 sürümünde geldi.

LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH seçeneğini yapılandırır.

WEBLATE_SSH_EXTRA_ARGS

4.9 sürümünde geldi.

SSH_EXTRA_ARGS seçeneğini yapılandırır.

WEBLATE_BORG_EXTRA_ARGS

4.9 sürümünde geldi.

BORG_EXTRA_ARGS seçeneğini yapılandırır.

WEBLATE_ENABLE_SHARING

4.14.1 sürümünde geldi.

ENABLE_SHARING seçeneğini yapılandırır.

WEBLATE_EXTRA_HTML_HEAD

4.15 sürümünde geldi.

EXTRA_HTML_HEAD seçeneğini yapılandırır.

WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE

4.15 sürümünde geldi.

PRIVATE_COMMIT_EMAIL_TEMPLATE seçeneğini yapılandırır.

WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN

4.15 sürümünde geldi.

PRIVATE_COMMIT_EMAIL_OPT_IN seçeneğini yapılandırır.

WEBLATE_UNUSED_ALERT_DAYS

4.17 sürümünde geldi.

`UNUSED_ALERT_DAYS` seçeneğini yapılandırır.

WEBLATE_CORS_ALLOWED_ORIGINS

4.16 sürümünde geldi.

Belirtilen kaynaklardan CORS istekleri yapılabilmesini sağlar.

Örnek:

```
environment:
  WEBLATE_CORS_ALLOWED_ORIGINS: https://example.com,https://weblate.org
```

CLIENT_MAX_BODY_SIZE

4.16.3 sürümünde geldi.

İç site sunucusu tarafından kabul edilecek en büyük gövde boyutu seçeneğini yapılandırır.

```
environment:
  CLIENT_MAX_BODY_SIZE: 200m
```

İpucu: Bu değişken, *Let's Encrypt ile SSL sertifikası sağlamak* içinde kullanılan üçüncü taraf kapsayıcısıyla paylaşıldığı için kasıtlı olarak WEBLATE_ ön ekini kullanmaz.

Kendiliğinden öneri ayarları

4.13 sürümünde değişti: Kendiliğinden öneri hizmetleri artık kullanıcı arayüzünden yapılandırılıyor. Bilgi almak için: [Kendiliğinden önerileri yapılandırmak](#).

Var olan ortam değişkenleri Weblate 4.13 sürümüne geçiş yapılırken içe aktarılır. Ancak bunların değiştirilmesinin başkaca bir etkisi olmaz.

Kimlik doğrulama ayarları

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

LDAP sunucusunun bind parolasını içeren dosyanın yolu.

Ayrıca bakınız:

`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**

LDAP kimlik doğrulama yapılandırması.

Doğrudan bağlama örneği:

```
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
```

Arama ve bağlama örneği:

```
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
```

Birleşik arama ve bağlama örneği:

```
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
```

Active Directory üzerinde arama ve bağlama örneği:

```
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)
```

Ayrıca bakınız:*LDAP kimlik doğrulaması*

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

GitHub kimlik doğrulaması seçeneğini etkinleştirir.

Bitbucket

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET

Bitbucket kimlik doğrulaması seçeneğini etkinleştirir.

Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY

WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET

Facebook OAuth 2 seçeneğini etkinleştirir.

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

Google OAuth 2 seçeneğini etkinleştirir.

GitLab

`WEBLATE_SOCIAL_AUTH_GITLAB_KEY`

`WEBLATE_SOCIAL_AUTH_GITLAB_SECRET`

`WEBLATE_SOCIAL_AUTH_GITLAB_API_URL`

GitLab OAuth 2 seçeneğini etkinleştirir.

Gitea

`WEBLATE_SOCIAL_AUTH_GITEA_API_URL`

`WEBLATE_SOCIAL_AUTH_GITEA_KEY`

`WEBLATE_SOCIAL_AUTH_GITEA_SECRET`

Gitea kimlik doğrulaması kullanılsın.

Azure Active Directory

`WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY`

`WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET`

Azure Active Directory kimlik doğrulamasını etkinleştirir. Bilgi almak için: *Microsoft Azure Active Directory*.

Kiracı destekli Azure Active Directory

`WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY`

`WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET`

`WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID`

Kiracı destekli Azure Active Directory kimlik doğrulamasını etkinleştirir. Bilgi almak için: *Microsoft Azure Active Directory*.

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`

Keycloak kimlik doğrulamasını etkinleştirir. Bilgi almak için: *Belgeler*.

Linux hizmet sağlayıcıları

Şu değişkenleri bir değere ayarlayarak Linux üreticilerinin kimlik doğrulama hizmetleri ile kimlik doğrulamayı etkinleştirebilirsiniz.

WEBLATE_SOCIAL_AUTH_FEDORA

WEBLATE_SOCIAL_AUTH_OPENSUSE

WEBLATE_SOCIAL_AUTH_OPENINFRA

WEBLATE_SOCIAL_AUTH_UBUNTU

Slack

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Slack kimlik doğrulamasını etkinleştirir. Bilgi almak için: [Slack](#).

OpenID Connect

4.13-1 sürümünde geldi.

WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT

WEBLATE_SOCIAL_AUTH_OIDC_KEY

WEBLATE_SOCIAL_AUTH_OIDC_SECRET

WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY

Genel OpenID Connect bütünleştirmesini yapılandırır.

Ayrıca bakınız:

[OIDC \(OpenID Connect\)](#)

SAML

Kendiliğinden imzalanan SAML anahtarları, ilk kapsayıcı başlatıldığında kendiliğinden oluşturulur. Kendi anahtarlarınızı kullanmak isterseniz, sertifikayı ve kişisel anahtarı `/app/data/ssl/saml.crt` ve `/app/data/ssl/saml.key` dosyalarına yerleştirin.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT

WEBLATE_SAML_IDP_IMAGE

WEBLATE_SAML_IDP_TITLE

SAML kimlik hizmeti sağlayıcısı ayarları. Bilgi almak için: [SAML kimlik doğrulaması](#).

Diğer kimlik doğrulama ayarları

WEBLATE_NO_EMAIL_AUTH

Herhangi bir değere ayarlandığında e-posta kimlik doğrulamasını devre dışı bırakır. Bilgi almak için: [Parola ile kimlik doğrulamayı kapatmak](#).

PostgreSQL veri tabanı kurulumu

Veri tabanı `docker-compose.yml` tarafından oluşturulur. Bu nedenle bu ayarlar hem Weblate hem de PostgreSQL kapsayıcılarını etkiler.

Ayrıca bakınız:

[Weblate için veri tabanı kurulumu](#)

POSTGRES_PASSWORD

PostgreSQL parolası.

POSTGRES_PASSWORD_FILE

PostgreSQL parolasının bulunduğu dosyanın yolu. `POSTGRES_PASSWORD` alternatif olarak kullanın.

POSTGRES_USER

PostgreSQL kullanıcı adı.

POSTGRES_DATABASE

PostgreSQL veri tabanı adı.

POSTGRES_HOST

PostgreSQL sunucusunun adı ya da IP adresi. Varsayılan değer: `database`.

POSTGRES_PORT

PostgreSQL sunucu bağlantı noktası. Varsayılan değer: Yok (varsayılan değeri kullanır).

POSTGRES_SSL_MODE

PostgreSQL sunucusunun SSL bağlantılarını nasıl işleyeceğini yapılandırın. Kullanılabilecek seçenekler için [SSL kipi açıklamaları](#) bölümüne bakabilirsiniz

POSTGRES_ALTER_ROLE

Aktarım sırasında değiştirilecek rol adını yapılandırır. Bilgi almak için: [Weblate yapılandırmasını PostgreSQL kullanacak biçimde ayarlamak](#).

POSTGRES_CONN_MAX_AGE

4.8.1 sürümünde geldi.

Saniye ve tamsayı olarak veri tabanı bağlantısının yaşam süresi. Her isteğin sonunda veri tabanı bağlantılarını kapatılması için 0 yazın (varsayılan davranış).

Bağlantı kalıcılığını etkinleştirmek genellikle veri tabanına daha açık bir bağlantı kurulmasına neden olur. Lütfen bu seçeneği etkinleştirmeden önce veri tabanı yapılandırmanızı ayarlayın.

Örnek yapılandırma:

```
environment:
  POSTGRES_CONN_MAX_AGE: 3600
```

Ayrıca bakınız:

`CONN_MAX_AGE`, [Persistent connections](#)

POSTGRES_DISABLE_SERVER_SIDE_CURSORS

4.9.1 sürümünde geldi.

Veri tabanındaki sunucu tarafı imleçlerini devre dışı bırakın. Bu uygulama, bazı **pgbouncer** kurulumlarında gereklidir.

Örnek yapılandırma:

```
environment :  
  POSTGRES_DISABLE_SERVER_SIDE_CURSORS: 1
```

Ayrıca bakınız:

`DISABLE_SERVER_SIDE_CURSORS`, Transaction pooling and server-side cursors

Veri tabanı yedekleme ayarları

Ayrıca bakınız:

Yedeklemeler için dökümü yapılan veriler

WEBLATE_DATABASE_BACKUP

`DATABASE_BACKUP` kullanarak günlük veri tabanı dökümünü yapılandırır. Varsayılan değer: `plain`.

Ön bellek sunucusu kurulumu

Weblate, Redis kullanılmasını önemle önerir ve Weblate Docker üzerinde çalıştırılırken bir Redis kopyası sağlamanız gerekir.

Ayrıca bakınız:

Ön bellek özelliğini açın

REDIS_HOST

Redis sunucusunun adı ya da IP adresi. Varsayılan değer: `cache`.

REDIS_PORT

Redis sunucusunun bağlantı noktası. Varsayılan değer: `6379`.

REDIS_DB

Redis veri tabanı numarası. Varsayılan değer: `1`.

REDIS_PASSWORD

Redis sunucusunun parolası. Varsayılan olarak kullanılmaz.

REDIS_PASSWORD_FILE

Redis sunucu parolasının bulunduğu dosyanın yolu.

Ayrıca bakınız:

`REDIS_PASSWORD`

REDIS_TLS

Redis bağlantısı için SSL kullanımını etkinleştirir.

REDIS_VERIFY_SSL

Redis bağlantısı için SSL sertifikası doğrulamasını devre dışı bırakmak için kullanılabilir.

E-posta sunucusu kurulumu

Giden e-postanın çalışmasını sağlamak için, bir e-posta sunucusu belirtmeniz gerekir.

Örnek TLS yapılandırması:

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

Örnek SSL yapılandırması:

```
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
```

Ayrıca bakınız:

Giden e-postayı yapılandırmak

WEBLATE_EMAIL_HOST

E-posta sunucusunun adı ya da IP adresi.

Ayrıca bakınız:

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

WEBLATE_EMAIL_PORT

E-posta sunucusunun bağlantı noktası. Varsayılan değer: 25.

Ayrıca bakınız:

`EMAIL_PORT`

WEBLATE_EMAIL_HOST_USER

E-posta ile kimlik doğrulama kullanıcısı.

Ayrıca bakınız:

`EMAIL_HOST_USER`

WEBLATE_EMAIL_HOST_PASSWORD

E-posta ile kimlik doğrulama parolası.

Ayrıca bakınız:

`EMAIL_HOST_PASSWORD`

WEBLATE_EMAIL_HOST_PASSWORD_FILE

E-posta ile kimlik doğrulama parolasını içeren dosyanın yolu.

Ayrıca bakınız:

`WEBLATE_EMAIL_HOST_PASSWORD`

WEBLATE_EMAIL_USE_SSL

SMTP sunucusuyla iletişim için örtülü TLS (güvenli) bağlantısının kullanılıp kullanılmayacağını yapılandırır. Çoğu e-posta belgesinde, bu tür TLS bağlantısı SSL olarak adlandırılır. Genellikle 465 numaralı bağlantı noktasıyla kullanılır. Sorun yaşıyorsanız `WEBLATE_EMAIL_USE_TLS` açık TLS ayarına bakın.

4.11 sürümünde değişti: SSL/TLS desteği `WEBLATE_EMAIL_PORT` seçeneğine göre kendiliğinden etkinleştirilir.

Ayrıca bakınız:

`WEBLATE_EMAIL_PORT`, `WEBLATE_EMAIL_USE_TLS`, `EMAIL_USE_SSL`

WEBLATE_EMAIL_USE_TLS

SMTP sunucusuyla iletişim için TLS (güvenli) bağlantısının kullanılıp kullanılmayacağını yapılandırır. Bu seçenek, genellikle 587 ya da 25 numaralı bağlantı noktalarında bulunan açık TLS bağlantıları için kullanılır. Bağlantıların askıda kalması sorunu yaşıyorsanız envvar: `WEBLATE_EMAIL_USE_SSL` seçeneğindeki örtülü TLS ayarına bakın.

4.11 sürümünde değişti: SSL/TLS desteği `WEBLATE_EMAIL_PORT` seçeneğine göre kendiliğinden etkinleştirilir.

Ayrıca bakınız:

`WEBLATE_EMAIL_PORT`, `WEBLATE_EMAIL_USE_SSL`, `EMAIL_USE_TLS`

WEBLATE_EMAIL_BACKEND

Django arka ucunu e-posta göndermek için yapılandırır.

Ayrıca bakınız:

E-posta gönderimini yapılandırın, `EMAIL_BACKEND`

WEBLATE_AUTO_UPDATE

Depoların Weblate tarafından güncellenip güncellenmeyeceğini ve nasıl güncelleneceğini yapılandırır.

Ayrıca bakınız:

`AUTO_UPDATE`

Not: Bu bir Boolean ayarıdır ("true" ya da "false" kullanın).

Site bütünleştirilmesi

WEBLATE_GET_HELP_URL

`GET_HELP_URL` seçeneğini yapılandırır.

WEBLATE_STATUS_URL

`STATUS_URL` seçeneğini yapılandırır.

WEBLATE_LEGAL_URL

`LEGAL_URL` seçeneğini yapılandırır.

WEBLATE_PRIVACY_URL

`PRIVACY_URL` seçeneğini yapılandırır.

Hataları bildirmek

Kurulumdaki hataları sistematik olarak toplamanız önerilir. Bilgi almak için: *Hata raporlarını derlemek*.

Rollbar desteğini etkinleştirmek için şunları ayarlayın:

ROLLBAR_KEY

Rollbar ileti sunucunuzun erişim kodu.

ROLLBAR_ENVIRONMENT

Rollbar ortamınız. Varsayılan değer: `production`.

Sentry desteğini etkinleştirmek için şunları ayarlayın:

SENTRY_DSN

Sentry DSN değerinizi.

SENTRY_ENVIRONMENT

Sentry ortamınız (isteğe bağlı).

Yerelleştirme CDN**WEBLATE_LOCALIZE_CDN_URL****WEBLATE_LOCALIZE_CDN_PATH**

4.2.1 sürümünde geldi.

JavaScript yerelleştirme CDN yapılandırması.

`WEBLATE_LOCALIZE_CDN_PATH` kapsayıcı içindeki yoldur. Geçici depolama biriminde değil, kalıcı birimde olmalıdır.

Olasılıklardan biri, Weblate verilerinin içinde tutulmasıdır:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

Not: Weblate tarafından oluşturulan dosyaların sunulmasını ayarlamak sizin sorumluluğunuzdur. Bu seçenek yalnızca dosyaları yapılandırılmış konumda tutar.

Ayrıca bakınız:

weblate-cdn, `LOCALIZE_CDN_URL`, `LOCALIZE_CDN_PATH`

Etkin uygulamaları, denetimleri, eklentileri ya da kendiliğinden düzeltmeleri değiştirmek

3.8-5 sürümünde geldi.

Etkin denetimlerin, eklentilerin ya da kendiliğinden düzeltmelerin iç yapılandırması şu değişkenlerle ayarlanabilir:

WEBLATE_ADD_APPS**WEBLATE_REMOVE_APPS****WEBLATE_ADD_CHECK****WEBLATE_REMOVE_CHECK****WEBLATE_ADD_AUTOFIX****WEBLATE_REMOVE_AUTOFIX****WEBLATE_ADD_ADDONS****WEBLATE_REMOVE_ADDONS**

Örnek:

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

Ayrıca bakınız:

`CHECK_LIST`, `AUTOFIX_LIST`, `WEBLATE_ADDONS`, `INSTALLED_APPS`

Kapsayıcı ayarları

WEBLATE_WORKERS

4.6.1 sürümünde geldi.

Kapsayıcıda çalışan işlemlerin temel sayısı. Ayarlanmadığında, kullanılabilir işlemci çekirdeği sayısına bağlı olarak kapsayıcı başlangıcında kendiliğinden belirlenir.

`CELERY_MAIN_OPTIONS`, `CELERY_NOTIFY_OPTIONS`, `CELERY_MEMORY_OPTIONS`, `CELERY_TRANSLATE_OPTIONS`, `CELERY_BACKUP_OPTIONS`, `CELERY_BEAT_OPTIONS`, ve `WEB_WORKERS` seçeneklerini ayarlamak için kullanılır. Bu seçenekleri ince ayar yapmak için kullanabilirsiniz.

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

Bu seçenekler, Celery işlemi seçeneklerini ayarlamanızı sağlar. Eşzamanlılığı ayarlamak (`--concurrency 16`) veya farklı havuz uygulaması kullanmak (`-pool =gevent`) yararlı olabilir.

Varsayılan olarak, eşzamanlı işlem sayısı `WEBLATE_WORKERS` seçeneğinden alınır.

Örnek:

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

Ayrıca bakınız:

Celery işlemi seçenekleri, *Celery ile arka plan görevlerini kullanmak*

WEB_WORKERS

Çalışacak uWSGI işlemi sayısını yapılandırın.

Varsayılan değer: `WEBLATE_WORKERS`.

Örnek:

```
environment:
  WEB_WORKERS: 32
```

WEBLATE_SERVICE

Kapsayıcı içinde hangi hizmetlerin yürütülmesi gerektiğini tanımlar. Bunu *Yatay ölçeklendirme* için kullanın.

Şu hizmetler tanımlanmıştır:

celery-beat

Celery görev zamanlayıcısı. Yalnızca bir kopya çalışıyor olmalıdır. Bu kapsayıcı aynı zamanda veri tabanı yapısı aktarımlarından da sorumludur ve diğerlerinden önce başlatılmalıdır.

celery-backup

Yedeklemeler için Celery işlemi. Yalnızca bir kopya çalışmalıdır.

celery-celery

Genel Celery işlemi.

celery-memory

Çeviri belleği Celery işlemi.

celery-notify

Bildirim Celery işlemi.

celery-translate

Kendiliğinden çeviri Celery işlemi.

web

Web sunucusu.

Docker kapsayıcısı birimleri

Weblate kapsayıcısı tarafından dışa aktarılan iki birim (veri ve ön bellek) bulunur. Ayrıca diğer hizmet kapsayıcılarının (PostgreSQL ya da Redis) da veri birimleri vardır. Ancak bunlar bu belgenin kapsamında değildir.

Veri birimi, kopyalanmış depolar gibi Weblate kalıcı verilerini depolamak veya Weblate kurulumunu özelleştirmek için kullanılır.

Docker biriminin barındırma sistemindeki konumu Docker yapılandırmanıza bağlıdır. Ancak genellikle `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/` (yol, docker-compose klasörünün adı, kapsayıcı ve birim adlarından oluşur) içinde tutulur. Kapsayıcıda `/app/data` olarak bağlanır.

Ön bellek birimi `/app/cache` olarak bağlanır ve durağan dosyalar ile `CACHE_DIR` klasörünü tutmak için kullanılır. İçeriği kapsayıcı başlangıcında yeniden oluşturulur ve birim `tmpfs` gibi kısa ömürlü dosya sistemi kullanılarak bağlanabilir.

Birimleri el ile oluştururken, klasörlerin sahibi kapsayıcı içindeki UID 1000 olmalıdır.

Ayrıca bakınız:

[Docker birimleri belgeleri](#)

Ortam değişkenlerinin yapılandırması

Docker ortam değişkenleri, Weblate kurulumları ile ilgili pek çok *yapılandırma ayarını* karşılamayı amaçlıyor.

Bir ortam değişkeni olarak karşılığı olmayan bir ayar bulursanız ve var olması gerektiğine inanıyorsanız, *Gelecek Weblate sürümlerine eklenmesini istemekten çekinmeyin*.

Docker ortam değişkeni olarak karşılığı olmayan bir ayarı değiştirmeniz gerekirse, bunu *veri biriminden* ya da *Docker kalıbını genişleterek* yapabilirsiniz.

Ayrıca bakınız:

Weblate platformunu özelleştirmek

Veri biriminden alınan ayarları değiştirmek

Ortam değişkenleri ile tanımlanan ayarları genişletmek ya da değiştirmek için veri birimi kök klasöründe bir `/app/data/settings-override.py` dosyası oluşturabilirsiniz.

Docker kalıbını genişleterek ayarları değiştirmek

Ayarları veri birimi yerine Docker kalıbı düzeyinde değiştirmek için:

1. *Özel bir Python paketi oluşturmak.*
2. Paketinize, `weblate.settings_docker` ögesinden tüm ayarları içe aktaran bir modül ekleyin.

Örneğin, *Bir Python modülü oluşturmak* adresinde tanımlanan örnek paket yapısı içinde, şu başlangıç koduyla bir `weblate_customization/weblate_customization/settings.py` dosyası oluşturabilirsiniz:

```
from weblate.settings_docker import *
```

3. Resmi Weblate Docker kalıbından devralınan ve ardından paketinizi kuran ve ayarlar modülünüze `DJANGO_SETTINGS_MODULE` ortam değişkenini gösteren özel bir Dockerfile oluşturun:

```
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. Resmi Weblate Docker kalıbını kullanmak yerine, bu Dockerfile dosyasından özel bir kalıp oluşturun.

Bunu `docker-compose.override.yml` ile yapmanın *temiz bir yolu yoktur*. Bu dosyadaki `weblate` düğümüne `build: .` ekleyebilirsiniz. Ancak daha sonra özel kalıbınız sisteminizde `"weblate/weblate"` olarak etiketlenir ve bu da soruna yol açabilir.

Bu nedenle, `docker-compose.yml` dosyasını doğrudan resmi depo <https://github.com/WeblateOrg/docker-compose> üzerinden değiştirmeden kullanmak ve `docker-compose.override.yml` ile genişletmek yerine, resmi `docker-compose.yml` dosyasının bir kopyasını oluşturmak ve kopyanızı `image: weblate/weblate` yerine `build: .` ile değiştirmek için düzenlemek isteyebilirsiniz.

`docker-compose` kullanırken kaynaktan kalıp oluşturma ile ilgili ayrıntılı bilgi almak için [Dosya oluşturma referansı](#) bölümüne bakın.

5. Ayarları tanımlamak veya yeniden tanımlamak için özel ayarlar modülünüzü genişletin.

Hangi ayarların öncelikli olduğunu belirlemek için ayarları yukarıdaki içe aktarma ifadesinden önce ya da sonra tanımlayabilirsiniz. İçe aktarma ifadesinden önce tanımlanmış ayarlar, ortam değişkenleri ve veri biriminde tanımlanmış ayar değişiklikleri tarafından değiştirilebilir. İçe aktarma ifadesinden sonra tanımlanan ayar değiştirilmez.

Ayrıca daha ileri gidebilirsiniz. Örneğin, ayarların karşılığını belirlemek gibi `weblate.docker_settings` uygulamasının https://github.com/WeblateOrg/weblate/blob/main/weblate/settings_docker.py yaptığı bazı şeyleri yeniden oluşturabilirsiniz. Ortam değişkenleri olarak veya veri birimindeki Python dosyalarından ayarların değiştirilmesini sağlayabilirsiniz.

Logo ve diğer durağan dosyaları değiştirmek

3.8-5 sürümünde geldi.

Weblate ile birlikte gelen durağan dosyalar `/app/data/python/customize/static` içinden değiştirilebilir (bilgi almak için: [Docker kapsayıcısı birimleri](#)). Örneğin, `/app/data/python/customize/static/favicon.ico` dosyası favicon simgesi için kullanılır.

İpucu: Dosyalar kapsayıcı başlatıldığında ilgili konuma kopyalanır. Bu nedenle birimin içeriğini değiştirdikten sonra Weblate yeniden başlatılmalıdır.

Bu yaklaşım, Weblate kalıplarını değiştirmek için de kullanılabilir. Örneğin [Yasal](#) belgeler `/app/data/python/customize/templates/legal/documents` içine yerleştirilebilir.

Alternatif olarak, kendi modülünüzü de ekleyebilir (bilgi almak için: `../customize`) ve Docker kapsayıcısına ayrı birim olarak ekleyebilirsiniz. Örneğin:

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

PostgreSQL sunucusunu yapılandırmak

PostgreSQL kapsayıcısı varsayılan PostgreSQL yapılandırmasını kullanır ve işlemci çekirdeklerini ya da belleği etkili bir şekilde kullanmaz. Başarımı artırmak için yapılandırmanın özelleştirilmesi önerilir.

Yapılandırma, https://hub.docker.com/_/postgres adresindeki *Database Configuration* bölümünde açıklandığı gibi ayarlanabilir. Ortamınıza uygun bir yapılandırma oluşturmak için <https://pgtune.leopard.in.ua/> kullanabilirsiniz.

Kapsayıcı iç yapısı

Kapsayıcı, bireysel hizmetleri başlatmak için `program: supervisor` kullanır. [Yatay ölçeklendirme](#) kullanılması durumunda, bir kapsayıcıda yalnızca tek bir hizmet başlatılır.

Hizmetlerin durumunu denetlemek için şunu kullanın:

```
docker-compose exec --user weblate weblate supervisorctl status
```

Her Celery kuyruğu için ayrı hizmetler bulunur (bilgi almak için: `ref:celery`). Uygun işlemi durdurarak bazı görevlerin yürütülmesini durdurabilirsiniz:

```
docker-compose exec --user weblate weblate supervisorctl stop celery-translate
```

Debian ve Ubuntu üzerine kurmak

Donanım gereksinimleri

Weblate, herhangi bir çağdaş donanımda sorunsuz çalışmalıdır, Tek bir sunucuda Weblate bileşenlerini (Weblate, veri tabanı ve site sunucusu) çalıştırmak için gerekli en az yapılandırma şu şekildedir:

- 3 GB RAM bellek
- 2 işlemci çekirdeği
- 1 GB depolama alanı

Bellek ne kadar çok olursa o kadar iyidir. Tüm düzeylerde (dosya sistemi, veri tabanı ve Weblate) ön bellek için kullanılır.

Eşzamanlı bulunacak kullanıcı sayısı, gerekli işlemci çekirdeği sayısını artırır. Yüzlerce bileşenden oluşan çeviriler için en az 4 GB RAM bellek bulunması önerilir.

Veri tabanı depolama alanı gereksinimi, barındırılacak 1 milyon sözcük için yaklaşık 300 MB olarak düşünebilir. Kopyalanmış depolar için gereken depolama alanı değişiklik gösterir. Ancak Weblate sığ kopyalama ile depo boyutlarını en az düzeyde tutmaya çalışır.

Not: Weblate kurulumunuz için gerçek gereksinimler, üzerinde bulunacak çevirilerin boyutuna bağlı olarak büyük ölçüde değişiklik gösterebilir.

Kurulum

Sistem gereksinimleri

Python modüllerini oluşturmak için gereken bağımlılıkları kurun (bilgi almak için: [Yazılım gereksinimleri](#)):

```
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
```

Kullanmak istediğiniz özelliklere bağlı olarak gereken isteğe bağlı bağımlılıkları kurun (bilgi almak için: [İsteğe bağlı bağımlılıklar](#)):

```
apt install -y \
  tesseract-ocr libtesseract-dev libleptonica-dev \
  libldap2-dev libldap-common libsasl2-dev \
  libxmlsec1-dev
```

İsteğe bağlı olarak üretim sunucusunu çalıştırmak için gereken yazılımı kurun. Bilgi almak için: sunucu, [Weblate için veri tabanı kurulumu](#), [Celery ile arka plan görevlerini kullanmak](#). Kurulumunuzun boyutuna bağlı olarak, bu bileşenleri adanmış sunucular üzerinde çalıştırmak isteyebilirsiniz.

Yerel kurulum yönergeleri:

```
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
apt install -y redis-server

# Database server: PostgreSQL
apt install -y postgresql postgresql-contrib

# SMTP server
apt install -y exim4
```

Python modülleri

İpucu: Sisteminizden ayrı bir ortama Weblate kurmak için virtualenv kullanıyoruz. Bu konuda bilginiz yoksa, virtualenv [User Guide](#) bölümüne bakabilirsiniz.

1. Weblate için virtualenv oluşturun:

```
virtualenv ~/weblate-env
```

2. Weblate için virtualenv etkinleştirin:

```
. ~/weblate-env/bin/activate
```

3. Tüm isteğe bağlı bağımlılıklar ile birlikte Weblate kurulumunu yapın:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

İsteğe bağlı bağımlılıkları ayarlamak için lütfen [İsteğe bağlı bağımlılıklar](#) seçeneğini denetleyin.

Not: Bazı Linux dağıtımlarında Weblate çalıştırıldığında libffi hatasıyla karşılaşılabilir:

```
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)
```

Bu durum, PyPI ile dağıtılan binary paketlerinin dağıtımla uyumsuz olmasından kaynaklanır. Bu sorunu gidermek için, paketi sisteminizde yeniden oluşturmanız gerekir:

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

Weblate platformunu yapılandırmak

Not: Aşağıdakiler, Weblate için virtualenv kullanımının etkinleştirilmiş olduğunu varsayar (`. ~/weblate-env/bin/activate` komutu yürütülerek). Virtualenv etkin değilse, **weblate** komutunun tam yolunu `~/weblate-env/bin/weblate` olarak belirtin.

1. `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` dosyasını `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py` dosyası olarak kopyalayın.
2. Yeni `settings.py` dosyasındaki değerleri istediğiniz gibi ayarlayın. En azından veri tabanı kimlik doğrulama bilgilerini ve Django gizli anahtarını belirtmeniz gerekir. Ancak üretim kurulumu için daha fazla değişiklik yapmak isteyeceksiniz. Bilgi almak için: [Yapılandırmayı ayarlama](#).

3. Weblate için veri tabanını ve yapısını oluşturun (örnek ayarlar PostgreSQL kullanır, üretime hazır bir kurulum için [Weblate için veri tabanı kurulumu](#) bölümüne bakabilirsiniz):

```
weblate migrate
```

4. Yönetici için bir hesap oluşturun ve parolasını panoya kopyalayıp daha sonra kullanmak üzere kaydedin:

```
weblate createadmin
```

5. Web sunucunuz için durağan dosyaları derleyin (bilgi almak için: [Sunucuyu çalıştırmak](#) ve [Durağan dosyalar sunmak](#)):

```
weblate collectstatic
```

6. JavaScript ve CSS dosyalarını sıkıştırın (isteğe bağlı, bilgi almak için: [İstemci varlıklarının sıkıştırılması](#)):

```
weblate compress
```

7. Celery işlemlerini başlatın. Bu adım, geliştirme ortamları için gerekli değildir. Ancak üretim ortamlarında yapılması önemle önerilir. Bilgi almak için [Celery ile arka plan görevlerini kullanmak](#):

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Geliştirme sunucusunu başlatın ([Sunucuyu çalıştırmak](#) bölümünden üretim kurulumunun ayrıntılarına bakabilirsiniz):

```
weblate runserver
```

Kurulum sonrası

Tebrikler, Weblate sunucunuz artık çalışıyor ve kullanmaya başlayabilirsiniz.

- Weblate arayüzüne <http://localhost:8000/> adresinden erişebilirsiniz.
- Kurulum sırasında belirtilen yönetici kimlik doğrulama bilgileriyle oturum açın ya da yeni kullanıcı hesapları açın.
- Weblate virtualenv etkinleştirildiğinde **weblate** komutunu kullanarak Weblate komutlarını yürütebilirsiniz, Bilgi almak için: [Yönetim komutları](#).
- Deneme sunucusunu Ctrl+C tuşlarıyla durdurabilirsiniz.
- Kurulumunuzla ilgili olası sorunları [/manage/performance/](#) adresinden (bilgi almak için: [Yönetim arayüzü](#)) ya da **weblate check --deploy** komutundan görebilirsiniz. Bilgi almak için: [Üretim kurulumu](#).

Çeviri eklemek

1. Yönetim arayüzünü açın (<http://localhost:8000/create/project/>) ve çevirmek istediğiniz bir proje oluşturun. Bilgi almak için [Proje yapılandırması](#).

Burada yalnızca proje adını ve sitesini belirtmeniz gerekir.

2. Çeviri için gerçek nesne olan bir bileşen oluşturun. Sürüm denetimi sistemi deposunu yönetilir ve hangi dosyaların çevrileceğini seçer. Bilgi almak için [Bileşen yapılandırması](#).

Buradaki önemli alanlar şunlardır: [Bileşen adı](#), [Kaynak kod deposu](#) ve çevrilebilir dosyaları bulmak için [Dosya maskesi](#). Weblate, [GNU gettext](#), [Android dizge kaynakları](#), [Apple iOS dizgeleri](#), [Java özellikleri](#), [Stringsdict biçimi](#) veya [Fluent biçimi](#) ile birlikte çok çeşitli biçimleri destekler. Bilgi almak için: [Desteklenen dosya biçimleri](#).

3. Yukarıdakiler tamamlandıktan sonra (sürüm denetimi sistemi deponuzun boyutuna ve çevrilecek ileti sayısına bağlı olarak uzun bir süreç olabilir), çeviri yapmaya başlayabilirsiniz.

SUSE ve openSUSE üzerine kurmak

Donanım gereksinimleri

Weblate, herhangi bir çağdaş donanımda sorunsuz çalışmalıdır, Tek bir sunucuda Weblate bileşenlerini (Weblate, veri tabanı ve site sunucusu) çalıştırmak için gerekli en az yapılandırma şu şekildedir:

- 3 GB RAM bellek
- 2 işlemci çekirdeği
- 1 GB depolama alanı

Bellek ne kadar çok olursa o kadar iyidir. Tüm düzeylerde (dosya sistemi, veri tabanı ve Weblate) ön bellek için kullanılır.

Eşzamanlı bulunacak kullanıcı sayısı, gerekli işlemci çekirdeği sayısını artırır. Yüzlerce bileşenden oluşan çeviriler için en az 4 GB RAM bellek bulunması önerilir.

Veri tabanı depolama alanı gereksinimi, barındırılacak 1 milyon sözcük için yaklaşık 300 MB olarak düşünebilir. Kopyalanmış depolar için gereken depolama alanı değişiklik gösterir. Ancak Weblate sığ kopyalama ile depo boyutlarını en az düzeyde tutmaya çalışır.

Not: Weblate kurulumunuz için gerçek gereksinimler, üzerinde bulunacak çevirilerin boyutuna bağlı olarak büyük ölçüde değişiklik gösterebilir.

Kurulum

Sistem gereksinimleri

Python modüllerini oluşturmak için gereken bağımlılıkları kurun (bilgi almak için: [Yazılım gereksinimleri](#)):

```
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
```

Kullanmak istediğiniz özelliklere bağlı olarak gereken isteğe bağlı bağımlılıkları kurun (bilgi almak için: [İsteğe bağlı bağımlılıklar](#)):

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

İsteğe bağlı olarak üretim sunucusunu çalıştırmak için gereken yazılımı kurun. Bilgi almak için: sunucu, [Weblate için veri tabanı kurulumu](#), [Celery ile arka plan görevlerini kullanmak](#). Kurulumunuzun boyutuna bağlı olarak, bu bileşenleri adanmış sunucular üzerinde çalıştırmak isteyebilirsiniz.

Yerel kurulum yönergeleri:

```
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# Database server: PostgreSQL
zypper install postgresql postgresql-contrib

# SMTP server
zypper install postfix
```

Python modülleri

İpucu: Sisteminizden ayrı bir ortama Weblate kurmak için virtualenv kullanıyoruz. Bu konuda bilginiz yoksa, virtualenv [User Guide](#) bölümüne bakabilirsiniz.

1. Weblate için virtualenv oluşturun:

```
virtualenv ~/weblate-env
```

2. Weblate için virtualenv etkinleştirin:

```
. ~/weblate-env/bin/activate
```

3. Tüm isteğe bağlı bağımlılıklar ile birlikte Weblate kurulumunu yapın:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

İsteğe bağlı bağımlılıkları ayarlamak için lütfen [İsteğe bağlı bağımlılıklar](#) seçeneğini denetleyin.

Not: Bazı Linux dağıtımlarında Weblate çalıştırıldığında libffi hatasıyla karşılaşılabilir:

```
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)
```

Bu durum, PyPI ile dağıtılan binary paketlerinin dağıtımla uyumsuz olmasından kaynaklanır. Bu sorunu gidermek için, paketi sisteminizde yeniden oluşturmanız gerekir:

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

Weblate platformunu yapılandırmak

Not: Aşağıdakiler, Weblate için virtualenv kullanımının etkinleştirilmiş olduğunu varsayar (`. ~/weblate-env/bin/activate` komutu yürütülerek). Virtualenv etkin değilse, **weblate** komutunun tam yolunu `~/weblate-env/bin/weblate` olarak belirtin.

1. `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` dosyasını `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py` dosyası olarak kopyalayın.
2. Yeni `settings.py` dosyasındaki değerleri istediğiniz gibi ayarlayın. En azından veri tabanı kimlik doğrulama bilgilerini ve Django gizli anahtarını belirtmeniz gerekir. Ancak üretim kurulumu için daha fazla değişiklik yapmak isteyeceksiniz. Bilgi almak için: [Yapılandırmayı ayarlama](#).

3. Weblate için veri tabanını ve yapısını oluşturun (örnek ayarlar PostgreSQL kullanır, üretime hazır bir kurulum için [Weblate için veri tabanı kurulumu](#) bölümüne bakabilirsiniz):

```
weblate migrate
```

4. Yönetici için bir hesap oluşturun ve parolasını panoya kopyalayıp daha sonra kullanmak üzere kaydedin:

```
weblate createadmin
```

5. Web sunucunuz için durağan dosyaları derleyin (bilgi almak için: [Sunucuyu çalıştırmak](#) ve [Durağan dosyalar sunmak](#)):

```
weblate collectstatic
```

6. JavaScript ve CSS dosyalarını sıkıştırın (isteğe bağlı, bilgi almak için: [İstemci varlıklarının sıkıştırılması](#)):

```
weblate compress
```

7. Celery işlemlerini başlatın. Bu adım, geliştirme ortamları için gerekli değildir. Ancak üretim ortamlarında yapılması önemle önerilir. Bilgi almak için [Celery ile arka plan görevlerini kullanmak](#):

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Geliştirme sunucusunu başlatın ([Sunucuyu çalıştırmak](#) bölümünden üretim kurulumunun ayrıntılarına bakabilirsiniz):

```
weblate runserver
```

Kurulum sonrası

Tebrikler, Weblate sunucunuz artık çalışıyor ve kullanmaya başlayabilirsiniz.

- Weblate arayüzüne <http://localhost:8000/> adresinden erişebilirsiniz.
- Kurulum sırasında belirtilen yönetici kimlik doğrulama bilgileriyle oturum açın ya da yeni kullanıcı hesapları açın.
- Weblate virtualenv etkinleştirildiğinde **weblate** komutunu kullanarak Weblate komutlarını yürütebilirsiniz, Bilgi almak için: [Yönetim komutları](#).
- Deneme sunucusunu Ctrl+C tuşlarıyla durdurabilirsiniz.
- Kurulumunuzla ilgili olası sorunları [/manage/performance/](#) adresinden (bilgi almak için: [Yönetim arayüzü](#)) ya da **weblate check --deploy** komutundan görebilirsiniz. Bilgi almak için: [Üretim kurulumu](#).

Çeviri eklemek

1. Yönetim arayüzünü açın (<http://localhost:8000/create/project/>) ve çevirmek istediğiniz bir proje oluşturun. Bilgi almak için [Proje yapılandırması](#).

Burada yalnızca proje adını ve sitesini belirtmeniz gerekir.

2. Çeviri için gerçek nesne olan bir bileşen oluşturun. Sürüm denetimi sistemi deposunu yönetilir ve hangi dosyaların çevrileceğini seçer. Bilgi almak için [Bileşen yapılandırması](#).

Buradaki önemli alanlar şunlardır: [Bileşen adı](#), [Kaynak kod deposu](#) ve çevrilebilir dosyaları bulmak için [Dosya maskesi](#). Weblate, [GNU gettext](#), [Android dizge kaynakları](#), [Apple iOS dizgeleri](#), [Java özellikleri](#), [Stringsdict biçimi](#) veya [Fluent biçimi](#) ile birlikte çok çeşitli biçimleri destekler. Bilgi almak için: [Desteklenen dosya biçimleri](#).

3. Yukarıdakiler tamamlandıktan sonra (sürüm denetimi sistemi deponuzun boyutuna ve çevrilecek ileti sayısına bağlı olarak uzun bir süreç olabilir), çeviri yapmaya başlayabilirsiniz.

RedHat, Fedora ve CentOS üzerine kurmak

Donanım gereksinimleri

Weblate, herhangi bir çağdaş donanımda sorunsuz çalışmalıdır, Tek bir sunucuda Weblate bileşenlerini (Weblate, veri tabanı ve site sunucusu) çalıştırmak için gerekli en az yapılandırma şu şekildedir:

- 3 GB RAM bellek
- 2 işlemci çekirdeği
- 1 GB depolama alanı

Bellek ne kadar çok olursa o kadar iyidir. Tüm düzeylerde (dosya sistemi, veri tabanı ve Weblate) ön bellek için kullanılır.

Eşzamanlı bulunacak kullanıcı sayısı, gerekli işlemci çekirdeği sayısını artırır. Yüzlerce bileşenden oluşan çeviriler için en az 4 GB RAM bellek bulunması önerilir.

Veri tabanı depolama alanı gereksinimi, barındırılacak 1 milyon sözcük için yaklaşık 300 MB olarak düşünebilir. Kopyalanmış depolar için gereken depolama alanı değişiklik gösterir. Ancak Weblate sıg kopyalama ile depo boyutlarını en az düzeyde tutmaya çalışır.

Not: Weblate kurulumunuz için gerçek gereksinimler, üzerinde bulunacak çevirilerin boyutuna bağlı olarak büyük ölçüde değişiklik gösterebilir.

Kurulum

Sistem gereksinimleri

Python modüllerini oluşturmak için gereken bağımlılıkları kurun (bilgi almak için: [Yazılım gereksinimleri](#)):

```
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
```

Kullanmak istediğiniz özelliklere bağlı olarak gereken isteğe bağlı bağımlılıkları kurun (bilgi almak için: [İsteğe bağlı bağımlılıklar](#)):

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

İsteğe bağlı olarak üretim sunucusunu çalıştırmak için gereken yazılımı kurun. Bilgi almak için: sunucu, [Weblate için veri tabanı kurulumu](#), [Celery ile arka plan görevlerini kullanmak](#). Kurulumunuzun boyutuna bağlı olarak, bu bileşenleri adanmış sunucular üzerinde çalıştırmak isteyebilirsiniz.

Yerel kurulum yöntemleri:

```
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# Database server: PostgreSQL
dnf install postgresql postgresql-contrib

# SMTP server
dnf install postfix
```

Python modülleri

İpucu: Sisteminizden ayrı bir ortama Weblate kurmak için virtualenv kullanıyoruz. Bu konuda bilginiz yoksa, virtualenv [User Guide](#) bölümüne bakabilirsiniz.

1. Weblate için virtualenv oluşturun:

```
virtualenv ~/weblate-env
```

2. Weblate için virtualenv etkinleştirin:

```
. ~/weblate-env/bin/activate
```

3. Tüm isteğe bağlı bağımlılıklar ile birlikte Weblate kurulumunu yapın:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

İsteğe bağlı bağımlılıkları ayarlamak için lütfen [İsteğe bağlı bağımlılıklar](#) seçeneğini denetleyin.

Not: Bazı Linux dağıtımlarında Weblate çalıştırıldığında libffi hatasıyla karşılaşılabilir:

```
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)
```

Bu durum, PyPI ile dağıtılan binary paketlerinin dağıtımla uyumsuz olmasından kaynaklanır. Bu sorunu gidermek için, paketi sisteminizde yeniden oluşturmanız gerekir:

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

Weblate platformunu yapılandırmak

Not: Aşağıdakiler, Weblate için virtualenv kullanımının etkinleştirilmiş olduğunu varsayar (`. ~/weblate-env/bin/activate` komutu yürütülerek). Virtualenv etkin değilse, **weblate** komutunun tam yolunu `~/weblate-env/bin/weblate` olarak belirtin.

1. `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` dosyasını `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py` dosyası olarak kopyalayın.
2. Yeni `settings.py` dosyasındaki değerleri istediğiniz gibi ayarlayın. En azından veri tabanı kimlik doğrulama bilgilerini ve Django gizli anahtarını belirtmeniz gerekir. Ancak üretim kurulumu için daha fazla değişiklik yapmak isteyeceksiniz. Bilgi almak için: [Yapılandırmayı ayarlama](#).

3. Weblate için veri tabanını ve yapısını oluşturun (örnek ayarlar PostgreSQL kullanır, üretime hazır bir kurulum için [Weblate için veri tabanı kurulumu](#) bölümüne bakabilirsiniz):

```
weblate migrate
```

4. Yönetici için bir hesap oluşturun ve parolasını panoya kopyalayıp daha sonra kullanmak üzere kaydedin:

```
weblate createadmin
```

5. Web sunucunuz için durağan dosyaları derleyin (bilgi almak için: [Sunucuyu çalıştırmak](#) ve [Durağan dosyalar sunmak](#)):

```
weblate collectstatic
```

6. JavaScript ve CSS dosyalarını sıkıştırın (isteğe bağlı, bilgi almak için: [İstemci varlıklarının sıkıştırılması](#)):

```
weblate compress
```

7. Celery işlemlerini başlatın. Bu adım, geliştirme ortamları için gerekli değildir. Ancak üretim ortamlarında yapılması önemle önerilir. Bilgi almak için [Celery ile arka plan görevlerini kullanmak](#):

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Geliştirme sunucusunu başlatın ([Sunucuyu çalıştırmak](#) bölümünden üretim kurulumunun ayrıntılarına bakabilirsiniz):

```
weblate runserver
```

Kurulum sonrası

Tebrikler, Weblate sunucunuz artık çalışıyor ve kullanmaya başlayabilirsiniz.

- Weblate arayüzüne <http://localhost:8000/> adresinden erişebilirsiniz.
- Kurulum sırasında belirtilen yönetici kimlik doğrulama bilgileriyle oturum açın ya da yeni kullanıcı hesapları açın.
- Weblate virtualenv etkinleştirildiğinde **weblate** komutunu kullanarak Weblate komutlarını yürütebilirsiniz, Bilgi almak için: [Yönetim komutları](#).
- Deneme sunucusunu Ctrl+C tuşlarıyla durdurabilirsiniz.
- Kurulumunuzla ilgili olası sorunları [/manage/performance/](#) adresinden (bilgi almak için: [Yönetim arayüzü](#)) ya da **weblate check --deploy** komutundan görebilirsiniz. Bilgi almak için: [Üretim kurulumu](#).

Çeviri eklemek

1. Yönetim arayüzünü açın (<http://localhost:8000/create/project/>) ve çevirmek istediğiniz bir proje oluşturun. Bilgi almak için [Proje yapılandırması](#).

Burada yalnızca proje adını ve sitesini belirtmeniz gerekir.

2. Çeviri için gerçek nesne olan bir bileşen oluşturun. Sürüm denetimi sistemi deposunu yönlendir ve hangi dosyaların çevrileceğini seçer. Bilgi almak için [Bileşen yapılandırması](#).

Buradaki önemli alanlar şunlardır: [Bileşen adı](#), [Kaynak kod deposu](#) ve çevrilebilir dosyaları bulmak için [Dosya maskesi](#). Weblate, [GNU gettext](#), [Android dizge kaynakları](#), [Apple iOS dizgeleri](#), [Java özellikleri](#), [Stringsdict biçimi](#) veya [Fluent biçimi](#) ile birlikte çok çeşitli biçimleri destekler. Bilgi almak için: [Desteklenen dosya biçimleri](#).

3. Yukarıdakiler tamamlandıktan sonra (sürüm denetimi sistemi deponuzun boyutuna ve çevrilecek ileti sayısına bağlı olarak uzun bir süreç olabilir), çeviri yapmaya başlayabilirsiniz.

macOS üzerine kurulum

Donanım gereksinimleri

Weblate, herhangi bir çağdaş donanımda sorunsuz çalışmalıdır, Tek bir sunucuda Weblate bileşenlerini (Weblate, veri tabanı ve site sunucusu) çalıştırmak için gerekli en az yapılandırma şu şekildedir:

- 3 GB RAM bellek
- 2 işlemci çekirdeği
- 1 GB depolama alanı

Bellek ne kadar çok olursa o kadar iyidir. Tüm düzeylerde (dosya sistemi, veri tabanı ve Weblate) ön bellek için kullanılır.

Eşzamanlı bulunacak kullanıcı sayısı, gerekli işlemci çekirdeği sayısını artırır. Yüzlerce bileşenden oluşan çeviriler için en az 4 GB RAM bellek bulunması önerilir.

Veri tabanı depolama alanı gereksinimi, barındırılacak 1 milyon sözcük için yaklaşık 300 MB olarak düşünebilir. Kopyalanmış depolar için gereken depolama alanı değişiklik gösterir. Ancak Weblate sık kopyalama ile depo boyutlarını en az düzeyde tutmaya çalışır.

Not: Weblate kurulumunuz için gerçek gereksinimler, üzerinde bulunacak çevirilerin boyutuna bağlı olarak büyük ölçüde değişiklik gösterebilir.

Kurulum

Sistem gereksinimleri

Python modüllerini oluşturmak için gereken bağımlılıkları kurun (bilgi almak için: [Yazılım gereksinimleri](#)):

```
brew install python pango cairo gobject-introspection libffi glib libyaml
pip install virtualenv
```

Homebrew tarafından sağlanan libffi ve openssl sürümlerinin pip tarafından bulunabileceğinden emin olun. Kurulum oluşturma adımı sırasında buna gerek duyulur.

```
export PKG_CONFIG_PATH="/usr/local/opt/libffi/lib/pkgconfig:/usr/local/opt/
↳ openssl@3/lib/pkgconfig"
```

Kullanmak istediğiniz özelliklere bağlı olarak gereken isteğe bağlı bağımlılıkları kurun (bilgi almak için: [İsteğe bağlı bağımlılıklar](#)):

```
brew install tesseract
```

İsteğe bağlı olarak üretim sunucusunu çalıştırmak için gereken yazılımı kurun. Bilgi almak için: sunucu, [Weblate için veri tabanı kurulumu](#), [Celery ile arka plan görevlerini kullanmak](#). Kurulumunuzun boyutuna bağlı olarak, bu bileşenleri adanmış sunucular üzerinde çalıştırmak isteyebilirsiniz.

Yerel kurulum yönergeleri:

```
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
brew install redis

# Database server: PostgreSQL
brew install postgresql
```

Python modülleri

İpucu: Sisteminizden ayrı bir ortama Weblate kurmak için virtualenv kullanıyoruz. Bu konuda bilginiz yoksa, virtualenv [User Guide](#) bölümüne bakabilirsiniz.

1. Weblate için virtualenv oluşturun:

```
virtualenv ~/weblate-env
```

2. Weblate için virtualenv etkinleştirin:

```
. ~/weblate-env/bin/activate
```

3. Tüm isteğe bağlı bağımlılıklar ile birlikte Weblate kurulumunu yapın:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

İsteğe bağlı bağımlılıkları ayarlamak için lütfen *İsteğe bağlı bağımlılıklar* seçeneğini denetleyin.

Not: Bazı Linux dağıtımlarında Weblate çalıştırıldığında libffi hatasıyla karşılaşılabilir:

```
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)
```

Bu durum, PyPI ile dağıtılan binary paketlerinin dağıtımla uyumsuz olmasından kaynaklanır. Bu sorunu gidermek için, paketi sisteminizde yeniden oluşturmanız gerekir:

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

Weblate platformunu yapılandırmak

Not: Aşağıdakiler, Weblate için virtualenv kullanımının etkinleştirilmiş olduğunu varsayar (. ~/weblate-env/bin/activate komutu yürütülerek). Virtualenv etkin değilse, **weblate** komutunun tam yolunu ~/weblate-env/bin/weblate olarak belirtin.

1. ~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py dosyasını ~/weblate-env/lib/python3.9/site-packages/weblate/settings.py dosyası olarak kopyalayın.
2. Yeni settings.py dosyasındaki değerleri istediğiniz gibi ayarlayın. En azından veri tabanı kimlik doğrulama bilgilerini ve Django gizli anahtarını belirtmeniz gerekir. Ancak üretim kurulumu için daha fazla değişiklik yapmak isteyeceksiniz. Bilgi almak için: *Yapılandırmayı ayarlama*.
3. Weblate için veri tabanını ve yapısını oluşturun (örnek ayarlar PostgreSQL kullanır, üretime hazır bir kurulum için *Weblate için veri tabanı kurulumu* bölümüne bakabilirsiniz):


```
weblate migrate
```

4. Yönetici için bir hesap oluşturun ve parolasını panoya kopyalayıp daha sonra kullanmak üzere kaydedin:

```
weblate createadmin
```

5. Web sunucunuz için durağan dosyaları derleyin (bilgi almak için: [Sunucuyu çalıştırmak](#) ve [Durağan dosyalar sunmak](#)):

```
weblate collectstatic
```

6. JavaScript ve CSS dosyalarını sıkıştırın (isteğe bağlı, bilgi almak için: [İstemci varlıklarının sıkıştırılması](#)):

```
weblate compress
```

7. Celery işlemlerini başlatın. Bu adım, geliştirme ortamları için gerekli değildir. Ancak üretim ortamlarında yapılması önemle önerilir. Bilgi almak için [Celery ile arka plan görevlerini kullanmak](#):

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Geliştirme sunucusunu başlatın ([Sunucuyu çalıştırmak](#) bölümünden üretim kurulumunun ayrıntılarına bakabilirsiniz):

```
weblate runserver
```

Kurulum sonrası

Tebrikler, Weblate sunucunuz artık çalışıyor ve kullanmaya başlayabilirsiniz.

- Weblate arayüzüne <http://localhost:8000/> adresinden erişebilirsiniz.
- Kurulum sırasında belirtilen yönetici kimlik doğrulama bilgileriyle oturum açın ya da yeni kullanıcı hesapları açın.
- Weblate virtualenv etkinleştirildiğinde **weblate** komutunu kullanarak Weblate komutlarını yürütebilirsiniz, Bilgi almak için: [Yönetim komutları](#).
- Deneme sunucusunu Ctrl+C tuşlarıyla durdurabilirsiniz.
- Kurulumunuzla ilgili olası sorunları [/manage/performance/](#) adresinden (bilgi almak için: [Yönetim arayüzü](#)) ya da **weblate check --deploy** komutundan görebilirsiniz. Bilgi almak için: [Üretim kurulumu](#).

Çeviri eklemek

1. Yönetim arayüzünü açın (<http://localhost:8000/create/project/>) ve çevirmek istediğiniz bir proje oluşturun. Bilgi almak için [Proje yapılandırması](#).

Burada yalnızca proje adını ve sitesini belirtmeniz gerekir.

2. Çeviri için gerçek nesne olan bir bileşen oluşturun. Sürüm denetimi sistemi deposunu yönetilir ve hangi dosyaların çevrileceğini seçer. Bilgi almak için [Bileşen yapılandırması](#).

Buradaki önemli alanlar şunlardır: [Bileşen adı](#), [Kaynak kod deposu](#) ve çevrilebilir dosyaları bulmak için [Dosya maskesi](#). Weblate, [GNU gettext](#), [Android dizge kaynakları](#), [Apple iOS dizgeleri](#), [Java özellikleri](#), [Stringsdict biçimi](#) veya [Fluent biçimi](#) ile birlikte çok çeşitli biçimleri destekler. Bilgi almak için: [Desteklenen dosya biçimleri](#).

3. Yukarıdakiler tamamlandıktan sonra (sürüm denetimi sistemi deponuzun boyutuna ve çevrilecek ileti sayısına bağlı olarak uzun bir süreç olabilir), çeviri yapmaya başlayabilirsiniz.

Kaynaklardan kurulum

1. Lütfen Weblate kurulumu için öncelikle sisteminize uygun kurulum yönergelerini izleyin:

- *Debian ve Ubuntu üzerine kurmak*
- *SUSE ve openSUSE üzerine kurmak*
- *RedHat, Fedora ve CentOS üzerine kurmak*

2. Git kullanarak güncel Weblate kaynaklarını edinin (ya da bir tarball indirip paketi ayıklayın):

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

Alternatif olarak, yayınlanmış arşiv dosyalarını da kullanabilirsiniz. Bunları sitemizden <<https://weblate.org/>> indirebilirsiniz. Bu indirmeler şifrelenerek imzalanmıştır. Bilgi almak için: ref: *verify*'.

3. Geçerli Weblate kodunu virtualenv üzerine kurun:

```
. ~/weblate-env/bin/activate  
pip install -e weblate-src
```

4. `weblate/settings_example.py` dosyasını `weblate/settings.py` olarak kopyalayın.
5. Yeni `settings.py` dosyasındaki değerleri istediğiniz gibi ayarlayın. En azından veri tabanı kimlik doğrulama bilgilerini ve Django gizli anahtarını belirtmeniz gerekir. Ancak üretim kurulumu için daha fazla değişiklik yapmak isteyeceksiniz. Bilgi almak için: *Yapılandırmayı ayarlama*.
6. Weblate tarafından kullanılacak veri tabanını oluşturun. Bilgi almak için: *Weblate için veri tabanı kurulumu*.
7. Django tablolarını, durağan dosyaları ve başlangıç verilerini oluşturun (bilgi almak için: *Veri tabanını doldurmak* ve *Durağan dosyalar sunmak*):

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

Not: Bu adım, her depo güncellemesi yaptığınızda yinelenmelidir.

OpenShift üzerine kurulum

OpenShift Weblate kalıbıyla kişisel Weblate bulut sunucunuzu saniyeler içinde çalışır duruma getirebilirsiniz. Tüm Weblate bağımlılıkları önceden katılmıştır. PostgreSQL varsayılan veri tabanı olarak ayarlanır ve kalıcı birim kullanılır.

Kalıbı <<https://github.com/WeblateOrg/openshift/>> adresinde bulabilirsiniz.

Kurulum

Aşağıdaki örneklerde, `oc` istemci aracının yüklü olduğu çalışan bir OpenShift v3.x ortamınız olduğu varsayılmıştır. Lütfen yönergeler için OpenShift belgelerine bakın.

`template.yml` dosyası OpenShift üzerinde tüm bileşenleri çalıştırmak için uygundur. Ayrıca `template-external-postgresql.yml` dosyası bir PostgreSQL sunucusunu başlatmaz ve bir dış PostgreSQL sunucusunu yapılandırmanızı sağlar.

Site konsolu

Ham içeriği `template.yml` dosyasından kopyalayın ve projenize aktarın. Ardından uygulamanızı oluşturmak için OpenShift internet konsolundaki `Create` düğmesini kullanın. Site konsolu, sizden kalıp tarafından kullanılan tüm parametrelerin değerlerini ister.

CLI

Weblate kalıbını geçerli projenizin kalıp kitaplığına yüklemek için şu komutta `template.yml` dosyasını belirtin:

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

Kalıp artık site konsolu ya da komut satırı (CLI) kullanılarak seçilebilir.

Parametreler

Değiştirebileceğiniz parametreler kalıbın parametreler bölümünde listelenir. Şu komutu kullanarak ve kullanılacak dosyayı belirterek bunları komut satırı (CLI) ile listeleyebilirsiniz:

```
$ 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
```

Hazırlamak

Komut satırını (CLI) kalıpları işlemek ve hazırlanan yapılandırmadan nesneleri hemen oluşturmak için de kullanabilirsiniz.

```
$ 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
```

Weblate kopyası, başarılı bir aktarım ve dağıtımdan sonra belirtilmiş `WEBLATE_SITE_DOMAIN` parametresinde kullanılabilir olmalıdır.

Kapsayıcı kurulumundan sonra, `WEBLATE_ADMIN_PASSWORD` seçeneğinde belirtilen parolayla ya da ayarlanmamışsa ilk başlatılmada oluşturulan rastgele bir parolayla `admin` kullanıcısı olarak oturum açabilirsiniz.

‘Admin’ parolasını sıfırlamak için, ilgili `Secret` bölümünde `WEBLATE_ADMIN_PASSWORD` seçeneğine yeni parola ayarladıktan sonra kapsayıcıyı yeniden başlatın.

Elemek

```
$ 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
```

Yapılandırma

Kalıbı işleyerek, Weblate kalıbını özelleştirmek için kullanılacak ilgili bir ConfigMap oluşturulur. ConfigMap doğrudan ortam değişkenleri olarak bağlanır ve her değiştirildiğinde yeni bir dağıtımı tetikler. Daha fazla yapılandırma seçeneği sunan ortam değişkenlerinin tam listesi için *Docker ortam değişkenleri* bölümüne bakabilirsiniz.

Kubernetes üzerine kurulum

Not: Bu rehberde, kurulumu daha ayrıntılı olarak ele alabilmek için Kubernetes konusunda deneyimli gönüllüler arıyoruz.

Kubernetes Helm çizelgesi ile kişisel Weblate bulut sunucunuzu saniyeler içinde çalışır duruma getirebilirsiniz. Tüm Weblate bağımlılıkları önceden katılmıştır. Varsayılan veri tabanı olarak PostgreSQL ayarlanır ve kalıcı birim kullanılır.

Çizelgeyi <<https://github.com/WeblateOrg/helm/>> adresinde bulabilir ve <<https://artifacthub.io/packages/helm/weblate/weblate>> adresinde görebilirsiniz.

Kurulum

```
helm repo add weblate https://helm.weblate.org
helm install my-release weblate/weblate
```

Yapılandırma

Daha fazla yapılandırma seçeneği sunan ortam değişkenlerinin tam listesi için *Docker ortam değişkenleri* bölümüne bakabilirsiniz.

Kurulumunuza ve deneyiminize bağlı olarak, size en uygun kurulum yöntemi seçin:

- Üretim kurulumları için önerilen *Docker ile kurmak*.
- Üretim kurulumları için önerilen virtualenv kurulumu:
 - *Debian ve Ubuntu üzerine kurmak*
 - *SUSE ve openSUSE üzerine kurmak*
 - *RedHat, Fedora ve CentOS üzerine kurmak*
 - *macOS üzerine kurulum*

- *Kaynaklardan kurulum*, geliştirme çalışmaları için önerilir.
- *OpenShift üzerine kurulum*
- *Kubernetes üzerine kurulum*

2.1.2 Yazılım gereksinimleri

İşletim sistemi

Weblate platformunun, Linux, FreeBSD ve macOS üzerinde çalıştığı biliniyor. Diğer Unix benzeri sistemlerde de büyük olasılıkla çalışacaktır.

Weblate, Windows üzerinde desteklenmez. Ancak yine de çalışabilir ve yazılım yamaları mutlulukla kabul edilir.

Diğer hizmetler

Weblate, çalışabilmek için başka hizmetleri kullanır. En azından şu hizmetlerin çalışması gerekir:

- PostgreSQL veri tabanı sunucusu. Bilgi almak için: *Weblate için veri tabanı kurulumu*.
- Ön bellek ve görev kuyruğu için Redis sunucusu. Bilgi almak için: *Celery ile arka plan görevlerini kullanmak*.
- Giden e-postalar için kullanılacak SMTP sunucusu. Bilgi almak için: *Giden e-postayı yapılandırmak*.

Python bağımlılıkları

Weblate, Python ile yazılmıştır ve Python 3.6 ya da üzerindeki sürümleri destekler. Bağımlılıkları pip kullanarak ya da dağıtım paketlerinizden kurabilirsiniz. Tam listeyi `requirements.txt` adresinde bulabilirsiniz.

En önemli bağımlılıklar:

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 çatısı

<https://www.django-rest-framework.org/>

İsteğe bağlı bağımlılıklar

Bazı Weblate özellikleri için şu modüller gereklidir. Hepsini `requirements-optional.txt` içinde bulabilirsiniz.

Mercurial (*Mercurial* depo desteği için isteğe bağlı)

<https://www.mercurial-scm.org/>

phply (*PHP dizgeleri* için isteğe bağlı)

<https://github.com/viraptor/phply>

tesseract (*Dizgeler için görsel bağlam ile karakter tanıma için isteğe bağlı*)

<https://github.com/sirfz/tesseract>

python-akismet (*İstenmeyen gönderim koruması için isteğe bağlı*)

<https://github.com/Nekmo/python-akismet>

ruamel.yaml (*YAML dosyaları için isteğe bağlı*)

<https://pypi.org/project/ruamel.yaml/>

Zeep (*Microsoft Terminology için isteğe bağlı*)

<https://docs.python-zeep.org/>

aeidon (*Alt yazı dosyaları için isteğe bağlı*)

<https://pypi.org/project/aeidon/>

fluent.syntax (*Fluent biçimi için isteğe bağlı*)

<https://projectfluent.org/>

İpucu: Pip kullanarak kurulum yapılırken istediğiniz özellikleri doğrudan belirtebilirsiniz:

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

Ya da tüm isteğe bağlı özelliklerle Weblate kurulumu yapabilirsiniz:

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

Ya da hiç bir isteğe bağlı özellik olmadan Weblate kurulumu yapabilirsiniz:

```
pip install Weblate
```

Veri tabanı arka plan bağımlılıkları

Weblate PostgreSQL, MySQL ve MariaDB veri tabanlarını destekler. Bilgi almak için *Weblate için veri tabanı kurulumu* ve yönetim bölümü belgelerine bakabilirsiniz.

Diğer sistem gereksinimleri

Sisteme şu bağımlılıkların kurulması gerekir:

Git

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

Pango, Cairo ve ilişkili üst bilgi dosyaları ile GObject iç gözlem verileri

<https://cairographics.org/>, <https://pango.gnome.org/>, bilgi almak için: *Pango ve Cairo*

git-review (Gerrit desteği için isteğe bağlı)

<https://pypi.org/project/git-review/>

git-svn (Subversion desteği için isteğe bağlı)

<https://git-scm.com/docs/git-svn>

tesseract ve verileri (ekran görüntülerinde karakter tanıma için isteğe bağlı)

<https://github.com/tesseract-ocr/tesseract>

licensee (bileşen oluştururken lisansın algılanması için isteğe bağlı)

<https://github.com/licensee/licensee>

Yapım zamanı bağımlılıkları

Bazı *Python bağımlılıkları* bağımlılıklarını kurmanız gerekebilir. Bu durum, bunları nasıl kurduğunuza bağlıdır. O yüzden her paketin kendi belgesine başvurun. `Pip` kullanarak yapılan kurulum sırasında ya da dağıtım paketlerini kullanırken önceden oluşturulmuş `Wheels` kullanıyorsanız, bunlara gerek duymazsınız.

Pango ve Cairo

3.7 sürümünde değişti.

Weblate, bitmap bileşenlerini (bilgi almak için: [promotion](#)) ve görüntüleme denetimlerini (bilgi almak için: [Yazı tiplerini yönetmek](#)) oluşturmak için Pango ve Cairo kullanır. Python bağlantılarını düzgün olarak kurmak için önce sistem kitaplıklarını kurmanız gerekir. Hem Cairo hem de Pango gereklidir, bunun için de GLib gereklidir. Tüm bunlar geliştirme dosyaları ve GObject içi gözlem verileriyle birlikte kurulmalıdır.

2.1.3 Sürüm imzalarını doğrulamak

Weblate sürümü, yayın geliştiricisi tarafından şifrelenmiş olarak imzalanır. Şu anda bu işi Michal Čihař yapıyor ve onun PGP anahtarının parmak izi:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

ve <https://keybase.io/nijel> adresinden diğer belirleyici bilgileri alabilirsiniz.

İmzanın indirdiğiniz arşivle eşleştiğini doğrulamanız gerekir. Bu şekilde, yayınlanan kodu değiştirilmemiş olduğundan emin olabilirsiniz. Güncel sürümü indirdiğinizden emin olmak için imzanın tarihini de doğrulamalısınız.

Her arşiv dosyası için, PGP imzasının bulunduğu `.asc` dosyaları bulunur. Bu iki dosyayı aynı klasöre kaydettikten sonra imzayı doğrulayabilirsiniz:

```
$ 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:
gpg: using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Can't check signature: public key not found
```

Görebileceğiniz gibi GPG, herkese açık anahtarı bilmediğini bildiriyor. Bu aşamada şu adımlardan birini uygulamanız gerekir:

- Anahtarı indirmek için `wkd` kullanın:

```
$ 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]
```

- Michal sunucusundan anahtarlığı indirip şununla içe aktarın:

```
$ gpg --import wxth3chu9jfxdxywj1skpmhsj311mzm
```

- Anahtarı bir anahtar sunucusundan indirip içe aktarın:

```
$ 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
```

Bu uygulama, durumu biraz iyileştirir. Bu noktada, belirtilen anahtardaki imzanın doğruluğundan emin olabilirsiniz. Ancak yine de anahtarda kullanılan ada güvenemezsiniz:

```
$ 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
```

Buradaki sorun, herhangi birinin aynı adlı bir anahtar yayınlayabilmesidir. Anahtarın gerçekten ilgili kişinin olduğundan emin olmanız gerekir. Bu konu, GNU gizlilik rehberinin [Herkes açık anahtarlığınızdaki diğer anahtarları doğrulama](#) bölümünde ele alınmıştır. En güvenilir yöntem, geliştiriciyle birebir tanışmak ve anahtar parmak izlerini takas etmektir. Ancak güven ağına da güvenebilirsiniz. Bu şekilde, geliştiriciyle birebir tanışmış başka kişilerin imzalarıyla anahtara geçici olarak güvenebilirsiniz.

Anahtara güvenildikten sonra uyarı görünmez olur:

```
$ 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]
```

İmza geçersiz olursa (arşiv değiştirilmişse), anahtara güvenilip güvenilmediğine bakılmaksızın açık bir hata iletisi görürsünüz:

```
$ 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 Dosya sistemi izinleri

Weblate işleminin, verileri tuttuğu [DATA_DIR](#) klasöründe okuma ve yazma yapabilmesi gerekir. Bu klasördeki tüm dosyaların sahibi, tüm Weblate işlemlerini çalıştıran kullanıcı olmalı ve bu klasöre yazabilmelidir (genellikle WSGI ve Celery, bilgi almak için sunucu ve [Celery ile arka plan görevlerini kullanmak](#)).

Varsayılan yapılandırmada bunlar Weblate kaynaklarıyla aynı ağaca yerleştirilir. Ancak bunları `/var/lib/weblate` gibi daha iyi bir konuma taşımayı yeğleyebilirsiniz.

Weblate bu klasörleri kendiliğinden oluşturmaya çalışır. Ancak bunu yapmak için yeterli izinleri yoksa bunu yapamaz.

[Yönetim komutları](#) komutunu çalıştırırken de dikkatli olmalısınız. Komut aynı Weblate kullanıcısı ile çalıştırılmalıdır yoksa bazı dosyaların izinler yanlış olabilir.

Docker kapsayıcısında, /app/data birimindeki tüm dosyaların sahibi kapsayıcı içindeki weblate kullanıcısı olmalıdır (UID 1000).

Ayrıca bakınız:

Durağan dosyalar sunmak

2.1.5 Weblate için veri tabanı kurulumu

Weblate için PostgreSQL veri tabanı sunucusunun kullanılması önerilir.

Ayrıca bakınız:

Güç bir veri tabanı sunucusu kullanın, Databases, Diğer veri tabanlarından PostgreSQL üzerine aktarmak

PostgreSQL

Django temelli siteler için genellikle en iyi seçim PostgreSQL kullanmaktır. Django tarafından referans olarak kullanılan veri tabanıdır.

Not: Weblate, bazı durumlarda ayrı olarak kurulması gereken trigram eklentisini kullanır. `Postgresql-contrib` ya da benzer şekilde adlandırılmış bir paket arayın.

Ayrıca bakınız:

PostgreSQL notes

PostgreSQL üzerinde bir veri tabanı oluşturmak

Weblate için ayrı bir kullanıcı hesabı ile ayrı bir veri tabanı kullanmak genellikle iyi bir fikirdir:

```
# 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
```

İpucu: Weblate kullanıcılarını PostgreSQL üzerinde süper kullanıcı yapmak istemiyorsanız, bu seçeneği atlayabilirsiniz. Bu durumda, PostgreSQL süper kullanıcısı ile Weblate şemasındaki bazı aktarım adımlarını el ile yapmanız gerekir:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
CREATE EXTENSION IF NOT EXISTS btree_gin WITH SCHEMA weblate;
```

Weblate yapılandırmasını PostgreSQL kullanacak biçimde ayarlamak

PostgreSQL için `settings.py` dosyasına eklenecek kod parçası:

```
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": "",
    }
}
```

Veri tabanı aktarımı, Weblate tarafından kullanılan **ALTER ROLE** veri tabanı rolüyle gerçekleştirir. Çoğu durumda rolün adı kullanıcı adıyla aynıdır. Daha karmaşık kurulumlarda rol adı kullanıcı adından farklıdır ve veri tabanı aktarımı sırasında rolün var olmadığı ile ilgili bir hata iletisi görürsünüz (`psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist`). Bu sorunun PostgreSQL için Azure veri tabanı ile ortaya çıktığı biliniyor. Ancak yalnızca bu durumla sınırlı değildir. Lütfen veri tabanı aktarımı sırasında Weblate tarafından kullanılacak rolün adını **ALTER_ROLE** seçeneğinden ayarlayın.

MySQL ve MariaDB

İpucu: Bazı Weblate özellikleri *PostgreSQL* ile daha iyi çalışır. PostgreSQL veri tabanı, tam metin özelliklerinin kullanılmasını sağlar ve arama ile çeviri belleği işlemlerinde daha üstündür.

Weblate, MySQL ya da MariaDB ile de kullanılabilir. Lütfen bunları Django ile kullanmakla ilgili uyarılar için [MySQL notes](#) ve [MariaDB notes](#) bölümlerine bakın. Bazı sınırlamalar nedeniyle, yeni kurulumlar için *PostgreSQL* kullanılması önerilir.

Weblate için en az MySQL 5.7.8 ya da MariaDB 10.2.7 sürümü kullanılmalıdır.

Weblate için önerilen yapılandırma:

- Daha yüksek Unicode düzlemlerinin (emojiler gibi) görüntülenmesini sağlamak için `utf8mb4` karakter kümesini kullanın.
- Metin alanlarında daha uzun dizinlerin kullanılabilmesini sağlamak için sunucuyu `innodb_large_prefix` seçeneği ile yapılandırın.
- Yalıtım düzeyini `READ COMMITTED` olarak ayarlayın.
- SQL kipi `STRICT_TRANS_TABLES` olarak ayarlanmalıdır.

MySQL 8.x, MariaDB 10.5.x ya da üzeri için varsayılan yapılandırma yeterlidir. Bu nedenle sunucunun ayarlanması gerekmez ve gereken her şey istemci tarafında yapılandırılabilir.

Aşağıda 8 GB RAM belleği olan bir sunucu için `/etc/my.cnf.d/server.cnf` dosyasının örneğini bulabilirsiniz. Bu ayarlar çoğu kurulum için yeterli olmalıdır. MySQL ve MariaDB için, sisteme aynı anda çok sayıda kullanıcının erişmesinin gerekmediği durumlarda sunucu başarımını artıracak seçenekler vardır. Ayrıntılı bilgiyi üretici belgelerinde bulabilirsiniz.

Weblate kurulumunuza başlamadan önce `innodb_file_per_table` seçeneğinin doğru ayarlanması ve MySQL ya da MariaDB veri tabanı sunucusunun yeniden başlatılması kurulum yaparken karşılaşılabilecek sorunları azaltmak için çok önemlidir.

```
[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
```

İpucu: #1071 - Specified key was too long; max key length is 767 bytes hatasını görürseniz, yapılandırmanızı yukarıdaki `innodb` ayarlarına uygun olarak güncelleyin ve kurulumu yeniden başlatın.

İpucu: #2006 - MySQL server has gone away hatasını görürseniz, `CONN_MAX_AGE` seçeneğini yapılandırmak yardımcı olabilir.

Weblate için MySQL/MariaDB yapılandırması

MySQL ve MariaDB için `settings.py` dosyasına eklenecek kod parçası:

```
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": {},
    }
}
```

Kurulumla başlamadan önce MySQL ya da MariaDB veri tabanı sunucusu üzerinde `weblate` kullanıcı hesabını oluşturmanız gerekir. Bunun için şu komutları kullanın:

```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

2.1.6 Diğer yapılandırmalar

Giden e-postayı yapılandırmak

Weblate, hesap etkinleştirme işlemleri ve kullanıcılar tarafından yapılandırılmış çeşitli bildirimler gibi çeşitli durumlar için e-posta gönderir. Bunun için bir SMTP sunucu erişiminin yapılandırılması gerekir.

E-posta sunucusu şu ayarlar kullanılarak yapılandırılır: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_TLS`, `EMAIL_USE_SSL`, `EMAIL_HOST_USER` ve `EMAIL_PORT`. Seçeneklerin adları oldukça açıktır. Ayrıntılı bilgi almak için Django belgelerine bakabilirsiniz.

İpucu: Kimlik doğrulanmasının desteklenmediği ile ilgili bir hata görürseniz (SMTP AUTH extension not supported by server gibi), bu sorun büyük olasılıkla güvenli olmayan bağlantı kullanımından kaynaklanıyordur ve sunucu kimliğinin bu şekilde doğrulanmasını reddeder. Böyle bir durumda `EMAIL_USE_TLS` seçeneğini etkinleştirmeyi deneyin.

Ayrıca bakınız:

Weblate e-postaları alınamıyor, Docker kapsayıcısında giden e-posta ayarlarını yapılandırmak

Ters vekil sunucu arkasında çalıştırmak

Hızı sınırlamak, *İstenmeyen gönderim koruması* ve *Denetim günlüğü* Weblate özellikleri istemci IP adresinin bilinmesine dayanır.

Varsayılan yapılandırmada Weblate, WSGI işleyicisi tarafından ayarlanan `REMOTE_ADDR` IP adresini alır.

Ters vekil sunucu kullanıyorsanız, bu alanda büyük olasılıkla ters vekil sunucu adresi bulunacaktır. Weblate yapılandırmasını ek HTTP üst bilgilerine güvenecek ve IP adresini bunlardan alacak şekilde ayarlamanız gerekir. Bu yapılandırma, ters vekil sunucu kullanmayan kurulumlar için IP adresi sahteciliğine olanak sağlayacağından varsayılan olarak etkinleştirilemez. `IP_BEHIND_REVERSE_PROXY` seçeneğini etkinleştirmek genel kurulumlar için yeterli olabilir. Ancak `IP_PROXY_HEADER` ve `IP_PROXY_OFFSET` seçeneklerini de ayarlamanız gerekebilir.

Dikkat edilmesi gereken başka bir şey de `Host` üst bilgisidir ve `SITE_DOMAIN` olarak yapılandırılmış değerle eşleşmelidir. Ters vekil sunucunuzda ek yapılandırma gerekebilir (örneğin, Apache için `ProxyPreserveHost On` ya da nginx ile `proxy_set_header Host $host;` kullanın).

Ayrıca bakınız:

İstenmeyen gönderim koruması, *Hızı sınırlamak*, *Denetim günlüğü*, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_HEADER`, `IP_PROXY_OFFSET`, `SECURE_PROXY_SSL_HEADER`

HTTP vekil sunucu

Weblate, sürüm denetimi sistemi komutlarını yürütür ve bunlar ortamın vekil sunucu yapılandırmasını alır. Önerilen yaklaşım, `settings.py` dosyasında vekil sunucu ayarlarını belirtmektir:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

Ayrıca bakınız:

Vekil sunucu ortam değişkenleri

2.1.7 Yapılandırmayı ayarlama

Ayrıca bakınız:

Örnek yapılandırma

`weblate/settings_example.py` dosyasını `weblate/settings.py` dosyasına kopyalayın ve kurulumunuza uygun olarak ayarlayın. Büyük olasılıkla şu seçenekleri ayarlamak isteyeceksiniz: `ADMINS`

Bir şeyler ters gittiğinde bildirim alacak site yöneticilerinin listesi. Başarısız olan birleştirme bildirimleri ya da Django hataları gibi.

Ayrıca bakınız:

`ADMINS`, *Yöneticileri düzgün şekilde yapılandırın*

`ALLOWED_HOSTS`

Bu seçeneği, sitenizin sunması gereken barındırma hizmetlerini listeleyecek biçimde ayarlamalısınız. Örneğin:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

Alternatif olarak genel arama karakteri ekleyebilirsiniz:

```
ALLOWED_HOSTS = ["*"]
```

Ayrıca bakınız:

`ALLOWED_HOSTS`, `WEBLATE_ALLOWED_HOSTS`, *Allowed hosts kurulumu*

`SESSION_ENGINE`

Oturumlarınızın nasıl kaydedileceğini yapılandırın. Varsayılan veri tabanı arka uç altyapısını korumanız durumunda, eski oturum verilerini veri tabanından kaldırmak için `weblate clearsessions` görevini zamanlamanız gerekir.

Ön bellek olarak Redis kullanıyorsanız (bilgi almak için: *Ön bellek özelliğini açın*) onu oturumlar için de kullanmanız önerilir:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

Ayrıca bakınız:

Configuring the session engine, `SESSION_ENGINE`

`DATABASES`

Veri tabanı sunucusu ile bağlantı. Ayrıntı bilgi almak için Django belgelerine bakabilirsiniz.

Ayrıca bakınız:

Weblate için veri tabanı kurulumu, `DATABASES`, *Databases*

`DEBUG`

Herhangi bir üretim sunucusunda bu seçeneği devre dışı bırakın. Hata ayıklama kipi etkinleştirildiğinde, Django kullanıcılara hata bildirimlerini görüntüler. Devre dışı bırakıldığında, hatalar e-posta olarak `ADMINS` seçeneğindeki adreslere gönderilir (yukarıya bakın).

Hata ayıklama kipi de Weblate işleyişini yavaşlatır. Çünkü Django bu durumda içeride çok daha fazla bilgi depolar.

Ayrıca bakınız:

`DEBUG`, *Hata ayıklama kipini kapatın*

`DEFAULT_FROM_EMAIL`

Hesap açma e-postaları gibi giden e-postalar için e-posta gönderici adresi.

Ayrıca bakınız:

`DEFAULT_FROM_EMAIL`

`SECRET_KEY`

Django tarafından çerezlerdeki bazı bilgileri imzalamak için kullanılan anahtar. Bilgi almak için: [Django gizli anahtarı](#).

Ayrıca bakınız:

`SECRET_KEY`

`SERVER_EMAIL`

Başarısız olan birleştirme bildirimleri gibi, yöneticiye gönderilecek e-postalarda gönderici adresi olarak kullanılacak e-posta adresleri.

Ayrıca bakınız:

`SERVER_EMAIL`

2.1.8 Veri tabanını doldurmak

Yapılandırmanız hazır olduktan sonra, veri tabanı yapısını oluşturmak için `weblate migrate` komutunu çalıştırabilirsiniz. Bundan sonra yönetici arayüzünü kullanarak çeviri projeleri oluşturabilirsiniz.

Bir kurulumu etkileşimsiz olarak yapmak isterseniz, `weblate migrate --noinput` komutunu kullanabilir ve ardından `createadmin` komutu ile bir yönetici kullanıcı oluşturabilirsiniz.

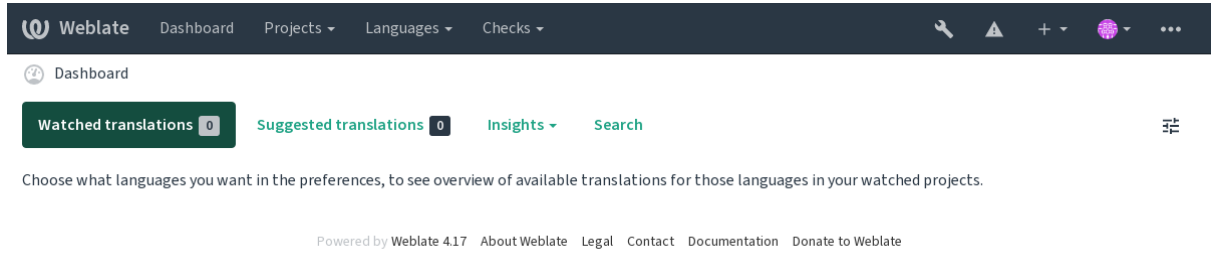
İşiniz bittiğinde, yönetici arayüzündeki `:guilabel:Başarım` raporu bölümüne bakmalısınız. Burada size site yapılandırmasını iyileştirmeniz için ipuçları sunulur.

Ayrıca bakınız:

[Yapılandırma, Yetkiler ve hazır rollerin listesi](#)

2.1.9 Üretim kurulumu

Bir üretim kurulumu için aşağıdaki bölümlerde açıklanan ayarlamaları yapmanız gerekir. Süper kullanıcı olarak oturum açıldığında en önemli ayarlar üst çubukta ünlem simgesi ile bir uyarı olarak görüntülenir:



Django tarafından tetiklenen denetimleri de incelemeniz önerilir (ancak hepsini düzeltmeniz gerekmez):

```
weblate check --deploy
```

Aynı denetim listesini [Yönetim arayüzü](#) bölümünden de gözden geçirebilirsiniz.

Ayrıca bakınız:

[Deployment checklist](#)

Hata ayıklama kipini kapatın

Şununla Django hata ayıklama kipini kaptın (*DEBUG*):

```
DEBUG = False
```

Hata ayıklama kipi açıkken, Django yürütülen tüm sorguları depolar ve kullanıcılara, üretim kipinde gerek duyulmayan hata izlerini görüntüler.

Ayrıca bakınız:

Yapılandırmayı ayarlama

Yöneticileri düzgün şekilde yapılandırın

Sunucuda bir sorun olması durumunda e-postaları kimlerin alacağını belirlemek için doğru yönetici adreslerini *ADMINS* seçeneği ile ayarlayın. Örneğin:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Ayrıca bakınız:

Yapılandırmayı ayarlama

Doğru site etki alanını ayarlayın

Yönetici arabiriminde site adını ve etki alanını ayarlayın. Yoksa RSS ya da kayıt e-postalarındaki bağlantılar çalışmaz. Bu ayar, site etki alanı adının yazılması gereken *SITE_DOMAIN* seçeneği ile yapılandırılır.

4.2 sürümünde değişti: 4.2 sürümünden önce bunun yerine Django site çatısı kullanılıyordu. Bilgi almak için [The “sites” framework](#).

Ayrıca bakınız:

Allowed hosts kurulumu, HTTPS ayarını düzgün biçimde yapın *SITE_DOMAIN*, *WEBLATE_SITE_DOMAIN*, *ENABLE_HTTPS*

HTTPS ayarını düzgün biçimde yapın

Weblate için şifrelenmiş HTTPS iletişim kuralını kullanmanız önemle önerilir. Sertifikanızı hazırladıktan sonra *ENABLE_HTTPS* seçeneğini ayarlamanız gerekir:

```
ENABLE_HTTPS = True
```

İpucu: Ayrıca HSTS özelliğini de etkinleştirmek isteyebilirsiniz. Bilgi almak için: [SSL/HTTPS](#).

Ayrıca bakınız:

ENABLE_HTTPS, *Allowed hosts kurulumu*, *Doğru site etki alanını ayarlayın*

SECURE_HSTS_SECONDS seçeneğini düzgün biçimde ayarlayın

Siteniz SSL üzerinden sunuluyorsa, HTTP sıkı aktarım güvenliği (HTTP Strict Transport Security) özelliğini etkinleştirmek için `settings.py` dosyasında `SECURE_HSTS_SECONDS` değerini ayarlamayı değerlendirmeniz gerekir. Bu seçenek varsayılan olarak, aşağıda gösterildiği gibi 0 olarak ayarlanmıştır.

```
SECURE_HSTS_SECONDS = 0
```

Sıfır olmayan bir tamsayı değerine ayarlanırsa, `django.middleware.security.SecurityMiddleware` üst bilgisi bulunmayan tüm yanıtlarda **HTTP Strict Transport Security** üst bilgisini ayarlar.

Uyarı: Bu ayar yanlış yapılsa, sitenizi geri dönüşü olmayan bir şekilde (bir süreliğine) bozabilir. Önce **HTTP Strict Transport Security** belgelerini okuyun.

Güç bir veri tabanı sunucusu kullanın

- Lütfen üretim ortamı için PostgreSQL kullanın. Bilgi almak için: *Weblate için veri tabanı kurulumu*.
- Veri tabanı sunucusunu çalıştırmak için yakın konumları kullanın. Yoksa ağ başarımı ya da güvenilirliği sorunları Weblate deneyiminizi mahvedebilir.
- Veri tabanı sunucusunun başarımını denetleyin veya yapılandırmasını değiştirin. Örneğin **PGTune** kullanabilirsiniz.

Ayrıca bakınız:

Weblate için veri tabanı kurulumu, *Diğer veri tabanlarından PostgreSQL üzerine aktarmak*, *Yapılandırmayı ayarlama*, *Databases*

Ön bellek özelliğini açın

Olabiliyorsa, CACHES yapılandırma değişkenini ayarlayarak Django Redis kullanın. Örneğin:

```
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",
        },
    },
}
```

İpucu: Ön bellek için Redis ayarlarını değiştirirseniz, bunları Celery için de ayarlamanız gerekebilir. Bilgi almak için: `:ref: Celery`.

Ayrıca bakınız:

Avatar ön belleği, Django's cache framework

Avatar ön belleği

Django ön belleğinin yanında, Weblate avatarları da ön belleğe alır. Bu amaçla ayrı, dosya temelli bir ön bellek kullanılması önerilir:

```
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,
        },
    },
}
```

Ayrıca bakınız:

[ENABLE_AVATARS](#), [AVATAR_URL_PREFIX](#), *Avatarlar*, *Ön bellek özelliğini açın*, Django's cache framework

E-posta gönderimini yapılandırın

Weblate tarafından bazı e-postaların gönderilmesi gerekir ve bu e-postaların doğru bir gönderici adresi olmalıdır. Lütfen [SERVER_EMAIL](#) ve [DEFAULT_FROM_EMAIL](#) değerlerini ortamınıza uygun şekilde yapılandırın. Örneğin:

```
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"
```

Not: Weblate tarafından e-posta gönderilmesini devre dışı bırakmak için [EMAIL_BACKEND](#) seçeneğini `django.core.mail.backends.dummy.EmailBackend` olarak ayarlayın.

Bu yapılandırma, kayıt ve parola sıfırlama e-postaları ile birlikte *tüm* e-posta gönderimini devre dışı bırakır.

Ayrıca bakınız:

Yapılandırmayı ayarlama, *Giden e-postayı yapılandırmak*, [EMAIL_BACKEND](#), [DEFAULT_FROM_EMAIL](#), [SERVER_EMAIL](#)

Allowed hosts kurulumu

Django, sitenizin sunmasına izin verilen etki alanı adlarının listesini [ALLOWED_HOSTS](#) seçeneğinde tutar. Bu seçenek boş bırakılırsa tüm istekler engellenir.

Bu seçenek, HTTP sunucunuzla eşleşecek şekilde yapılandırılmamışsa, Invalid HTTP_HOST header: '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS. gibi hata iletileri görürsünüz

İpucu: Docker kapsayıcısında [WEBLATE_ALLOWED_HOSTS](#) seçeneği kullanılabilir.

Ayrıca bakınız:

`ALLOWED_HOSTS`, `WEBLATE_ALLOWED_HOSTS`, *Doğru site etki alanını ayarlayın*

Django gizli anahtarı

`SECRET_KEY` ayarı Django tarafından çerezleri imzalamak için kullanılır. Örnek kurulumdaki değeri kullanmayıp kendiniz için oluşturacağınız gerçek bir değeri kullanmanız gerekir.

Weblate içindeki `weblate-generate-secret-key` komutunu kullanarak yeni bir anahtar oluşturabilirsiniz.

Ayrıca bakınız:

`SECRET_KEY`

Bakım görevlerini yürütmek

En iyi başarımlar için, bazı bakım görevlerini arka planda çalıştırmak iyi bir fikirdir. Bu işlemler *Celery ile arka plan görevlerini kullanmak* tarafından kendiliğinden yapılır ve şu görevleri kapsar:

- Yapılandırma sağlığı denetimi (saatlik).
- Bekleyen değişiklikleri işlemek (saatlik). Bilgi almak için: *Lazy commit işlemleri* ve `commit_pending`.
- Bileşen uyarılarını güncellemek (günlük).
- Uzak dalları güncellemek (gecelik). Bilgi almak için: `:setting: 'AUTO_UPDATE'`.
- Çeviri belleğinin JSON yedeğini almak (günlük). Bilgi almak için: *dump_memory*.
- Tam metin ve veri tabanı bakım görevleri (günlük ve haftalık görevler). Bilgi almak için: *cleanuptrans*.

3.2 sürümünde değişti: 3.2 sürümünden bu yana, bu görevleri yapmak için varsayılan olarak Celery kullanılır. Weblate içinde buna uygun yapılandırma hazırdır. Bilgi almak için: ref: *celery*.

Sistem yerel ayarları ve kodlama

Sistem yerel ayarları UTF-8 destekleyenler ile yapılandırılmalıdır. Çoğu Linux dağıtımında varsayılan ayar böyledir. Sisteminizde durumun böyle olmaması durumunda, lütfen yerel ayarları UTF-8 çeşidi olarak değiştirin.

Örneğin, `/etc/default/locale` dosyasını düzenleyerek `LANG = "C.UTF-8"` olarak ayarlayabilirsiniz.

Bazı durumlarda, her hizmetin ayrı yerel ayar yapılandırması vardır. Bunlar, dağıtımlara ve site sunucularına göre farklılık gösterebilir. Bu konuda bilgi almak için site sunucusu paketlerinizin belgelerine bakın.

Ubuntu üzerinde Apache `/etc/apache2/envvars` kullanır:

```
export LANG='en_US.UTF-8'
export LC_ALL='en_US.UTF-8'
```

CentOS üzerinde Apache `/etc/sysconfig/httpd` (or `/opt/rh/httpd24/root/etc/sysconfig/httpd`) kullanır:

```
LANG='en_US.UTF-8'
```

Özel sertifika kurulumu kullanmak

Weblate, HTTP istekleri sırasında SSL sertifikalarını doğrular. Varsayılan paketlerde güvenilmeyen bir özel sertifika kurulumu kullanıyorsanız, sertifikasını güvenilir olarak eklemeniz gerekir.

Yeğlenen yaklaşım bunu sistem düzeyinde yapmaktır. Ayrıntılı bilgi almak için dağıtım belgelerinize bakın (örneğin debian üzerinde bu işlem, sertifika kurulumunun sertifikasını `/usr/local/share/ca-certificates/` içine yerleştirip `update-ca-certificates` komutunu çalıştırarak yapılabilir).

Bu işlem yapıldıktan sonra, sistem araçları sertifikaya güvenir ve Git tarafından da bu sertifika kullanılır.

Python kodu için, kendi içinde gelen sertifika yerine istekleri sistem sertifika kurulumu paketini kullanacak şekilde yapılandırmanız gerekir. Bu işlem, aşağıdaki kod parçasını `settings.py` dosyasına ekleyerek yapılabilir (yol Debian sistemine özgüdür):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

İstemci varlıklarının sıkıştırılması

Weblate, bazı JavaScript ve CSS dosyalarıyla birlikte gelir. Başarımı artırmak için bunların bir istemciye gönderilmeden önce sıkıştırılması iyidir. Varsayılan yapılandırmada bu işlem çok az ek yük oluşturarak anında yapılır. Büyük kurulumlarda, çevrimdışı sıkıştırma kipini etkinleştirmeniz önerilir. Bunun yapılandırmada ayarlanması ve sıkıştırmanın her Weblate yükseltmesinde tetiklenmesi gerekir.

Yapılandırma, `django.conf.settings.COMPRESS_OFFLINE` seçeneği ve `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` seçeneği etkinleştirilerek kolayca ayarlanır (ikincisi zaten örnek yapılandırmalara eklenmiştir):

```
COMPRESS_OFFLINE = True
```

Her dağıtımda, dosyaları geçerli sürümle eşleşecek şekilde sıkıştırmanız gerekir:

```
weblate compress
```

İpucu: Resmi Docker kalıbında bu özellik etkinleştirilmiştir.

Ayrıca bakınız:

[Common Deployment Scenarios](#), *Durağan dosyalar sunmak*

2.1.10 Sunucuyu çalıştırmak

İpucu: Aşağıda açıklanan hizmetlerle ilgili deneyiminiz yoksa, *Docker ile kurmak* adresine bakmak isteyebilirsiniz.

Weblate çalıştırmak için birkaç hizmete gerek duyacaksınız. Önerilen kurulum şunlardan oluşur:

- Veri tabanı sunucusu (bilgi almak için: *Weblate için veri tabanı kurulumu*)
- Ön bellek sunucusu (bilgi almak için: *Ön bellek özelliğini açın*)
- Durağan dosyalar ve SSL ucu için ön yüz site sunucusu (bilgi almak için: *Durağan dosyalar sunmak*)
- Devingen içerik için WSGI sunucusu (bilgi almak için: *NGINX ve uWSGI için örnek yapılandırma*)
- Arka plan görevlerini yürütmek için Celery (bilgi almak için: *Celery ile arka plan görevlerini kullanmak*)

Not: Hizmetler arasında bazı bağımlılıklar bulunur. Örneğin Celery veya uwsgi işlemlerini başlatırken ön bellek ve veri tabanı çalışıyor olmalıdır.

Çoğu durumda, tüm hizmetleri tek (sanal) sunucu üzerinde çalıştırırsınız. Ancak kurulumunuzun yükü ağırsa, hizmetleri ayırabilirsiniz. Bu durumda, Celery ve wsgi sunucularının `DATA_DIR` klasörüne erişmesini sağlamanız yerlidir.

Not: WSGI işlemini çalıştıran kullanıcı ile Celery işlemini çalıştıran kullanıcı aynı olmalıdır. Yoksa `DATA_DIR` klasöründeki dosyaların sahiplikleri karışır ve bu da çalışma sırasında sorunlara yol açar.

Ayrıca *Dosya sistemi izinleri* ve *Celery ile arka plan görevlerini kullanmak* bölümlerine bakın.

Web sunucusunu çalıştırmak

Weblate çalıştırmak, diğer Django temelli programları çalıştırmaktan farklı değildir. Django genellikle uWSGI ya da fcgi olarak çalıştırılır (aşağıda farklı site sunucuları için örnekler bulabilirsiniz).

Deneme amacıyla, Django üzerindeki yerleşik site sunucusunu kullanabilirsiniz:

```
weblate runserver
```

Uyarı: BU SUNUCUYU ÜRETİM AYARIYLA KULLANMAYIN. Güvenlik denetimlerinden ya da başarımlarından geçirilmemiştir. Ayrıca `runserver` ile ilgili Django belgelerine bakın.

İpucu: Django içindeki sunucu, yalnızca geliştirme amaçlı olduğundan yalnızca `DEBUG` seçeneği etkin olan duran dosyaları sunar. Üretim kullanımı için lütfen *NGINX ve uWSGI için örnek yapılandırma*, *Örnek Apache yapılandırması*, *Örnek Apache ve Gunicorn yapılandırması* ve *Duran dosyalar sunmak* içindeki wsgi kurulumlarına bakın.

Duran dosyalar sunmak

Duran Django dosyalarının tek bir klasörde toplanması gerekiyor. Bunun için `weblate collectstatic --noinput` komutunu çalıştırın. Bu komut, duran dosyaları `STATIC_ROOT` seçeneği ile belirtilen bir klasöre kopyalar (bu klasör, varsayılan olarak `DATA_DIR` içindeki `static` klasörüdür).

Duran dosyaları doğrudan site sunucunuzdan sunmanız önerilir. Bunu şu yollar ile kullanmalısınız:

/static/

Weblate ve yönetici arayüzünün duran dosyalarını sunar (`STATIC_ROOT` ile tanımlanmış).

/media/

Kullanıcıların ortam yüklemeleri için kullanılır (ekran görüntüleri gibi).

/favicon.ico

`/static/favicon.ico` dosyasını sunmak için bir kural yeniden yazılmalıdır.

Ayrıca bakınız:

NGINX ve uWSGI için örnek yapılandırma, *Örnek Apache yapılandırması*, *Örnek Apache ve Gunicorn yapılandırması*, *İstemci varlıklarının sıkıştırılması*, *How to deploy Django*, *How to deploy static files*

İçerik güvenliği ilkesi

Varsayılan Weblate yapılandırması, Content-Security-Policy veya X-XSS-Protection gibi güvenlikle ilgili HTTP üst bilgilerini ayarlayan `weblate.middleware.SecurityMiddleware` ara yazılımını etkinleştirir. Bunlar varsayılan olarak Weblate ve yapılandırması ile çalışacak şekilde ayarlanmıştır. Ancak bunun için ortamınıza uygun özelleştirmeler yapmanız gerekebilir.

Ayrıca bakınız:

`CSP_SCRIPT_SRC`, `CSP_IMG_SRC`, `CSP_CONNECT_SRC`, `CSP_STYLE_SRC`, `CSP_FONT_SRC`

NGINX ve uWSGI için örnek yapılandırma

Üretim amacıyla kullanılacak site sunucusu çalıştırmak için, Weblate ile kurulan wsgi sarmalayıcısını kullanın (sanal ortamdaysanız `~/weblate-env/lib/python3.9/site-packages/weblate/wsgi.py` olarak kurulur). Kullandığınız virtualenv üzerinde Python arama yolunu ayarlamayı da unutmayın (uWSGI üzerinde `virtualenv = /home/user/weblate-env` gibi).

Aşağıdaki yapılandırma, NGINX site sunucusu üzerinde uWSGI olarak Weblate çalıştırır.

NGINX için yapılandırma (`weblate/examples/weblate.nginx.conf` dosyasında da bulunabilir):

```
#
# 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:

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
uwsgi_pass unix:///run/uwsgi/app/weblate/socket;
# uwsgi_pass 127.0.0.1:8080;
}
}
```

uWSGI için yapılandırma (weblate/examples/weblate.uwsgi.ini dosyasında da bulunabilir):

```
#
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# Do not log some errors caused by client disconnects
ignore-sigpipe = true
ignore-write-errors = true
disable-write-exception = true
```

Ayrıca bakınız:

How to use Django with uWSGI

Örnek Apache yapılandırması

Weblate ile WSGI kullanırken prefork MPM kullanılması önerilir.

Aşağıdaki yapılandırma WSGI olarak Weblate çalıştırır. `mod_wsgi` modülünün etkinleştirilmiş olması gerekir (weblate/examples/apache.conf dosyasında bulunabilir):

```
#
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    </Files>
  </Directory>

</VirtualHost>

```

Not: Weblate için Python 3 gereklidir. Bu nedenle modwsgi modülünün Python 3 çeşidini çalıştırdığınızdan emin olun. Genellikle libapache2-mod-wsgi-py3 gibi ayrı bir paket olarak bulunur.

Ayrıca bakınız:

Sistem yerel ayarları ve kodlama, [How to use Django with Apache and mod_wsgi](#)

Örnek Apache ve Gunicorn yapılandırması

Aşağıdaki yapılandırma Gunicorn ve Apache 2.4 üzerinde Weblate çalıştırır (weblate/examples/apache.gunicorn.conf dosyasında bulunabilir):

```

#
# 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/

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
ProxyPreserveHost On
</VirtualHost>
```

Ayrıca bakınız:

How to use Django with Gunicorn

Bir yol altında Weblate çalıştırma yapılandırması

Weblate ile WSGI kullanırken prefork MPM kullanılması önerilir.

Weblate çalıştırmak için /weblate gibi bir yol kullanan örnek Apache yapılandırması. Yine mod_wsgi kullanarak (weblate/examples/apache-path.conf dosyasında bulunabilir):

```
#
# VirtualHost for Weblate, running under /weblate path
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /weblate/favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /weblate/static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>

    # DATA_DIR/media/
    Alias /weblate/media/ /home/weblate/data/media/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate_
    ↳request-timeout=600
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

    WSGIScriptAlias /weblate /home/weblate/weblate-env/lib/python3.9/site-packages/
    ↳weblate/wsgi.py process-group=weblate
    WSGIPassAuthorization On

    <Directory /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/>
        <Files wsgi.py>
            Require all granted
        </Files>
    </Directory>
</VirtualHost>
```

Ek olarak `weblate/settings.py` dosyasını ayarlamalısınız:

```
URL_PREFIX = "/weblate"
```

2.1.11 Celery ile arka plan görevlerini kullanmak

3.2 sürümünde geldi.

Weblate, düzenli olarak yapılan arka plan görevlerini yerine getirmek için Celery kullanır. Bu işlemleri yapacak bir Celery hizmeti çalıştırmanız gerekiyor. Örnek olarak, aşağıdaki işlemlerin yapılı (bu liste tam değildir):

- Dış hizmetlerden internet kancalarını almak (bilgi almak için: *Bildirim kancaları*).
- Yedeklemeler, temizlemeler, günlük eklentiler veya güncellemeler gibi düzenli bakım görevlerini çalıştırmak (bilgi almak için *Weblate yedeğini alma ve taşıma*, `BACKGROUND_TASKS`, *Eklentiler*).
- *Kendiliğinden çeviri* çalıştırmak.
- Toplu bildirimleri göndermek.
- Yük oluşturan işlemleri wsgi işlemi üzerinden almak.
- Bekleyen değişiklikleri göndermek (Bilgi almak için: *Lazy commit işlemleri*).

Arka uçta Redis kullanan tipik bir kurulum şöyle görünür:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

Ayrıca bakınız:

[Celery içindeki Redis işleyici yapılandırması](#)

Görevleri işlemek ve zamanlanmış görevleri başlatmak için Celery işlemini de başlatmalısınız. Bu işlem doğrudan komut satırından yapılabilir (genellikle hata ayıklama veya geliştirme sırasında yararlıdır):

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

Not: Celery işlemini çalıştıran kullanıcı, WSGI işlemini çalıştıran kullanıcı ile aynı olmalıdır. Yoksa `DATA_DIR` klasöründeki dosyaların sahiplikleri karışır ve bu da çalışma sırasında sorunlara yol açar.

Ayrıca *Dosya sistemi izinleri* ve *Sunucuyu çalıştırmak* bölümlerine bakabilirsiniz.

Wsgi üzerinde Celery görevlerini yürütürken eager kipini kullanmak

Not: Bu seçeneğin kullanılmasının site arayüzü başarımı üzerinde ciddi etkisi olur ve normal tetikleyiciye bağlı olarak özellikleri bozar (bekleyen değişiklikleri işlemek, özet bildirimleri veya yedeklemeler gibi).

Geliştirme ortamları için, tüm görevleri anında işleyen eager yapılandırmasını kullanmak isteyebilirsiniz:

```
CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True
```

Celery uygulamasını sistem hizmeti olarak çalıştırmak

Büyük olasılıkla `celery:userguide/daemonizing` bölümünde anlatıldığı gibi bir Celery daemon çalıştırmak isteyeceksiniz. `systemd` kullanan en yaygın Linux kurulumu için, hazır gelen ve aşağıda listesi bulunan `examples` klasöründeki örnek dosyaları kullanabilirsiniz.

`/etc/systemd/system/celery-weblate.service` olarak yerleştirilecek `systemd` birimi:

```
[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
```

`/etc/default/celery-weblate` olarak yerleştirilecek ortam yapılandırması:

```
# Name of nodes to start
CELERYD_NODES="celery notify memory backup translate"

# Absolute or relative path to the 'celery' command:
CELERY_BIN="/home/weblate/weblate-env/bin/celery"

# App instance to use
# comment out this line if you don't use an app
CELERY_APP="weblate.utils"

# Extra command-line arguments to the worker,
# increase concurrency if you get weblate.E019
CELERYD_OPTS="--beat:celery --queues:celery =celery --prefetch-
↳multiplier:celery =4 \
  --queues:notify =notify --prefetch-multiplier:notify =10 \
  --queues:memory =memory --prefetch-multiplier:memory =10 \
  --queues:translate =translate --prefetch-multiplier:translate =4 \
  --concurrency:backup =1 --queues:backup =backup --prefetch-multiplier:backup =2
↳"

# Logging configuration
# - %n will be replaced with the first part of the nodename.
# - %I will be replaced with the current child process index
#   and is important when using the prefork pool to avoid race conditions.
CELERYD_PID_FILE="/run/celery/weblate-%n.pid"
CELERYD_LOG_FILE="/var/log/celery/weblate-%n%I.log"
CELERYD_LOG_LEVEL="INFO"
```

Celery günlüklerini döndürmek için **logrotate** komutu kullanılarak `/etc/logrotate.d/celery` olarak yerleştirilecek ek yapılandırma:

```
/var/log/celery/*.log {
    weekly
    missingok
    rotate 12
    compress
    notifempty
}
```

Celery atımını kullanarak görevleri zamanlamak

Weblate, hazır bir zamanlanmış görevler kurulumuyla gelir. Bununla birlikte, `settings.py` dosyasından ek görevler tanımlayabilirsiniz. Örnek olarak *Lazy commit işlemleri* bölümüne bakabilirsiniz.

Görevlerin Celery beats daemon tarafından yürütülmesi gerekiyor. Beklediği gibi çalışmıyorsa, daemon çalışmıyor veya veri tabanı bozulmuş olabilir. Bu durumda temel sorunu bulmak için Celery başlangıç günlüklerine bakın.

Celery durumunu izlemek

Celery görev kuyruklarının geçerli uzunluğunu *Yönetim arayüzü* içinden ya da komut satırında `celery_queues` komutunu kullanarak görebilirsiniz. Kuyruğun çok uzaması durumunda, yönetici arayüzünde de yapılandırma hatası görürsünüz.

Uyarı: Celery hataları varsayılan olarak yalnızca Celery günlüğüne kaydedilir ve kullanıcı tarafından görülemez. Bu tür hataların özetini görmek istiyorsanız, *Hata raporlarını derlemek* seçeneğini yapılandırmanız önerilir.

Ayrıca bakınız:

Weblate uygulamasını izlemek, *Weblate kurulumunun düzgün olup olmadığını nasıl denetleyebilirim?*, *Configuration and defaults*, *Workers Guide*, *Daemonization*, *Monitoring and Management Guide*, `celery_queues`

2.1.12 Weblate uygulamasını izlemek

Weblate, örneğin Kubernetes gibi basit durum denetimlerinde kullanılmak üzere `/healthz/` adresini sunar. Docker kapsayıcısında, bu adresi kullanan iç sistem durumu denetimi bulunur.

Weblate ölçümlerini izlemek için `GET /api/metrics/` API uç noktasını kullanabilirsiniz.

Ayrıca bakınız:

Weblate kurulumunun düzgün olup olmadığını nasıl denetleyebilirim?, *Celery durumunu izlemek*, *Munin için Weblate uygulama eki*

2.1.13 Hata raporlarını derlemek

Weblate, diğer yazılımlar gibi sorun çıkarabilir. Yardımcı olabilecek sorun durumlarını derlemek için üçüncü taraf hizmetlerini kullanmanızı öneririz. Bu uygulama, özellikle Celery görevlerinin yapılamaması durumunda kullanışlıdır. Yoksa hata yalnızca günlüklere bildirilir ve bunlar ile ilgili bildirim almazsınız. Weblate tarafından desteklenen hizmetler şunlardır:

Sentry

Weblate, [Sentry](#) desteği sunar. Kullanmak için, `settings.py` dosyasında `SENTRY_DSN` seçeneğini ayarlamak yeterlidir:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

Weblate, [Rollbar](#) desteği sunar. Kullanmak için, [Rollbar notifier for Python](#) yönergelerini izlemek yeterlidir.

Özetle, `settings.py` dosyasını ayarlamanız gerekir:

```
# 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",
}
```

Bunun dışındaki her şey kendiliğinden bütünleştirilmiştir. Artık hem sunucu hem de istemci tarafı hatalarını derleyebilirsiniz.

Not: Hata günlüğü, düzgün bir şekilde işlenen özel durumları da içerir. Ancak yüklenen bir dosyanın işlenememesi gibi bir sorunu da gösteriyor olabilir.

2.1.14 Weblate kurulumunu başka bir sunucuya aktarmak

Weblate kurulumunu başka bir sunucuya aktarmak oldukça kolaydır. Ancak veriler dikkatli bir şekilde aktarmanız gereken birkaç konumda bulunur. En iyi yaklaşım, aktarım sırasında Weblate kopyasını durdurmaktır.

Veri tabanını aktarmak

Veri tabanı arka ucunuza bağlı olarak, veri tabanını aktarmak için birkaç seçeneğiniz olabilir. En basit yaklaşım, genellikle en etkili araçlar olduklarından, veri tabanının kendi araçlarını kullanmaktır (**mysqldump** ya da **pg_dump** gibi). Alternatif olarak, veri tabanınızın desteklemesi durumunda çoğaltma (replication) özelliğini kullanabilirsiniz.

Ayrıca bakınız:

Veri tabanları arasında aktarım *Diğer veri tabanlarından PostgreSQL üzerine aktarmak* bölümünde açıklanmıştır.

Sürüm denetimi sistemi depolarını aktarmak

`DATA_DIR` altına kaydedilmiş sürüm denetimi sistemi depolarının da taşınması gerekir. Aktarımı daha etkili bir şekilde yapmak için bunları kopyalayabilir ya da **rsync** komutunu kullanabilirsiniz.

Diğer notlar

Weblate tarafından kullanılıyor olabilecek, Redis, zamanlanmış Cron görevleri veya özel kimlik doğrulama arka uçları gibi diğer hizmetleri taşımayı unutmayın.

2.2 Weblate dağıtımları

Weblate, bulutunuza kolayca kurulabilir. Lütfen platformunuz için ayrıntılı bilgiler içeren rehbera bakın:

- *Docker ile kurmak*
- *OpenShift üzerine kurulum*
- *Kubernetes üzerine kurulum*

2.2.1 Üçüncü taraf Weblate dağıtımları

Not: Şu dağıtımlar Weblate ekibi tarafından geliştirilmez ya da desteklenmez. Kurulum aşamaları bu belgede anlaşılandır farklı olabilir.

Bitnami Weblate paketi

Bitnami, <<https://bitnami.com/stack/weblate>> adresinden birçok platform için hazır Weblate paketleri sunar.

Ayrıca bakınız:

[Bitnami Weblate paketi](#)

Cloudron Weblate paketi

Cloudron internet uygulamalarını kendinizin barındırabileceğiniz bir platformdur. Cloudron ile kurulan Weblate kendiliğinden güncel tutulur. Paket, Cloudron ekibi tarafından [Weblate paket deposu](#) üzerinde güncel tutulur.



YunoHost üzerinde Weblate

Kendi kendine barındırma projesi [YunoHost](#) bir Weblate paketi sağlar. YunoHost kurulumunuzu yaptıktan sonra, Weblate başka bir uygulama olarak kurulabilir. Yedekleme ve geri yükleme ile tam olarak çalışan bir paket sunar. Bununla birlikte yine de belirli kullanımlar için ayarlar dosyanızı düzenlemeniz gerekebilir.

Yönetim arayüzünü ya da bu düğmeyi kullanabilirsiniz (sizi sunucunuza götürür):



Komut satırı arayüzü de kullanılabilir:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

2.3 Weblate yazılımını güncellemek

2.3.1 Docker kalıbı güncellemeleri

Resmi Docker kalıbına (bilgi almak için [Docker ile kurmak](#)) tüm Weblate yükseltme adımları bütünleştirilmiştir. Genellikle güncel sürümü çekmenin dışında el ile yapılması gereken bir adım bulunmaz.

Ayrıca bakınız:

[Docker kapsayıcısını yükseltmek](#)

2.3.2 Genel yükseltme yönergeleri

Yükseltmeden önce, lütfen değişmiş olabilecekleri için güncel [Yazılım gereksinimleri](#) değerini denetleyin. Tüm gereksinimler kurulduktan ya da güncellendikten sonra, lütfen `settings.py` dosyasını yapılandırmadaki değişikliklere uygun şekilde ayarlayın (doğru değerler için `settings_example.py` bölümüne bakabilirsiniz).

Yükseltmeden önce her zaman [Sürüme özel yönergeler](#) bölümünü denetleyin. Bazı sürümleri atlamamız durumunda, lütfen yükseltmede atladığınız tüm sürümlerin yönergelerini izleyin. Bazen sorunsuz bir aktarım yapmak için önce bazı ara sürümlere yükseltmek daha iyidir. Bir kerede birkaç sürüm yukarı yükseltilebilir, ancak tek sürüm yükseltmelerindeki kadar iyi denenmemiştir.

Not: Yükseltmenin yapılamaması durumunda veri tabanını önceki sürüme döndürebilmeniz için yükseltmeden önce veri tabanının tam yedeğini almanız önerilir. Bilgi almak için: [Weblate yedeğini alma ve taşıma](#).

1. wsgi ve Celery işlemlerini durdurun. Yükseltme işlemi, veri tabanında uyumsuz değişiklikler yapabilir. Bu nedenle yükseltme sırasında eski işlemlerin çalışmasını önlemek her zaman daha güvenli olur.
2. Weblate kodunu yükseltin.

Pip kurulumları için şöyle yapılabilir:

```
pip install -U "Weblate[all] ==version"
```

Ya da yalnızca güncel sürümü edinmek istiyorsanız:

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

İsteğe bağlı bağımlılıkların tümünü kurmak istemiyorsanız şunu yapın:

```
pip install -U Weblate
```

Git checkout ile yeni kaynak kodunu almanız ve kurulumunuzu güncellemeniz gerekir:

```
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. Yeni Weblate sürümünde yeni [İsteğe bağlı bağımlılıklar](#) bulunabilir. Lütfen istediğiniz özellikleri kapsayıp kapsamadığını denetleyin.

4. Yapılandırma dosyasını yükseltin. Gerekli adımlar için `settings_example.py` ya da [Sürüme özel yönergeler](#) dosyalarına bakabilirsiniz.

5. Veri tabanı yapısını yükseltin:

```
weblate migrate --noinput
```

6. Güncellenmiş durağan dosyaları derleyin (bilgi almak için: [Sunucuyu çalıştırmak](#) ve [Durağan dosyalar sunmak](#)):

```
weblate collectstatic --noinput --clear
```

7. JavaScript ve CSS dosyalarını sıkıştırın (isteğe bağlı, bilgi almak için: [İstemci varlıklarının sıkıştırılması](#)):

```
weblate compress
```

8. Git üzerindeki sürümü çalıştırıyorsanız, her yükseltme yaptığınızda yerel ayar dosyalarını da yeniden oluşturmamız gerekir. Bu işlemi şu şekilde yapabilirsiniz:

```
weblate compilemessages
```

9. Kurulumunuzun doğru olduğunu doğrulayın (bilgi almak için: `:ref:`production``):

```
weblate check --deploy
```

10. Celery işlemini yeniden başlatın (bilgi almak için: [Celery ile arka plan görevlerini kullanmak](#)).

2.3.3 Sürüme özel yönergeler

2.x sürümünden yükseltmek

2.x sürümünden yükseltiyorsanız, her zaman ilk olarak 3.0.1 sürümüne yükseltin. Ardından 3.x serisinde yükseltme işlemini sürdürün. Bu adımı atlayan yükseltme işlemleri desteklenmez ve bozulur.

Ayrıca bakınız:

[Weblate 3.0 belgelerinde 2.20 sürümünden 3.0 sürümüne yükseltme](#)

3.x sürümünden yükseltmek

3.x sürümünden yükseltme yapıyorsanız, her zaman ilk olarak 4.0.4 veya 4.1.1 sürümüne yükseltin. Ardından 4.x serisinde yükseltmeyi sürdürün. Bu adımı atlayan yükseltmeler desteklenmez ve bozulur.

Ayrıca bakınız:

[Weblate 4.0 belgelerinde 3.11 sürümünden 4.0 sürümüne yükseltmek](#)

4.0 sürümünden 4.1 sürümüne yükseltmek

Güncellemek için [Genel yükseltme yönergeleri](#) adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- `settings_example.py` içinde birkaç değişiklik yapıldı. En önemlileri ara yazılım değişiklikleri. Lütfen ayarlarınızı buna göre yapın.
- Yeni dosya biçimleri eklendi. `WEBLATE_FORMATS` seçeneğini değiştirdiyseniz bunları katmak isteyebilirsiniz.
- Yeni kalite denetimleri eklendi. `CHECK_LIST` seçeneğini değiştirdiyseniz bunları katmak isteyebilirsiniz.
- API hız sınırlamasının raporlanmasını sağlamak için `DEFAULT_THROTTLE_CLASSES` ayarında değişiklik yapıldı.

- Bazı gereksinimler eklendi ve güncellendi.
- `INSTALLED_APPS` içinde bir değişiklik yapıldı.
- `MT_DEEPL_API_VERSION` ayarı 4.7 sürümünde kaldırıldı. *DeepL* makine çevirisi artık yeni `MT_DEEPL_API_URL` kullanıyor. `MT_DEEPL_API_URL` seçeneğini aboneliğinize uygun şekilde ayarlamanız gerekebilir.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.1 sürümünden 4.2 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- 3.x sürümlerinden yükseltme işlemi artık desteklenmiyor. Lütfen önce 4.0 ya da 4.1 sürümüne yükseltin.
- Bazı gereksinimler eklendi ve güncellendi.
- `settings_example.py` içinde, en dikkat çekicileri yeni ara katman yazılımında ve değişen uygulama sıralamasında olan birkaç değişiklik yapıldı.
- JSON tabanlı biçimlerin anahtarlarından baştaki nokta kaldırıldı. Dizgeler veri tabanı aktarımı sırasında ayarlanır. Ancak dış aktarma işlemlerinde ya da API üzerinde anahtarlara güvenmeniz durumunda dış bileşenlerin ayarlanması gerekebilir.
- Celery yapılandırması `memory` kuyruğu kullanılmayacak şekilde değiştirildi. Lütfen başlangıç betiklerinizi ve `CELERY_TASK_ROUTES` seçeneğini ayarlayın.
- Weblate etki alanı yapılandırması ayarlar bölümüne alındı. Bilgi almak için: *SITE_DOMAIN* (ya da *WEBLATE_SITE_DOMAIN*). Weblate çalıştırmadan önce bu ayarı yapılandırmanız gerekir.
- Kullanıcı veri tabanındaki kullanıcı adı ve e-posta alanları artık büyük/küçük harfe duyarız olarak eşsiz. PostgreSQL için yanlışlıkla zorunlu kılınmamıştı.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.2 sürümünden 4.3 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- Kalite denetimlerinde bazı değişiklikler yapıldı. *CHECK_LIST* seçeneğini değiştirdiyseniz bunları katmak isteyebilirsiniz.
- Kaynak dil öznetliği, projeden API üzerinde kullanıma sunulan bir bileşene taşındı. *Weblate istemcisi* kullanıyorsanız güncellenmeniz gerekir.
- Veri tabanının 4.3 sürümüne aktarımı, çevirdiğiniz dizge sayısına bağlı olarak uzun sürebilir (100.000 kaynak dizge için aktarım yaklaşık bir saat sürebilir).
- `INSTALLED_APPS` içinde bir değişiklik yapıldı.
- `django:SESSION_COOKIE_AGE_AUTHENTICATED` seçeneğini tamamlayan `SESSION_COOKIE_AGE` ayarı eklendi.
- GitHub ya da GitLab bütünlüştürmesi için **hub** ya da **lab** kullanıyorsanız, bunu yeniden yapılandırmanız gerekir. Bilgi almak için: *GITHUB_CREDENTIALS* ve `:setting: GITLAB_CREDENTIALS`.

4.3.1 sürümünde değişti:

- Celery yapılandırması `memory` kuyruğu eklenecek şekilde değiştirildi. Lütfen başlangıç betiklerinizi ve `CELERY_TASK_ROUTES` seçeneğini ayarlayın.

4.3.2 sürümünde değişti:

- Eklentilerin `post_update` yöntemine `skip_push` parametresi eklendi.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.3 sürümünden 4.4 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- `INSTALLED_APPS` seçeneğinde `weblate.configuration` şeklinde bir değişiklik yapıldı.
- Artık Django 3.1 sürümü gerekiyor.
- MySQL ya da MariaDB kullanıyorsanız, gereken en düşük sürümler yükseltildi. Bilgi almak için: *MySQL ve MariaDB*.

4.4.1 sürümünde değişti:

- *Tek dilli gettext* artık var olduğunda hem `msgid` hem de `msgctxt` kullanıyor. Bu özellik, ekran görüntüleri ya da onay durumları gibi genişletilmiş Weblate verilerine bağlantıları bozan bu tür dosyalardaki çeviri dizgilerinin kimliğini değiştirir. Lütfen yükseltmeden önce bu tür dosyalarda bekleyen değişiklikleri işlediğinizden emin olun. *loadpo* kullanarak etkilenen bileşenin yüklenmesini dayatmanız önerilir.
- Çeşitli dosya biçimi sorunlarını gidermek için gereken en düşük `translate-toolkit` sürümü yükseltildi.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.4 sürümünden 4.5 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- Büyük sözlükleriniz varsa aktarım işlemi oldukça uzun sürebilir.
- Sözlükler artık normal bileşenler olarak tutuluyor.
- Sözlük API uygulaması kaldırıldı. Sözlüklere erişmek için normal çeviri API uygulamasını kullanın.
- `INSTALLED_APPS` seçeneğinde `weblate.metrics` şeklinde bir değişiklik yapıldı.

4.5.1 sürümünde değişti:

- *Pyahocorasick* modülüne yeni bir bağımlılık eklendi.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.5 sürümünden 4.6 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- Yeni dosya biçimleri eklendi. `WEBLATE_FORMATS` seçeneğini değiştirdiyseniz bunları katmak isteyebilirsiniz.
- Bileşen oluşturan API uygulaması artık kendiliğinden *İç Weblate adresleri* kullanıyor. Bilgi almak için: `POST /api/projects/(string:project)/components/`.
- Bağımlılıklarda ve `django: 'PASSWORD_HASHERS` kalıplarında bir değişiklik yapıldı. Artık parolaların şifrelenmesi için Argon2 kullanılıyor.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.6 sürümünden 4.7 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- `settings_example.py` içinde birkaç değişiklik yapıldı. En önemlisi ara yazılım katmanı değişiklikleri (MIDDLEWARE). Lütfen ayarlarınızı buna göre değiştirin.
- *DeepL* makine çevirisine farklı abonelik modellerine daha esnek bir şekilde uyum sağlamak için genel bir `MT_DEEPL_API_URL` ayarı eklendi. `MT_DEEPL_API_VERSION` ayarı artık kullanılmıyor.
- Artık Django 3.2 sürümü gerekiyor.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.7 sürümünden 4.8 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

Bu sürüm için ek yükseltme adımlarına gerek yoktur.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.8 sürümünden 4.9 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

- Ölçümlerin depolanmasında bir değişiklik yapıldı. Büyük sitelerde yükseltme işlemi uzun sürebilir.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.9 sürümünden 4.10 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

- Proje gruplarında bir değişiklik yapıldı. Binlerce projesi bulunan sitelerde yükseltme işlemi uzun sürebilir.
- Django 4.0 ile bazı geriye dönük uyumsuzluğa yol açan değişiklikler yapıldı. Bilgi almak için: [Backwards incompatible changes in 4.0](#). Weblate, bunlardan herhangi birinin sorunlu olması durumunda şimdilik Django 3.2 sürümünü desteklemeyi sürdürüyor. Weblate platformunu etkileyebilecek en önemli değişiklikler:
 - PostgreSQL 9.6 desteği kaldırılan Django 4.0, PostgreSQL 10 ve üzerindeki sürümleri destekliyor.
 - `CSRF_TRUSTED_ORIGINS` biçimi değiştirildi.
- Docker kapsayıcısı artık Django 4.0 kullanıyor. Değişiklikler için yukarıya bakabilirsiniz.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.10 sürümünden 4.11 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

- Weblate için artık Django 3.2 gerekiyor.
- *Her proje için erişim denetimi yönetimi* uygulaması değiştirildi, proje ön eki grup adlarından kaldırıldı. Bu değişiklik, API kullanıcılarını etkiler.
- Weblate artık karakter kümesinin algılanması için `chardet` modülü yerine `charset-normalizer` kullanıyor.
- **4.11.1 sürümünde değiştirildi:** `REST_FRAMEWORK` ayarında bir değişiklik yapıldı (`DEFAULT_AUTHENTICATION_CLASSES` seçeneğindeki arka uçlardan biri kaldırıldı).

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.11 sürümünden 4.12 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

- Herhangi bir özel adım gerekmez.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.12 sürümünden 4.13 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

- *Dil tanımları* artık yükseltme sırasında kendiliğinden güncelleniyor. Bu özelliği devre dışı bırakmak için `UP-DATE_LANGUAGES` seçeneğini kullanın.
- *Windows RC dosyaları*, *HTML dosyaları*, *IDML biçimi* ve *Metin dosyaları* dosya biçimleri için bağlam ve konumun işlenmesi değiştirildi. Çoğu durumda bağlam artık konum olarak gösteriliyor.
- Makine çevirisi hizmetleri artık kullanıcı arabiriminden yapılandırılabilir. Yapılandırma dosyasındaki ayarlar veri tabanı aktarımı sırasında içe aktarılacaktır.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.13 sürümünden 4.14 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

- Java biçimlendirme denetimleri artık GNU gettext imleriyle eşleşiyor. Weblate üzerinde ayarlanan imler kendiliğinden taşınır. Ancak üçüncü taraf betiklerinin `java-format` yerine `java-printf-format` ve `java-messageformat` yerine `java-format` kullanması gerekir.
- *jellyfish* bağımlılığı *rapidfuzz* ile değiştirildi.
- **4.14.2 sürümünde değiştirildi:** VCS hizmeti API anahtarlarının güvenli olmayan yapılandırması, `_CREDENTIALS` listesi yerine `_TOKEN/_USERNAME/` yapılandırmasıyla kullanımdan kaldırıldı. Docker üzerinde lütfen eşleşen `_HOST` yönergelerini ekleyin. Örneğin `WEBLATE_GITHUB_HOST` ve `GIT-HUB_CREDENTIALS` bölümlerine bakın.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.14 sürümünden 4.15 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

- Artık Weblate için PostgreSQL üzerinde “btree_gin” eklentisi gerekiyor. Aktarım işlemi, yeterli izinleri varsa bunu kurar. El ile kurulum için *PostgreSQL üzerinde bir veri tabanı oluşturmak* bölümüne bakabilirsiniz.
- Docker kalıbı artık varsayılan olarak hata ayıklama kipini etkinleştirmiyor. İsterseniz, `WEBLATE_DEBUG` ortam seçeneği ile etkinleştirebilirsiniz.
- Büyük kopyalarda veri tabanı aktarımı, bazı dizinlerin yeniden oluşturulması nedeniyle saatlerce sürer.
- **4.15.1 sürümünde değiştirildi:** Rest framework ayarlarında `DEFAULT_PAGINATION_CLASS` seçeneğinin varsayılan değeri değiştirildi.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.15 sürümünden 4.16 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

- Celery vuruşu artık görev zamanlamasını veri tabanında saklıyor. Bunun için `CELERY_BEAT_SCHEDULER` ve `INSTALLED_APPS` seçeneklerinde değişiklik yapılmalıdır.
- Kullanımdan kaldırılan sürüm denetimi sistemi kimlik doğrulama bilgileri ayarı artık desteklenmiyor. Bilgi almak için: *4.13 sürümünden 4.14 sürümüne yükseltmek*.
- *django-crispy-forms* yükseltmesi için `INSTALLED_APPS` içinde değişiklikler yapılmalıdır.
- *django-cors-headers* bütünleştirmesi için `INSTALLED_APPS` ve `MIDDLEWARE` seçeneklerinde değişiklik yapılmalıdır.

Ayrıca bakınız:

Genel yükseltme yönergeleri

4.16 sürümünden 4.17 sürümüne yükseltmek

Güncellemek için *Genel yükseltme yönergeleri* adımlarını izleyin.

- Yeniden yazılmış ölçüm depolama alanına aktarım, büyük Weblate kopyalarında önemli ölçüde zaman alabilir (`metrics_metric` tablosu için GB başına yaklaşık 15 dakika bekleyebilirsiniz). Kesinti süresini azaltmak için `weblate/metrics/migrations/*.py` dosyasını Weblate 4.17 sürümünden 4.16 sürümüne kopyalayarak aktarımı arka planda başlatabilirsiniz. İşlem tamamlandığında, normal tam yükseltme işlemi yapın.

Ayrıca bakınız:

Genel yükseltme yönergeleri

2.3.4 Python 2 sürümünden Python 3 sürümüne yükseltmek

Weblate artık Python için 3.6 sürümünden öncesini desteklemiyor. Hala eski sürümde çalışıyorsanız, lütfen önce geçerli sürümden Python 3 sürümüne aktarım yapın ve ardından yükseltin. [Weblate 3.11.1 belgelerinde Python 2 sürümünden Python 3 sürümüne yükseltme](#) bölümüne bakabilirsiniz.

2.3.5 Diğer veri tabanlarından PostgreSQL üzerine aktarmak

Weblate için PostgreSQL dışında başka bir veri tabanı kullanıyorsanız, Weblate ile en iyi başarıyı gösteren PostgreSQL kullanmaya geçmeyi düşünmelisiniz. Aşağıdaki adımlar, verilerinizi veri tabanları arasında aktarmakta size yol gösterir. Lütfen aktarımdan önce hem site hem de Celery sunucularını durdurmayı unutmayın, yoksa tutarsız verilerle karşılaşabilirsiniz.

PostgreSQL üzerinde bir veri tabanı oluşturmak

Weblate için ayrı bir kullanıcı hesabı ile ayrı bir veri tabanı kullanmak genellikle iyi bir fikirdir:

```
# 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
```

Django JSON dökümlerini kullanarak aktarmak

Aktarım için en basit yaklaşım, Django JSON dökümlerini kullanmaktır. Bu yöntem, küçük kurulumlar için iyidir. Daha büyük kurulumlarda bunun yerine pgloader kullanmak isteyebilirsiniz. Bilgi almak için: [Pgloader ile PostgreSQL üzerine aktarmak](#).

1. PostgreSQL seçeneğini `settings.py` dosyasına ek veritabanı bağlantısı olarak ekleyin:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# 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. Aktarımları yürütün ve tablolara eklenen tüm verileri kaldırın:

```

weblate migrate --database=postgresql
weblate sqlflush --database=postgresql | weblate dbshell --database=postgresql

```

3. Eski veri tabanlarının dökümünü alın ve PostgreSQL üzerine aktarın

```

weblate dumpdata --all --output weblate.json
weblate loaddata weblate.json --database=postgresql

```

4. Varsayılan olarak yalnızca PostgreSQL veri tabanının kullanılması için DATABASES ayarını yapın. Eski bağlantıyı kaldırın.

Böylece Weblate artık PostgreSQL veri tabanı ile çalışmaya hazır olmalıdır.

Pgloader ile PostgreSQL üzerine aktarmak

pgloader, verileri PostgreSQL üzerine taşımak için kullanılan bir genel aktarım aracıdır. Weblate veri tabanını aktarmak için kullanabilirsiniz.

1. Veri tabanı olarak PostgreSQL kullanmak için `settings.py` ayarınızı yapın.
2. Şemayı PostgreSQL veri tabanına aktarın:

```

weblate migrate
weblate sqlflush | weblate dbshell

```

- Verileri aktarmak için pgloader uygulamasını çalıştırın. Veri tabanını aktarmak için aşağıdaki betik kullanılabilir, ancak ne yaptığını anlamak ve kurulumunuza uyacak şekilde küçük ayar değişiklikleri yapmak için [pgloader](#) ile ilgili ayrıntılı bilgi almak isteyebilirsiniz:

```
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 Pootle üzerinden aktarmak

Weblate başlangıçta Pootle yerine yazıldığından, kullanıcı hesapları Pootle üzerinden aktarılabilir. Kullanıcıları Pootle üzerinden indirip ve `importusers` ile içe aktarabilirsiniz.

2.4 Weblate yedeğini alma ve taşıma

2.4.1 Proje düzeyinde yedekler

4.14 sürümünde geldi.

Uyarı: Yedeklemeler yalnızca veri tabanı olarak PostgreSQL ya da MariaDB 10.5+ kullanıldığında geri yüklenebilir.

Proje, Weblate üzerindeki tüm çeviri içeriğini yedekler (proje, bileşenler, çeviriler, dizge yorumları, öneriler ve denetimler). Bir projeyi başka bir Weblate kopyasına aktarmak için uygundur.

Bir projeyi *Yönetim* ↓ *Yedekler* bölümünden yedekleyebilirsiniz. Bir proje oluşturulurken yedekleme geri yüklenebilir (bilgi almak için: *Çeviri projelerini ve bileşenleri eklemek*).

Şu anda yedeklerde erişim denetimi ve geçmiş bilgileri bulunmuyor.

Yorumlar ve öneriler, bunları oluşturan kullanıcının kullanıcı adıyla yedeklenir. İçe aktarıldıktan sonra eşleşen bir kullanıcı ile ilişkilendirilir. Yedekteki kullanıcı adına sahip bir kullanıcı bulunamazsa, anonim kullanıcı ile ilişkilendirilir.

Oluşturulan yedekler, `PROJECT_BACKUP_KEEP_DAYS` ve `PROJECT_BACKUP_KEEP_COUNT` tarafından yapılandırıldığı şekilde sunucuda tutulur (varsayılan olarak 30 gün boyunca en fazla 3 yedek tutulur).

2.4.2 BorgBackup ile kendiliğinden yedekleme

3.9 sürümünde geldi.

Weblate, [BorgBackup](#) kullanarak hizmet yedeklerinin alınmasını destekler. Borg, bulutta güvenli bir şekilde saklanabilen, depolama alanı açısından etkili şifrelenmiş yedekler alır. Yedekler, yönetim arayüzünde *Yedekler* sekmesinden yönetilebilir.

4.4.1 sürümünde değişti: Kendiliğinden alınan yedeklere hem PostgreSQL hem de MySQL/MariaDB veri tabanları katılır.

Borg kullanan yedekler artımlıdır ve Weblate şu yedekleri koruyacak şekilde yapılandırılır:

- 14 gün geriye dönük günlük yedekler

- 8 hafta geriye dönük haftalık yedekler
- 6 ay geriye dönük aylık yedekler

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/tmpt32bhxkuweblate

Backup service credentials
April 17, 2023

Backup repository
/tmp/tmpt32bhxkuweblate

Passphrase
4W*BEnGO (Up99eNT8BfGmMF4u0%PBpbd^)1vwGxV@VEbP*NfJD
The passphrase is used to encrypt the backups and is necessary to restore them.

SSH key
Download private key
The private key is needed to access the remote backup repository.

Deleted the oldest backups
April 17, 2023

Backup performed
April 17, 2023

Repository initialization
April 17, 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 4.17
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Borg şifreleme anahtarı

BorgBackup şifrelenmiş yedekler alır ve parola olmadan bunları geri yükleyemezsiniz. Parola, yeni bir yedekleme hizmeti eklerken oluşturulur ve bunu kopyalayıp güvenli bir yerde saklamanız gerekir.

Weblate tarafından sunulan yedekleme depolama alanı kullanıyorsanız, yedeklerinize erişmek için kullanıldığından, lütfen kişisel SSH anahtarınızı da yedekleyin.

Ayrıca bakınız:

`borg init`

Yedeklemeyi özelleştirme

- Veri tabanı yedeklemesi `DATABASE_BACKUP` ile yapılandırılabilir.
- Yedek oluşturma `BORG_EXTRA_ARGS` kullanılarak özelleştirilebilir.

2.4.3 Weblate tarafından sunulan yedekleme depolama alanı

Weblate bulut sunucunuzu yedeklemenin en kolay yolu **`weblate.org <https://weblate.org/support/#backup> üzerinden yedekleme hizmeti`** satın almaktır. Şu şekilde çalışır duruma getirebilirsiniz:

1. <https://weblate.org/support/#backup> üzerinden *Yedekleme hizmeti* satın alın.
2. Size verilen anahtarı yönetim bölümüne yazın. Bilgi almak için: *Destek bütünleşmesi*.
3. Weblate bulut hizmeti ile bağlantı kurar ve yedekler için erişim bilgilerini alır.
4. *Yedekler* sekmesinden yeni yedekleme yapılandırmasını etkinleştirin.
5. Yedekleri geri yükleyebilmek için Borg kimlik doğrulama bilgilerinizi yedekleyin. Bilgi almak için: *Borg şifreleme anahtarı*.

İpucu: Güvenliğiniz için el ile her şeyi etkinleştirme adımı bulunur. İzniniz olmadan, hesap açma işlemi yoluyla elde edilen yedekleme havuzuna herhangi bir veri gönderilmez.

2.4.4 Özel yedekleme depolama alanını kullanmak

Yedekler için kendi depolama alanınızı da kullanabilirsiniz. SSH, yedekleri uzak hedefte depolamak için kullanılabilir. Hedef sunucuda *BorgBackup* kurulu olmalıdır.

Ayrıca bakınız:

Borg belgelerindeki *General* bölümünde

Yerel dosya sistemi

Yerel yedekleme için mutlak yolun belirtilmesi önerilir. Örneğin `/yedek/klasoru/yolu`. Klasör, Weblate çalıştıran kullanıcı tarafından yazılabilir olmalıdır (bilgi almak için: *Dosya sistemi izinleri*). Klasör yoksa, Weblate oluşturmaya çalışır, ancak bunu yapmak için uygun izinlere gerek duyar.

İpucu: Docker üzerinde Weblate çalıştırırken, lütfen yedekleme konumunun Weblate kapsayıcısında bir birim olarak gösterildiğinden emin olun. Yoksa, içinde bulunduğu kapsayıcı yeniden başlatıldığında yedekler Docker tarafından atılır.

Yedekleri `/app/data/borgbackup` gibi var olan bir birime yerleştirmek bir seçenek olabilir. Bu birim kapsayıcıda vardır.

Docker Compose dosyasındaki yedekler için örneğin `/borgbackup` kullanarak yeni bir kapsayıcı da ekleyebilirsiniz:

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

Yedeklerin depolanacağı klasör için UID 1000 olmalıdır. Yoksa Weblate yedekleri bu klasöre yazamaz.

Uzak yedeklemeler

Uzak yedekler almak için, Weblate SSH anahtarını kullanarak SSH üzerinden Weblate dağıtımınız için erişilebilen başka bir sunucuya **BorgBackup** kurmanız gerekir:

1. Yedeklerinizin depolanacağı bir sunucu hazırlayın.
2. SSH sunucusunu üzerine kurun (çoğu Linux dağıtımında varsayılan olarak bulunur).
3. Bu sunucuya **BorgBackup** kurun. Çoğu Linux dağıtımında kullanılacak paketleri bulunur (bilgi almak için: [Installation](#)).
4. Var olan bir kullanıcı seçin ya da yedekleme için kullanılacak yeni bir kullanıcı ekleyin.
5. Weblate tarafından parola olmadan sunucu ile SSH bağlantısının kurulabilmesi için kullanıcıya Weblate SSH anahtarını ekleyin (bilgi almak için: [Weblate SSH anahtarı](#)).
6. Yerel yedeklemeler için `kullanici@sunucu:/yedeklere/giden/yol` şeklinde ya da uzak SSH yedeklemeleri için `ssh://kullanici@sunucu:baglantinoktasi/yedeklere/giden/yol` şeklinde kullanın.

İpucu: *Weblate tarafından sunulan yedekleme depolama alanı* herhangi bir çaba harcamadan kendiliğinden uzak yedekleme olanağı sağlar.

Ayrıca bakınız:

[Weblate SSH anahtarı](#), [General](#)

2.4.5 BorgBackup yedeklerini geri yüklemek

1. Yedekleme deponuza erişimi geri yükleyin ve yedekleme parolanızı hazırlayın.
2. Sunucudaki tüm yedekleri `borg list REPOSITORY` komutunu kullanarak listeleyin.
3. İstedığınız yedeği geçerli klasöre geri yüklemek için `borg extract REPOSITORY::ARCHIVE` komutunu kullanın.
4. Veri tabanını, Weblate data klasörü içindeki backup klasörüne kaydedilen SQL dökümünden geri yükleyin (bilgi almak için: [Yedeklemeler için dökümü yapılan veriler](#)).
5. Weblate yapılandırmasını (`backups/settings.py`, bilgi almak için: [Yedeklemeler için dökümü yapılan veriler](#)) doğru konuma kopyalayın. Bilgi almak için: [Yapılandırmayı ayarlama](#).

Docker kapsayıcısını kullanırken, ayarlar dosyası zaten kapsayıcı içine eklenmiştir ve özgün ortam değişkenlerini geri yüklemeniz gerekir. `environment.yml` dosyası bu konuda size yardımcı olabilir (bilgi almak için: [Yedeklemeler için dökümü yapılan veriler](#)).

6. Geri yüklenen tüm data klasörünü `DATA_DIR` ile yapılandırılmış konuma kopyalayın.

Docker kapsayıcısını kullanırken verileri veri birimine yerleştirin. Bilgi almak için: [Docker kapsayıcısı birimleri](#).

Lütfen dosyaların sahiplik ve erişim izinlerinin doğru olduğundan emin olun. Bilgi almak için: [Dosya sistemi izinleri](#).

Borg oturumu şunun gibi görünmelidir:

```
$ 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:
```

Ayrıca bakınız:

[borg list](#), [borg extract](#)

2.4.6 El ile yedekleme

Neyi kaydetmek istediğinize bağlı olarak, Weblate tarafından her biri ilgili yere kaydedilmiş olan veri türünü yedekleyin.

İpucu: El ile yedekleme yapıyorsanız, Weblate yedekleme eksikliği uyarısını, `settings.py` dosyasındaki `SILENCED_SYSTEM_CHECKS` ya da Docker için `WEBLATE_SILENCED_SYSTEM_CHECKS` seçeneğini `weblate.I028` ile ekleyerek kapatmak isteyebilirsiniz.

```
SILENCED_SYSTEM_CHECKS.append("weblate.I028")
```

Veri tabanı

Gerçek depolama konumu, veri tabanı kurulumunuza bağlıdır.

İpucu: Veri tabanı en önemli depolama alanıdır. Veri tabanınızı düzenli olarak yedeklenecek biçimde ayarlayın. Veri tabanı olmadan, tüm çeviriler kaybolur.

Doğal veri tabanı yedeklemesi

Veri tabanı dökümününün `pg_dump` ya da `mysqldump` gibi veri tabanına özgü araçlar kullanılarak kaydedilmesi önerilir. Başarımı genellikle Django yedeklemesinden daha iyidir ve tüm tablolar tüm verileriyle geri yüklenir.

Bu yedeği daha yeni bir Weblate sürümünde geri yükleyebilirsiniz, `migrate` içinde çalışırken gerekli tüm aktarımlar yapılır. Sürümler arasında nasıl yükseltme yapılacağı ile ilgili bilgi almak için [Weblate yazılımını güncellemek](#) bölümüne bakabilirsiniz.

Django veri tabanı yedeklemesi

Alternatif olarak, `dumpdata` Django komutunu kullanarak veri tabanınızı yedekleyebilirsiniz. Bu şekilde, alınan yedek veri tabanından bağımsız olur ve veri tabanı işleyicisini değiştirmek istediğinizde kullanılabilir.

Veri tabanını geri yüklemekten önce, yedeklemenin yapıldığı Weblate sürümüyle tam olarak aynı sürümü çalıştırıyor olmanız gerekir. Veri tabanı yapısı sürümler arasında değiştiğinden ve verilerin bir şekilde bozulmasına neden olduğundan böyle olması gereklidir. Aynı sürümü kurduktan sonra, tüm veri tabanı aktarımlarını yapmak için `migrate` komutunu kullanın.

Daha sonra veri tabanında bazı kayıtlar zaten oluşturulmuş olacak ve bunlar veri tabanı yedeğinde de bulunuyor olacak. Önerilen yaklaşım, yönetim kabuğunu kullanarak bu tür kayıtları el ile silmektir (bilgi almak için [Yönetim komutlarını çağırmak](#)):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

Dosyalar

Yeterli yedekleme alanınız varsa, `DATA_DIR` klasörünü tümüyle yedekleyin. Bu yedek, istemediğiniz bazı dosyaları içerse bile en güvenlisidir. Aşağıdaki bölümlerde, neleri yedeklemeniz gerektiğini ve neleri atlayabileceğinizi ayrıntılı olarak görebilirsiniz.

Yedeklemeler için dökümü yapılan veriler

4.7 sürümünde değişti: Docker ortamlarında geri yüklemeye yardımcı olacak ortam dökümü `environment.yml` olarak eklenmiştir.

`DATA_DIR/backups` içine kaydedilir.

Weblate buraya çeşitli verileri döker. Daha eksiksiz yedeklemeler için bu dosyaları ekleyebilirsiniz. Dosyalar günlük olarak güncellenir (çalışan bir Çeler beats sunucusu gerekir. Bilgi almak için: [Celery ile arka plan görevlerini kullanmak](#)). Şu anda, bulunan veriler:

- Weblate ayarları için `settings.py` dosyası (`settings-expanded.py` dosyasında genişletilmiş bir sürümü de bulunur).
- PostgreSQL veri tabanı yedeği için `database.sql`.
- Ortam dökümü için `environment.yml`.

Veri tabanı yedekleri varsayılan olarak düz metin biçiminde kaydedilir. Ancak `DATABASE_BACKUP` seçeneği ile sıkıştırılabilir ya da tümüyle atlanabilir.

Veri tabanı yedeğini geri yüklemek için veri tabanı araçlarını kullanın. Örneğin:

```
psql --file=database.sql weblate
```

Sürüm denetimi depoları

`DATA_DIR/vcs` konumunda tutulur.

Sürüm denetimi depolarında, Weblate değişiklikleriyle birlikte yukarı akış depolarınızın bir kopyası bulunur. Tüm çeviri bileşenleriniz için *İşleme ile itme* seçeneği etkinse, tüm Weblate değişiklikleri yukarı akışa dahil edilir. Weblate tarafındaki depoları yedeklemek gerekmez çünkü veri kaybı olmadan yukarı akış konumlarından yeniden kopyalanabilir.

SSH ve GPG anahtarları

`DATA_DIR/ssh` ve `DATA_DIR/home` içinde bulunur.

Weblate tarafından oluşturulmuş SSH ya da GPG anahtarlarını kullanıyorsanız, bu konumları yedeklemeniz gerekir. Yoksa kişisel anahtarlarınızı kaybedersiniz ve yeniden oluşturmanız gerekir.

Kullanıcının yüklediği dosyalar

`DATA_DIR` /media konumunda tutulur.

Kullanıcı tarafından yüklenen tüm dosyaları yedeklemelisiniz (*Dizgeler için görsel bağlam* gibi).

Celery görevleri

Celery görev kuyruğunda bazı bilgiler bulunabilir. Ancak genellikle yedeklenmesi gerekli değildir. En fazla, henüz çeviri belleğine işlenmemiş güncellemeleri kaybedersiniz. Gene de, tam metin ya da depo güncellemesinin geri yük-
lendikten sonra yapılması önerilir, bu nedenle bunların kaybedilmesi bir sorun oluşturmaz.

Ayrıca bakınız:

Celery ile arka plan görevlerini kullanmak

Komut satırından el ile yedek alma

Bir zamanlanmış görev (cron) kullanarak, her gün yürütülecek bir Bash komutu ayarlayabilirsiniz. Örneğin:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz \
↳ backups vcs ssh home media fonts secret
```

`XZ_OPT` işleminden sonra tırnak işaretleri arasındaki dizge, xz seçeneklerinizi, örneğin sıkıştırma için kullanılan bellek miktarını seçmenizi sağlar. Bilgi almak için: <https://linux.die.net/man/1/xz>

Klasör ve dosya listesini gereksinimlerinize göre ayarlayabilirsiniz. Çeviri belleğini (yedeklemeler klasöründe) kay-
detmekten kaçınmak için şunu kullanabilirsiniz:

```
$ 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 Yedeği el ile geri yüklemek

1. Yedeklediğiniz tüm verileri geri yükleyin.
2. `:d jadmin:updategit` komutunu kullanarak tüm depoları güncelleyin.

```
weblate updategit --all
```

2.4.8 Bir Weblate kurulumunu taşımak

Yukarıdaki yedekleme ve geri yükleme yönergelerini izleyerek kurulumunuzu farklı bir sisteme taşıyın.

Ayrıca bakınız:

Python 2 sürümünden Python 3 sürümüne yükseltmek, Diğer veri tabanlarından PostgreSQL üzerine aktarmak

2.5 Kimlik doğrulama

2.5.1 Kullanıcı hesabı açma

Weblate varsayılan kurulumda, yeni açılan kullanıcıların hesaplarını işlemek için sitesindeki bir form olan python-social-auth kullanır. Yeni bir kullanıcı, e-posta adresini onayladıktan sonra, üçüncü taraf hizmetlerden birini kullanarak katkıda bulunabilir veya kimliğini doğrulayabilir.

Yeni kullanıcı hesaplarının açılmasını `REGISTRATION_OPEN` komutu ile de kapatabilirsiniz.

Kimlik doğrulama girişimleri *Hızı sınırlamak* değeriyle sınırlanır.

2.5.2 Kimlik doğrulama arka plan uygulamaları

Hazır Django çözümü, çeşitli sosyal ağ seçenekleri ile birlikte kimlik doğrulama için kullanılır. Bu özelliği kullanmak, diğer Django temelli projelerin kullanıcı veri tabanını içe aktarabilmenizi sağlar (bilgi almak için: [Pootle üzerinden aktarmak](#)).

Django başka yollarla da kimlik doğrulaması yapacak şekilde ayarlanabilir.

Ayrıca bakınız:

Kimlik doğrulama ayarları, belgesinde resmi Docker kalıbında kimlik doğrulamasının nasıl yapılandırılacağı anlatılmıştır.

2.5.3 Sosyal ağ kimlik doğrulaması

Weblate [Welcome to Python Social Auth's documentation!](#) kullanarak, GitLab, Ubuntu, Fedora gibi birçok üçüncü taraf hizmeti ile kimlik doğrulamayı destekler.

Django Framework belgesinden genel yapılandırma yönergelerine bakabilirsiniz.

Not: Weblate varsayılan olarak, doğrulanmış bir e-posta adresi sağlamak için üçüncü taraf kimlik doğrulama hizmetlerine güvenir. Kullanmak istediğiniz hizmetlerden bazıları bunu desteklemiyorsa, lütfen bunlar için `FORCE_EMAIL_VALIDATION` yapılandırması ile Weblate tarafında e-posta doğrulamasını zorunlu kılın. Örneğin:

```
SOCIAL_AUTH_OPENSEUSE_FORCE_EMAIL_VALIDATION = True
```

Ayrıca bakınız:

Pipeline

Tek tek arka uçları etkinleştirmek oldukça kolaydır. Bunun için yalnızca `AUTHENTICATION_BACKENDS` seçeneği ile bir kayıt ve olabilecek belirli bir kimlik doğrulama yöntemi için gerekli anahtarları eklemek gerekir. Bazı arka uçların varsayılan olarak kullanıcı e-posta adresi sağlamadığını, açıkça istemeniz gerektiğini, yoksa kullanıcılarının yaptığı katkıların hesabının Weblate tarafından düzgün bir şekilde tutulamayacağını unutmayın.

İpucu: Kimlik doğrulama arka uçlarının çoğu için HTTPS gerekir. Web sunucunuzda HTTPS özelliğini etkinleştirdikten sonra lütfen Weblate için `ENABLE_HTTPS` ya da Docker kapsayıcısı için `WEBLATE_ENABLE_HTTPS` seçeneklerini yapılandırarak düzgün bir şekilde bildirilmesini sağlayın.

Ayrıca bakınız:

Python Social Auth arka yüzü

OpenID kimlik doğrulaması

OpenID kullanan hizmetler için genellikle yalnızca onları etkinleştirmek gerekir. Aşağıdaki bölüm OpenSUSE, Fedora ve Ubuntu için OpenID kimlik doğrulamasını etkinleştirir:

```
# 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",
)
```

Ayrıca bakınız:

[OpenID](#)

GitHub kimlik doğrulaması

GitHub üzerinde bir OAuth uygulaması oluşturduktan sonra tüm erişim bilgilerini Weblate üzerine yazmanız gerekir:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.github.GithubOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = "GitHub Client ID"
SOCIAL_AUTH_GITHUB_SECRET = "GitHub Client Secret"
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]
```

GitHub, geri çağırma adresi “<https://site.com/accounts/complete/github/>” olacak biçimde yapılandırılmalıdır.

Kuruluşlar için GitHub ve Takımlar için GitHub benzer kimlik doğrulama arka yüzleri vardır. Ayarları `SOCIAL_AUTH_GITHUB_ORG_*` ve `SOCIAL_AUTH_GITHUB_TEAM_*` olarak adlandırılmıştır ve `SOCIAL_AUTH_GITHUB_ORG_NAME` ya da `SOCIAL_AUTH_GITHUB_TEAM_ID` kapsamları için ek ayarlara gerek duyarlar. Geri çağırma adresleri <https://site.com/accounts/complete/github-org/> ve <https://site.com/accounts/complete/github-teams/> biçimindedir.

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

[GitHub](#)

Bitbucket kimlik doğrulaması

Bitbucket üzerinde bir OAuth uygulaması oluşturduktan sonra tüm erişim bilgilerini Weblate üzerine yazmanız gerekir:

```
# 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
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

Bitbucket

Google OAuth 2

Google OAuth 2 kullanmak için <<https://console.developers.google.com/>> adresinden bir uygulama kaydedip Google+ API uygulamasını etkinleştirmeniz gerekir.

Yönlendirme adresi: `https://WEBLATE_SERVER/accounts/complete/google-oauth2/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = "Client ID"
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = "Client secret"
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

Google

Facebook OAuth 2

OAuth 2 hizmetlerinde her zamanki gibi, uygulamanızı Facebook üzerine kaydetmeniz gerekir. Bunu yaptıktan sonra, Weblate yapılandırmasını uygulamayı kullanacak biçimde ayarlayabilirsiniz:

Yönlendirme adresi: `https://WEBLATE_SERVER/accounts/complete/facebook/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.facebook.FacebookOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_FACEBOOK_KEY = "key"
SOCIAL_AUTH_FACEBOOK_SECRET = "secret"
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

Facebook

GitLab OAuth 2

GitLab OAuth 2 kullanmak için <https://gitlab.com/profile/applications> adresinden bir uygulama kaydetmeniz gerekir.

Yönlendirme adresi: `https://WEBLATE_SERVER/accounts/complete/gitlab/` and ensure you mark the `read_user` scope.

```
# 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/'
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

GitLab

Microsoft Azure Active Directory

Weblate, kimlik doğrulaması için genel ya da belirli kiracıları kullanacak şekilde yapılandırılabilir.

Yönlendirme adresi: Genel kiracılar için `https://WEBLATE SERVER/accounts/complete/azuread-oauth2/` ve belirli kiracılar için `https://WEBLATE SERVER/accounts/complete/azuread-tenant-oauth2/` biçimindedir.

```
# 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 = ""
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

Microsoft Azure Active Directory

Slack

Slack OAuth 2 kullanmak için `<https://api.slack.com/apps>` adresinden bir uygulama kaydetmeniz gerekir.

Yönlendirme adresi: `https://WEBLATE SERVER/accounts/complete/slack/`.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.slack.SlackOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SLACK_KEY = ""
SOCIAL_AUTH_SLACK_SECRET = ""
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

[Slack](#)

Kimlik doğrulama yöntemi adlarını ve simgelerini değiştirmek

Kimlik doğrulama yönteminin görüntülenen adını ve simgesini `SOCIAL_AUTH_<NAME>_IMAGE` ve `SOCIAL_AUTH_<NAME>_TITLE` ayarları ile değiştirebilirsiniz. Örneğin Auth0 adı değiştirildiğinde şöyle görüntülenir:

```
SOCIAL_AUTH_AUTH0_IMAGE = "custom.svg"
SOCIAL_AUTH_AUTH0_TITLE = "Custom auth"
```

Parola ile kimlik doğrulamayı kapatmak

E-posta ve parola kimlik doğrulaması `AUTHENTICATION_BACKENDS` bölümündeki `social_core.backends.email.EmailAuth` seçeneği kaldırılarak kapatılabilir. Temel Weblate işlevleri için gerekli olduğundan `weblate.accounts.auth.WeblateUserBackend` seçeneğini her zaman orada tutun.

E-posta kimlik doğrulamasını devre dışı bırakmak, e-postay ile ilgili tüm işlevleri (kullanıcı daveti ya da parola sıfırlama özelliği) devre dışı bırakır.

Tüyo: Yönetici arayüzü için, orada el ile oluşturduğunuz kullanıcılarda parola ile kimlik doğrulamasını kullanmayı sürdürebilirsiniz. ```/admin/login/``` adresine gitmeniz yeterlidir.

Örneğin, yalnızca openSUSE Open ID hizmet sağlayıcısını kullanan kimlik doğrulaması şunu kullanarak yapılabilir:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

2.5.4 Parola ile kimlik doğrulama

Varsayılan `settings.py` dosyasında mantıklı bir `AUTH_PASSWORD_VALIDATORS` kümesi bulunur:

- Parola, kişisel bilgilerinize çok benzer bir şey olamaz.
- Parola en az 10 karakter uzunluğunda olmalıdır.
- Parola yaygın kullanılan bir şey olamaz.
- Parola tümüyle rakamlardan oluşamaz.
- Parola tek bir karakterden ya da boşluk karakterinden oluşamaz.
- Parola daha önce kullandığınız bir parola olamaz.

Bu seçeneği parola ilkenize uygun biçimde değiştirebilirsiniz.

Ek olarak, parola zorluğu ile ilgili oldukça gerçekçi öngörüler veren ve belirli bir eşğin altında kalan parolaların reddedilmesini sağlayan `django-zxcvbn-password` dosyasını da yükleyebilirsiniz.

2.5.5 SAML kimlik doğrulaması

4.1.1 sürümünde geldi.

Lütfen yapılandırma için Python Social Auth yönergelerini izleyin. Önemli farklar:

- Weblate, ``SOCIAL_AUTH_SAML_ENABLED_IDPS`` içinde `weblate` olarak adlandırılması gereken tek kimlik hizmeti sağlayıcısını destekler.
- SAML XML üst verileri adresi: `/accounts/metadata/saml/`.
- Şu seçenekler kendiliğinden doldurulur: `SOCIAL_AUTH_SAML_SP_ENTITY_ID`, `SOCIAL_AUTH_SAML_TECHNICAL_CONTACT`, `SOCIAL_AUTH_SAML_SUPPORT_CONTACT`

Örnek yapılandırma:

```
# 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"
}
```

Varsayılan yapılandırma, şu özniteliklerden kullanıcı bilgilerini ayıklar, kimlik hizmeti sağlayıcınızı bu öznitelikleri sunacak şekilde yapılandırın:

Öznitelik	SAML URI referansı
Ad soyad	urn:oid:2.5.4.3
Ad	urn:oid:2.5.4.42
Soyad	urn:oid:2.5.4.4
E-posta	urn:oid:0.9.2342.19200300.100.1.3
Kullanıcı adı	urn:oid:0.9.2342.19200300.100.1.1

İpucu: Yukarıdaki örnek ve Docker görüntüsü `weblate` adlı bir kimlik hizmeti sağlayıcısı tanımlamaktadır. Bu dizgeyi kimlik hizmeti sağlayıcınızda *Relay* olarak yapılandırmanız gerekebilir.

Ayrıca bakınız:

Docker içinde SAML yapılandırması, SAML

2.5.6 LDAP kimlik doğrulaması

LDAP kimlik doğrulaması en iyi şekilde *django-auth-ldap* paketi kullanılarak sağlanabilir. Her zamanki gibi kurabilirsiniz:

```
# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap
```

İpucu: Bu paket Docker kapsayıcısı ile birlikte gelir. Bilgi almak için: *Docker ile kurmak*.

Not: Python LDAP 3.1.0 modülünde, bu sürümü kullanmanızı engelleyebilecek bazı uyumsuzluklar vardır. *AttributeError: 'module' nesnesinin '_trace_level' özniteliği yok* hatasıyla karşılaşarsanız, *python-ldap* modülünü 3.0.0 sürümüne düşürmek yardımcı olabilir.

Paketi kurduktan sonra, Django kimlik doğrulaması ile bağlayabilirsiniz:

```
# 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
```

Not: `:setting:django:AUTHENTICATION_BACKENDS` ayarından `social_core.backends.email.EmailAuth` seçeneğini kaldırmalısınız. Yoksa kullanıcılar Weblate üzerinde kendi parolalarını ayarlayabilir ve bunu kullanarak kimlik doğrulaması yapabilir. İzinler almak ve anonim kullanıcılara kolaylık sağlamak için `weblate.accounts.auth.WeblateUserBackend` seçeneğinin tutulması hala gereklidir. Ayrıca, yerel bir yönetici hesabı oluşturduysanız (örneğin `:d jadmin:createadmin` kullanarak) oturum açmanızı da sağlar.

BIND parolasını kullanarak

Kimlik doğrulaması için doğrudan BIND kullanamıyorsanız, aramayı kullanmanız ve arama için bağlanacak bir kullanıcı belirtmeniz gerekir. Örneğin:

```
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)"
)
```

Active Directory bütünleştirilmesi

```
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
```

Ayrıca bakınız:

Django Authentication Using LDAP, Authentication

2.5.7 CAS kimlik doğrulaması

CAS kimlik doğrulaması, *django-cas-ng* gibi bir paket kullanılarak yapılabilir.

Birinci adım, kullanıcının e-posta alanını CAS aracılığıyla açıklamaktır. Bunun CAS sunucusunun kendisinde yapılandırılması gerekir ve CAS v1 öznelikleri tarafından desteklenmediğinden en az CAS v2 çalıştırmanız gerekir.

İkinci adım, Weblate yapılandırmasını CAS sunucunuzu ve özneliklerinizi kullanacak şekilde güncellemektir.

django-cas-ng kurmak için:

```
pip install django-cas-ng
```

Paketi kurduktan sonra, `settings.py` dosyasını değiştirerek Django kimlik doğrulama sistemiyle bağlantı kurabilirsiniz:

```
# 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")
```

Son olarak, e-posta alanını kullanıcı nesnesiyle eşlemek için bir sinyal kullanabilirsiniz. Bunun çalışması için sinyali *django-cas-ng* paketinden içe aktarmanız ve kodunuzu bu sinyale bağlamanız gerekir. Bunu ayarlar dosyasında yapmak sorunlara neden olabilir. Bu nedenle şunları koymanız önerilir:

- Uygulamanızın yapılandırmasında `:p y:meth: django:django.apps.AppConfig.ready` yöntemi
- Projenin `urls.py` dosyasında (herhangi bir model var olmadığında)

```
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()
```

Ayrıca bakınız:

Django CAS NG

2.5.8 Üçüncü taraf Django kimlik doğrulamasını yapılandırmak

Genellikle herhangi bir Django kimlik doğrulama uygulama eki Weblate ile çalışmalıdır. Yalnızca eklentinin yönergelerini izleyin. Weblate kullanıcı arka ucunu kurulmuş olarak tutmayı unutmayın.

Ayrıca bakınız:

LDAP kimlik doğrulaması, CAS kimlik doğrulaması

Genellikle kurulum, `AUTHENTICATION_BACKENDS` seçeneğine bir kimlik doğrulama arka ucu eklemekten ve `INSTALLED_APPS` içine bir kimlik doğrulama uygulaması (varsa) kurmaktan oluşur:

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (
    # Install authentication app here
)
```

2.6 Erişim denetimi

Weblate, kullanıcı izinlerinin uygulamanın tamamı ya da bir bölümü için atanabileceği ayrıntılı bir yetki sistemi sunar.

3.0 sürümünde değişti: Weblate 3.0 sürümünden önce, yetki sistemi yalnızca Django izin sistemine dayanıyordu. Ancak şimdi Weblate uygulamasına özel olarak oluşturuldu. Daha eski bir sürüm kullanıyorsanız, lütfen kullandığınız sürümün belgelerine bakın.

2.6.1 Basit erişim denetimi

Tüm Weblate kurulumunu yönetmiyorsanız ve yalnızca belirli projelere erişiminiz varsa ([Hosted Weblate](#) gibi), erişim denetimi yönetimi ayarlarınız aşağıdaki seçeneklerle kısıtlıdır. Daha karmaşık bir kurulum gerektirmiyorsanız, bunlar sizin için yeterlidir.

Proje erişim denetimi

Not: Hosted Weblate üzerinde ücretsiz Libre tarifesi kullanan projeler her zaman *Public* şeklindedir. Projenize erişimi kısıtlamak istiyorsanız ücretli tarife geçebilirsiniz.

Farklı bir *Erişim denetimi* ayarı seçerek kullanıcının bireysel projelere erişimini sınırlayabilirsiniz. Kullanılabilecek seçenekler şunlardır:

Herkese açık

Herkes tarafından görülebilir.

Kimliği doğrulanmış tüm kullanıcılar tarafından katkıda bulunulabilir.

Sürüm denetimi sistemi herkese açılabilir.

Açık kaynaklı projeler için ya da Weblate kopyanız özel ya da kilitli olduğunda bunu seçin.

Korunmuş

Herkes tarafından görülebilir.

Yalnızca seçilmiş kullanıcılar tarafından katkıda bulunulabilir.

Yalnızca seçilmiş kullanıcılar sürüm denetimi sistemi deposuna erişebilir.

Görünürlük kazanmak için bunu seçin. Ancak yine de kimlerin katkıda bulunabileceğini denetleyin.

Kişisel

Yalnızca seçilmiş kullanıcılar tarafından görülebilir.

Yalnızca seçilmiş kullanıcılar tarafından katkıda bulunulabilir.

Yalnızca seçilmiş kullanıcılar sürüm denetimi sistemi deposuna erişebilir.

Herkese açık olmaması gereken projeler için bunu seçin.

Özel

Yalnızca seçilmiş kullanıcılar tarafından görülebilir.

Yalnızca seçilmiş kullanıcılar tarafından katkıda bulunulabilir.

Yalnızca seçilmiş kullanıcılar sürüm denetimi sistemi deposuna erişebilir.

Hosted Weblate üzerinde kullanılamaz.

Tüm izinleri *Özel erişim denetimi* ile ayarlamanız gerekir.

Erişimi belirli ve hassas bir şekilde özelleştirilebilir bir şekilde ayarlamak istiyorsanız kendi Weblate kopyanızda bunu seçin.

Bir projenin *Erişim denetimi* yapılandırması, ilgili projenin (*Yönetim* ↓ *Ayarlar*) *Erişim* sekmesinde değiştirilebilir.

The screenshot shows the Weblate interface with the 'Access' tab selected. Under 'Access control', there are four radio buttons: 'Public', 'Protected' (selected), 'Private', and 'Custom'. Each option has a description of its permissions. A message at the bottom indicates that the user does not have permission to change the project access control and suggests checking the billing status. A 'Save' button is located at the bottom left of the settings area.

Varsayılan değer *DEFAULT_ACCESS_CONTROL* ile değiştirilebilir.

Not: *Kişisel* projeler için bile, projenizle ilgili bazı bilgiler açığa çıkar: Uygulama kopyasının tamamı için istatistikler ve dil özetinde, erişim denetimi ayarına rağmen tüm projelerin sayıları bulunur. Proje adınız ve diğer bilgileriniz açığa çıkmaz.

Not: *Herkese açık*, *Korunmuş* ve *Kişisel* projelerde kullanıcılar için varsayılan olarak var olan gerçek izinler kümesi, Weblate kopyasının yöneticisi tarafından *özel ayar* kullanılarak yeniden tanımlanabilir.

Ayrıca bakınız:

Erişim denetimi

Her proje için erişim denetimi yönetimi

Proje erişimi yönetimi yetkisi olan kullanıcılar (bilgi almak için *Yetkiler ve hazır rollerin listesi*), projelerdeki kullanıcıları takımlara ekleyerek yönetebilir. İlk takım grubu Weblate tarafından sağlanır. Daha ayrıntılı erişim denetimi sağlamak için ek takımlar tanımlanabilir. Ekipleri dillerle sınırlayabilir ve onlara belirli erişim rolleri atayabilirsiniz (ayrıntılı bilgi almak için *Yetkiler ve hazır rollerin listesi*).

Her proje için şu takımlar kendiliğinden oluşturulur:

Herkese açık, *Korunmuş* ve *Kişisel* projeler için:

Yönetim

Proje için kullanılacak tüm izinleri kapsar.

Onaylama (yalnızca *onaylama iş akışı* özelliği açıksa)

Çevirileri gözden geçirip onaylayabilir.

Yalnızca *Korunmuş* ve *Kişisel* projeler için:

Çevir

Projeyi çevirebilir ve çevrim dışı yapılmış çevirileri yükleyebilir.

Kaynaklar

(*Proje ayarları* içinde izin verilmiş ise) kaynak dizgeleri ve kaynak dizge bilgilerini düzenleyebilir.

Diller

Çevrilecek dilleri yönetebilir (çevirileri ekleyebilir ya da kaldırabilir).

Sözlük

Sözlükleri yönetebilir (kayıtları ekleyebilir ya da kaldırabilir ve yükleyebilir).

Bellek

Çeviri belleğini yönetebilir.

Ekran görüntüleri

Ekran görüntülerini yönetebilir (ekleyip kaldırabilir ve kaynak dizgelerle ilişkilendirebilir).

Kendiliğinden çeviri

Kendiliğinden çevirileri kullanabilir.

Sürüm denetimi sistemi

Sürüm denetimi sisteminin yönetimi yapabilir ve dışa aktarılmış depoya erişebilir.

Faturalama

Faturalama bilgilerine ve ayarlarına erişebilir (bilgi almak için *Faturalama*).

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Access control

Users Teams API access

Users

Username	Full name	E-mail	Last sign in	Teams
testuser	Weblate Test	weblate@example.org	16 seconds ago	<div>Translate</div> <div></div>

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 4.17 About Weblate Legal Contact Documentation Donate to Weblate

Bu özellikler, projenin *Yönetim* ↓ *Kullanıcılar* menüsünden erişilebilen *Erişim denetimi* sayfasında bulunur.

Takım yöneticileri

4.15 sürümünde geldi.

Her takımda, takıma kullanıcı ekleyip kaldırabilen bir takım yöneticisi olabilir. Bu seçenek, kendi kendini yöneten takımlar oluşturmak istediğinizde kullanışlıdır.

Yeni kullanıcı daveti

Var olan bir kullanıcıyı projeye eklemenin yanında yeni kullanıcılar davet edilebilir. Yeni kullanıcılar hemen oluşturulur. E-posta yoluyla gönderilen davetlerde ise kullanıcı davet içindeki bağlantıyı kullanarak oturum açana kadar hesap etkinleştirilmez. Bunu yapabilmek için site genelinde herhangi bir yetki verilmesi gerekmez. Proje kapsamında erişim yönetimi izni (*Yönetim* takımının üyesi olmak gibi) yeterli olur.

İpucu: Davet edilen kullanıcı davetiye'nin geçerlilik süresini geçirdiyse, hesap zaten oluşturulmuş olduğundan, parola sıfırlama formunda davet edildiği e-posta adresini kullanarak parolasını ayarlayabilir.

3.11 sürümünde geldi: Kullanıcı davet e-postaları yeniden gönderilebilir (önceki davet e-postası geçersiz olur).

Aynı türde davetler site genelinde *Kullanıcılar* sekmesindeki *yönetim bölümü* içinden kullanılabilir.

Kullanıcıları engelleme

4.7 sürümünde geldi.

Projenizdeki bazı kullanıcıların kötü davranışlar sergilemesi durumunda, katkıda bulunmalarını engelleme seçeneğiniz vardır. Engellenen kullanıcının izinleri varsa, projeyi görmeye sürdürebilir. Ancak katkıda bulunamaz.

Her projeye özel izin yönetimi

Projelerinizi *Korunmuş* ya da *Kişisel* olarak ayarlayabilir ve Weblate kullanıcı arayüzünde her proje için *kullanıcıları yönetebilirsiniz*.

Varsayılan olarak bu durum, bu takımların kendi yapılandırması nedeniyle Weblete tarafından *Kullanıcılar* ve *Görüntüleyenler* :ref: varsayılan takımlar <default-teams> için sunulan erişim izinlerinin verilmesini engeller. Bu durum, varsayılan takımları değiştirerek, yeni bir tane oluşturarak ya da aşağıdaki *Özel erişim denetimi* bölümünde açıklandığı gibi tek tek bileşenler için ek özel ayarlar oluşturarak site genelinde bu projelere izin vermenizi engellemez.

Weblate kullanıcı arayüzü üzerinden izinleri yönetmenin en önemli avantajlarından biri, izinlerin diğer kullanıcılar süper kullanıcı yetkisi verilmeden devredilebilmesidir. Bunun için bu kullanıcıları projenin *Yönetim* ekibine ekleyin.

2.6.2 Özel erişim denetimi

Not: Bu özellik, Hosted Weblate üzerinde kullanılamaz.

İzin sistemi takımlara ve rollere dayanır. Roller bir dizi izin tanımlar ve takımlar bunları kullanıcılara ve çevirilere bağlar. Bilgi almak için *Kullanıcılar, roller, takımlar ve izinler* bölümüne bakabilirsiniz.

Weblate erişim denetimi sisteminin şimdilik en güçlü özellikleri yalnızca *Django yönetim arayüzü* üzerinden kullanılabilir. Bu arayüzden herhangi bir projenin izinlerini yönetebilirsiniz. Kullanabilmek için *Erişim denetimini* mutlaka *Özel* olarak değiştirmeniz gerekmez. Ancak süper kullanıcı yetkilerinizin olması gerekir.

Uygulamanın ayrıntılarıyla ilgilenmiyorsanız ve yalnızca varsayılan ayarlara dayanan yeterince basit bir yapılandırma oluşturmak istiyorsanız ya da Weblate kurulumunun tamamına site genelinde erişiminiz yoksa (*Hosted Weblate* gibi), lütfen *Basit erişim denetimi* bölümüne bakın.

Sık kullanılan kurulumlar

Bu bölümde, ilginizi çekebilecek bazı genel yapılandırmaların özetini bulabilirsiniz.

Site genelinde izin yönetimi

Uygulama kopyasının tamamındaki izinleri bir kerede yönetmek için kullanıcıları uygun *Varsayılan takımlar* ögesine ekleyin:

- *Kullanıcılar* (bu varsayılan olarak *kendiliğinden takım ataması* ile yapılır).
- *Onaylayanlar* (:ref: onaylayan kişiler ile onaylama iş akışını <reviews> kullanıyorsanız).
- *Yöneticiler* (yönetim işlemlerinin çoğunu başkasına devretmek istiyorsanız).

Tüm projeleri *Herkese açık* olarak yapılandırılmış tutmalısınız (bilgi almak için [Proje erişim denetimi](#)). Yoksa *Kullanıcılar* ve *Onaylayanlar* takımlarına üye olarak elde edilen site geneli izinlerin herhangi bir etkisi olmaz.

Varsayılan takımlara seçtiğiniz bazı ek izinleri de verebilirsiniz. Örneğin, tüm *Kullanıcılara* ekran görüntülerini yönetme izni vermek isteyebilirsiniz.

Bazı yeni özel takımlar da tanımlayabilirsiniz. Bu takımlar için izinlerinizi site genelinde yönetmeyi sürdürmek istiyorsanız, *Proje seçimi* için uygun bir değer seçin (*Tüm projeler* ya da *Tüm herkese açık projeler* gibi).

Diller, bileşenler veya projeler için özel izinler

Diller, bileşenler ve projeler gibi farklı nesnelerin izinlerini yönetmek için kendi ayrılmış takımlarınızı oluşturabilirsiniz. Bu takımlar yalnızca ek izinler verebilse de, başka bir özel takım ekleyerek site genelinde ya da her proje için takımlar tarafından verilen izinleri iptal edemezsiniz.

Örnek:

(Nedeni ne olursa olsun) belirli bir dilde çevirinin (*Çekçe* gibi) yalnızca belirli güvenilir bir çevirmen grubu tarafından yapılmasına izin vermek ve diğer dillerde yapılan çevirileri herkese açık tutmak istiyorsanız şunu yapmanız gerekir:

1. Tüm kullanıcılardan *Çekçe* dilini çevirme iznini kaldırın. Varsayılan yapılandırmada, *Kullanıcılar varsayılan takım* değiştirilerek yapılabilir.

Tablo 1: *Kullanıcılar* grubu

Dil seçimi	Tanımlandığı gibi
Diller	Çekçe dışında tümü

2. *Çekçe* çevirmenleri için özel bir takım ekleyin.

Tablo 2: *Çekçe çevirmenleri* grubu

Roller	Uzman kullanıcılar
Proje seçimi	Tüm herkese açık projeler
Dil seçimi	Tanımlandığı gibi
Diller	Çekçe

3. İzin vermek istediğiniz kullanıcıları bu takıma ekleyin.

Gördüğünüz gibi, bu şekilde izin yönetimi güçlüdür. Ancak oldukça sıkıcı bir iş olabilir. Süper kullanıcı izinleri vermediğiniz sürece izinleri başka bir kullanıcıya devredemezsiniz.

Kullanıcılar, roller, takımlar ve izinler

Kimlik doğrulama modelleri birkaç ögeden oluşur:

İzin

Weblate tarafından tanımlanan bireysel izin. İzinler kullanıcılara atanamaz. Bu yalnızca rol ataması ile yapılabilir.

Rol

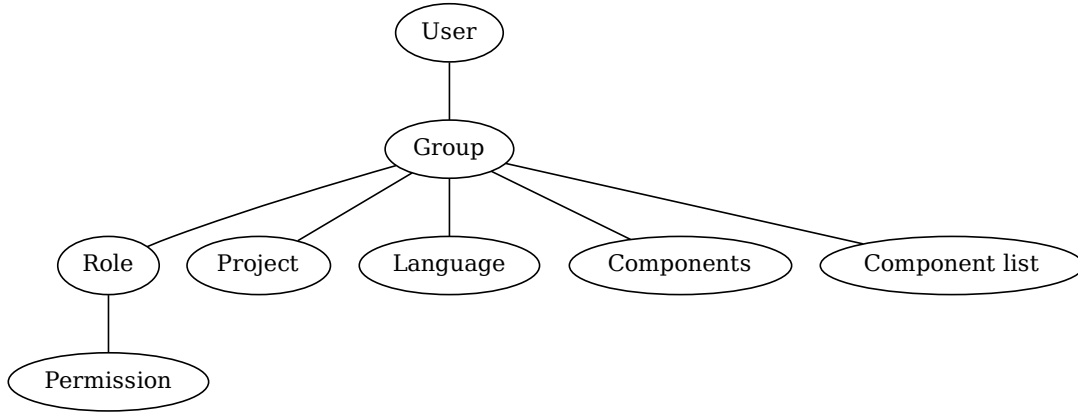
Rol, bir izin kümesi tanımlar. Böylece, bu kümeler birkaç yerde yeniden kullanılabilir ve yönetim kolaylaşır.

Kullanıcı

Kullanıcı birkaç takımın üyesi olabilir.

Grup

Gruplar, roller, kullanıcılar ve kimlik doğrulama nesneleri (projeler, diller ve bileşen listeleri) arasında bağlantı kurar.



Not: Bir takımın kendisine atanmış herhangi bir rolü olmayabilir. Bu durumda projeye herhangi birinin göz atma erişimi olacağı varsayılır (aşağıya bakın).

Bir projeye göz atma erişimi

Kullanıcı, projeye bağlı bir takımın veya bu projenin içindeki herhangi bir bileşenin üyesi olmalıdır. Üye olmak yeterlidir. Projeye göz atabilmek için ayrıca özel bir izin gerekmez (bu, varsayılan *Görüntüleyiciler* takımında kullanılır. Bilgi almak için [Takım listesi](#)).

Bir bileşene göz atma erişimi

Bir kullanıcı, bileşenlerin projesine erişebildiğinde kısıtlanmamış bileşenlere erişebilir (ve kullanıcıya proje için verilen tüm izinlere sahip olur). *Kısıtlı erişim* açıkken, bileşene erişim, bileşen (veya bileşenin içinde bulunduğu bileşen listesi) için açık izinler gerekir.

Takımların kapsamı

Takımlardaki roller tarafından atanan izin kapsamı şu kurallara göre uygulanır:

- Takım herhangi bir *Bileşen listesi* belirtirse, takıma bağlı bileşen listelerindeki tüm bileşenler için o takımın üyelerine tüm izinler verilir ve bu bileşenlerin içinde bulunduğu tüm projeler için ek izinleri olmayan bir erişim verilir. *Bileşenler* ve *Projeler* yok sayılır.
- Takımda herhangi bir *Bileşen* belirtiliyorsa, takıma bağlı tüm bileşenler için o takımın üyelerine tüm izinler verilir ve bu bileşenlerin içinde bulunduğu tüm projeler için ek izinleri olmayan bir erişim verilir. *Projeler* yok sayılır.
- Yoksa, takımda herhangi bir *Proje* belirtiliyorsa, bunları doğrudan listeleyerek veya *Proje seçimi* değerini *Tüm herkese açık projeler* gibi bir değere ayarlayarak, tüm bu izinler tüm projelere uygulanır ve bu da tüm projelere erişmek için aynı izinleri etkin bir şekilde verir *kısıtlanmamış bileşenler*.
- Bir takımın *Diller* değeri tarafından uygulanan kısıtlamalar, kullanıcının belirli işlemleri yapma izni olup olmadığına bakılarak ayrı olarak uygulanır. Yani, yalnızca çeviri sürecinin kendisiyle doğrudan ilgili onaylama, çevirileri kaydetme, öneri ekleme gibi işlemlere uygulanır.

İpucu: Tüm dillerin veya projelerin kendiliğinden katılması için *Dil seçimi* ya da *Proje seçimi* kullanın.

Örnek:

Diyelim ki şu bileşenleri olan bir `foo` projesi var: `foo/bar` ve `foo/baz` ve şu takım:

Tablo 3: Grup *İspanyol Yönetici-Onaylayıcılar*

Roller	<i>Dizgeleri onayla, Depo yönetimi</i>
Bileşenler	<code>foo/bar</code>
Diller	<i>İspanyolca</i>

Bu takımın üyelerinin şu izinleri olacak (varsayılan rol ayarları kabul edilerek):

- Şu iki bileşenin de bulunduğu `foo` projesi için genel (göz atma) izni: `foo/bar` ve `foo/baz`.
- `foo/bar` *İspanyolca* çevirisinde (başka bir yerde değil) dizgeleri onaylama izni.
- Tüm `foo/bar` deposu için sürüm denetimi sistemi yönetimi. Çevirmenler tarafından tüm diller için yapılmış bekleyen değişikliklerin işlenmesi gibi.

Kendiliğinden takım atamaları

Django yönetim arayüzü içindeki *Grup* düzenleme sayfasının alt bölümünde, yeni oluşturulan kullanıcıları e-posta adreslerine göre bir takıma kendiliğinden atanması için kullanılacak kurallı ifade listesi ile *Kendiliğinden takım ataması* yapabilirsiniz. Bu atama yalnızca hesap oluşturulduktan sonra yapılır.

Bu özelliğin en yaygın kullanım örneği, tüm yeni kullanıcıların bazı varsayılan takımlara atanmasıdır. Bunu yapmak için, büyük olasılıkla kurallı ifade alanındaki varsayılan değeri (`^^.*$`) korumak isteyeceksiniz. Bu seçeneğin başka bir kullanım örneği, varsayılan olarak şirketinizin çalışanlarına bazı ek ayrıcalıklar vermek olabilir. Hepsinin etki alanındaki kurumsal e-posta adreslerini kullandığını varsayarsak, bu `“^.*@mycompany.com”` gibi bir ifadeyle gerçekleştirilebilir.

Not: *Kullanıcılar* ve *Görüntüleyiciler* için kendiliğinden takım ataması, bir Weblate sürümünden diğerine yükseltme yaparken her zaman yeniden oluşturulur. Bu seçeneği kapatmak istiyorsanız, kurallı ifadeyi `“^$”` olarak ayarlayın (hiçbir şeyle eşleşmez).

Not: Şimdilik, zaten var olan kullanıcıları kullanıcı arayüzü üzerinden bazı takımlara toplu olarak eklemenin bir yolu yok. Bunun için *REST API* kullanmayı deneyebilirsiniz.

Varsayılan takımlar ve roller

Kurulumdan sonra, bir varsayılan takım kümesi oluşturulur (bilgi almak için *Takım listesi*).

Bu roller ve takımlar kurulum sonrasında oluşturulur. Hazır roller, yükseltme yapılırken veri tabanı aktarımı tarafından her zaman güncel tutulur. Bunları gerçekten değiştiremezsiniz. Kendi izin kümenizi tanımlamak istiyorsanız lütfen yeni bir rol tanımlayın.

Yetkiler ve hazır rollerin listesi

Kapsam	İzin	Roller
Faturalama (bilgi almak için <i>Faturalama</i>)	Fatura bilgilerini görüntüle	<i>Yönetim, Faturalama</i>
Değişiklikler	Değişiklikleri indir	<i>Yönetim</i>
Yorumlar	Yorum gönderme	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
	Yorum silme	<i>Yönetim</i>
	Yorumu çözümleme	<i>Yönetim, Dizgeleri onaylama</i>
Bileşen	Bileşen ayarlarını düzenleme	<i>Yönetim</i>
	Bileşeni kilitleme, çevirileri engelleme	<i>Yönetim</i>
Sözlük	Sözlük kaydı ekleme	<i>Yönetim, Sözlük yönetimi, Uzman kull</i>
	Sözlük kaydını düzenleme	<i>Yönetim, Sözlük yönetimi, Uzman kull</i>
	Sözlük kaydını silme	<i>Yönetim, Sözlük yönetimi, Uzman kull</i>
	Sözlük kayıtları yükleme	<i>Yönetim, Sözlük yönetimi, Uzman kull</i>
Kendiliğinden öneriler	Kendiliğinden önerileri kullanma	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
Çeviri belleği	Çeviri belleğini düzenleme	<i>Yönetim, Çeviri belleği yönetimi</i>
	Çeviri belleğini silme	<i>Yönetim, Çeviri belleği yönetimi</i>
Projeler	Proje ayarlarını düzenleme	<i>Yönetim</i>
	Proje erişimini yönetme	<i>Yönetim</i>
Raporlar	Raporları indirme	<i>Yönetim</i>
Ekran görüntüleri	Ekran görüntüsü ekleme	<i>Yönetim, Ekran görüntüleri yönetimi</i>
	Ekran görüntüsü düzenleme	<i>Yönetim, Ekran görüntüleri yönetimi</i>
	Ekran görüntüsü silme	<i>Yönetim, Ekran görüntüleri yönetimi</i>
Kaynak dizgeleri	Ek dizge bilgilerini düzenleme	<i>Yönetim, Kaynağı düzenleme</i>
Dizgeleri	Yeni dizge ekleme	<i>Yönetim</i>
	Bir dizgeyi silme	<i>Yönetim</i>
	Denetim hatasını yok sayma	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
	Dizgeleri düzenleme	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
	Dizgeleri onaylama	<i>Yönetim, Dizgeleri onaylama</i>
	Öneriler dayatıldığında dizgeyi düzenleme	<i>Yönetim, Dizgeleri onaylama</i>
	Kaynak dizgeleri düzenleme	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
Öneriler	Öneriyi kabul etme	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
	Öneri ekleme	<i>Yönetim, Kaynağı düzenleme, Öneri e</i>
	Öneri silme	<i>Yönetim, Uzman kullanıcı</i>
	Öneriye oy verme	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
Çeviriler	Çeviri dili ekleme	<i>Yönetim, Uzman kullanıcı, Dil yönetimi</i>
	Kendiliğinden çevirme	<i>Yönetim, Kendiliğinden çevirme</i>
	Var olan çeviriyi silme	<i>Yönetim, Dil yönetimi</i>
	Çeviri dosyasını indirme	<i>Yönetim, Kaynağı düzenleme, Depoya</i>
	Çevirilecek birkaç dil ekleyin	<i>Yönetim, Dil yönetimi</i>
Yüklemeler	Yüklenmiş çeviriye katkıda bulunanı tanımlama	<i>Yönetim</i>
	Yükleme ile var olan dizgelerin üzerine yazma	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
	Çeviri yükleme	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
Sürüm denetimi sistemi	İç depoya erişme	<i>Yönetim, Depoya erişme, Uzman kull</i>
	Değişiklikleri iç depoya işleme	<i>Yönetim, Depo yönetimi</i>
	İç depodaki değişiklikleri işleme	<i>Yönetim, Depo yönetimi</i>
	İç depodaki değişiklikleri sıfırlama	<i>Yönetim, Depo yönetimi</i>
	Yukarı akış deposu konumunu görüntüleme	<i>Yönetim, Depoya erişme, Uzman kull</i>
	İç depoyu güncelleme	<i>Yönetim, Depo yönetimi</i>
Site genelindeki izinler	Yönetim arayüzünü kullanma	
	Yeni projeler ekleme	
	Dil tanımları ekleme	
	Dil tanımlı yönetimi	
	Takım yönetimi	
	Kullanıcı yönetimi	

Tablo 4 – önceki sayfadan devam

Kapsam	İzin	Roller
	Rol yönetimi	
	Duyuru yönetimi	
	Çeviri belleği yönetimi	
	Makine yönetimi	
	Bileşen listeleri yönetimi	

Not: Site genelindeki izinler herhangi bir varsayılan role verilmez. Bu izinler güçlüdür ve süper kullanıcı yetkilerine oldukça yakındır. Çoğu Weblate kurulumundaki tüm projeleri etkiler.

Takım listesi

Aşağıdaki takımlar kurulumun ardından (ya da *setupgroups* çalıştırıldıktan sonra) oluşturulur ve bunları istediğiniz gibi değiştirebilirsiniz. Ancak, silerseniz ya da yeniden adlandırırsanız aktarım işlemi bunları yeniden oluşturur.

Konuklar

Kimliği doğrulanmamış kullanıcıların izinlerini belirler.

Bu takımda yalnızca anonim kullanıcılar bulunur (bilgi almak için *ANONYMOUS_USER_NAME*).

Kimliği doğrulanmamış kullanıcıların izinlerini sınırlamak için bu takımdaki rolleri kaldırabilirsiniz.

Varsayılan roller: *Öneri ekleme*, *Depoya erişme*

Görüntüleyiciler

Bu rol, herkese açık projelerin tüm kullanıcılara görüldüğünden emin olunmasını sağlar. Varsayılan olarak tüm kullanıcılar bu grubun üyesidir.

Varsayılan olarak, *kendiliğinden takım ataması*, katılan tüm yeni hesapları bu takımın bir üyesi yapar.

Varsayılan roller: Yok

Kullanıcılar

Tüm kullanıcıların varsayılan takımı.

Varsayılan olarak, *kendiliğinden takım ataması*, katılan tüm yeni hesapları bu takımın bir üyesi yapar.

Varsayılan roller: *Uzman kullanıcı*

Onaylayanlar

Onaylayan kullanıcıların takımı (bilgi almak için *Çeviri iş akışları*).

Varsayılan roller: *Dizgeleri onaylama*

Yöneticiler

Yönetici kullanıcıların takımı.

Varsayılan roller: *Yönetim*

Uyarı: Hazır gelen Weblate takımlarını ve kullanıcılarını asla silmeyin. Bu işlem beklenmedik sorunlara yol açabilir! Onları kullanmıyorsanız, silmek yerine tüm izinlerini kaldırabilirsiniz.

2.6.3 Ek erişim kısıtlamaları

Weblate kurulumunuzu herkese daha az açık olacak biçimde kullanmak, yani yeni kullanıcıları yalnızca davet ederek izin vermek istiyorsanız, Weblate yapılandırmasını yalnızca bilinen kullanıcıların erişebileceği şekilde ayarlayabilirsiniz. Bu amaçla, yeni kullanıcıların hesap açmasını önlemek için [REGISTRATION_OPEN](#) seçeneğini `Yanlış` olarak ve [REQUIRE_LOGIN](#) seçeneğini `/'` olarak ayarlayarak tüm site sayfalarına erişmek için oturum açmayı zorunlu kılabilirsiniz. Böylece kolayca Weblate kurulumunuzu kilitlemiş olursunuz.

İpucu: Yeni kullanıcılar eklemek için iç *Yeni kullanıcı daveti* komutunu kullanabilirsiniz.

2.7 Çeviri projeleri

2.7.1 Çeviri kuruluşu

Weblate, projenin/bileşenlerin çevrilebilir sürüm denetimi sistemi içeriğini ağaç benzeri bir yapıya dönüştürür.

- En alt düzeydeki nesne *Proje yapılandırması* olur ve tüm çevirileri bir arada bulundurmalıdır (bir uygulamanın birkaç sürümünün ve/veya ekindeki belgelerin çevirisi gibi).
- Yukarıdaki düzeyde, çevrilecek bileşen olan *Bileşen yapılandırması*, kullanılacak sürüm denetimi sistemi deposu ve çevrilecek dosyaların maskesi gibi bilgileri tanımlayabilirsiniz.
- Yukarıda *Bileşen yapılandırması* sürüm denetimi sistemi deposunda, çeviri dosyaları (*Bileşen yapılandırması* içinde tanımlanan *Dosya maskesi* ile eşleşen) Weblate tarafından otomatik olarak işlenen bireysel çeviriler bulunur.

Weblate, Translate Toolkit tarafından desteklenen çok çeşitli çeviri biçimlerini (hem iki dilli hem de tek dilli olanları) destekler. Bilgi almak için: `ref:formats`.

Not: Kopyalanmış sürüm denetimi sistemi depolarını *İç Weblate adresleri* ile paylaşabilirsiniz. Aynı sürüm denetimi sistemini paylaşan birçok bileşeniniz olduğunda bu özelliği kullanmanız önemle önerilir. Başarımı artırır ve kullanılacak disk alanını azaltır.

2.7.2 Çeviri projelerini ve bileşenleri eklemek

3.2 sürümünde değişti: Proje ve bileşen eklemek için bir arayüz vardır ve artık *Django yönetim arayüzü* kullanmanız gerekmez.

3.4 sürümünde değişti: Bileşen ekleme işlemi artık çoğu parametrenin otomatik olarak keşfedilmesiyle çok aşamalı olarak yapılır.

İzinlerinize göre, yeni çeviri projeleri ve bileşenleri ekleyebilirsiniz. *Yeni proje ekleyebilir* izni olan kullanıcılara her zaman izin verilir ve bulut sunucunuz faturalama kullanıyorsa (<https://hosted.weblate.org/> gibi, bilgi almak için: *Faturalama*), faturalamayı yöneten kullanıcı hesabından tarifenizin sınırlarına göre de bunları oluşturabilirsiniz.

Var olan faturalama tarifenizi ayrı bir sayfada görüntüleyebilirsiniz:

[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	2023-04-16 - 2023-04-18
Projects limit	Used 0 of 1 <div></div>
Projects	No projects currently assigned! Add new translation project
Terminate billing plan	

Invoices		
Invoice period	Invoice amount	Download invoice
04/16/2023 - 04/18/2023	19.0 EUR	Not available

Powered by Weblate 4.17 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Proje ekleme işlemi oradan ya da başlatılabilir ya da gezinme çubuğundaki menüden eklemeyi tamamlamak için çeviri projesiyle ilgili temel bilgiler doldurulabilir:

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

[Create project](#)

[Add new translation project](#)
[Import translation project](#)

Project name

Display name

URL slug

Name used in URLs and filenames.

Project website

Main website of translated project.

Translation instructions

https://weblate.org/contribute/

You can use Markdown and mention users by @username.

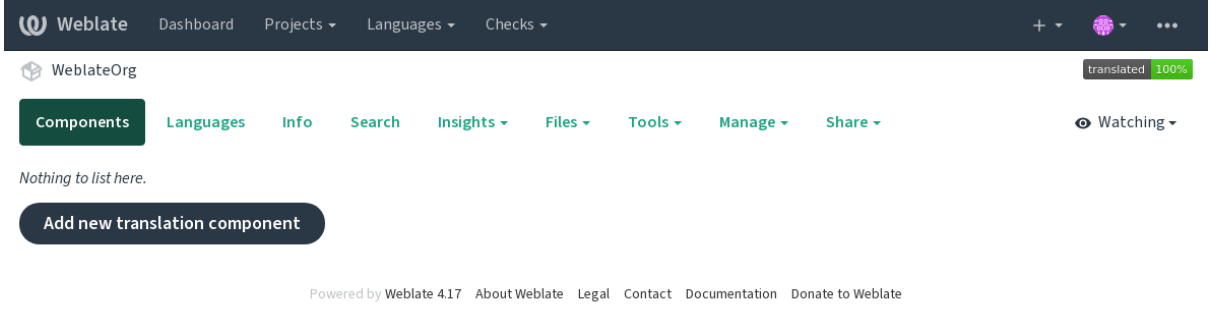
Billing

Weblate Test (Basic plan)

[Save](#)

Powered by Weblate 4.17 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Proje eklendikten sonra, doğrudan proje sayfasına yönlendirilirsiniz:



Yeni bir çeviri bileşeni oluşturmak, oradan tek bir tıklama ile başlatılabilir. Bileşen ekleme işleminin birkaç aşaması vardır ve çoğu çeviri parametresini otomatik olarak algılar. Bileşen eklemek için birkaç yaklaşım vardır:

Sürüm denetiminden

Bileşeni uzak sürüm denetimi deposundan ekler.

Var olan bileşenden

Bileşeni seçilmiş farklı dosyalar ile var olan bileşene ek bileşen olarak ekler.

Ek dal

Bileşeni yalnızca farklı dallar için var olan bileşene ek bileşen olarak ekler.

Çeviri dosyalarını yükleyerek

Sürüm denetimi kullanmadığınız ya da Weblate ile bütünleştirmek istemediğiniz durumlarda çeviri dosyalarını Weblate üzerine yükleyebilirsiniz. İçeriği daha sonra site arayüzü ya da [Weblate REST API uygulaması](#) ile güncelleyebilirsiniz.

Belgeyi çevirerek


Tek bir belge veya çeviri dosyası yükleyip ve çevirebilirsiniz.

Sıfırdan başlayarak

Boş bir çeviri projesi oluşturabilir ve dizgeleri el ile ekleyebilirsiniz.

Çeviri bileşenlerini ekledikten sonra, aynı depoyu kullanarak ek dosyalar ya da dallar için kolayca yeni bileşenler ekleyebilirsiniz.

Öncelikle adı ve depo konumunu yazmanız gerekir:

 Weblate


Dashboard

Projects ▾

Languages ▾

Checks ▾

+ ▾



...

Create component

From version control

Upload translations files

Translate document

Start from scratch

Create a new translation component from remote version control system repository.

Component name ⓘ

Display name

URL slug ⓘ

Name used in URLs and filenames.

☐ Use as a glossary ⓘ

Project ⓘ

WebplateOrg ▾

Source language ⓘ

English ▾

Language used for source strings in all components

Version control system ⓘ

Git ▾

Version control system to use to access your repository containing translations. You can also choose additional integration with third party providers to submit merge requests.

Source code repository ⓘ

URL of a repository, use weblate://project/component to share it with other component.


Repository branch ⓘ

Repository branch to translate

Continue

Powered by Weblate 4.17 About Weblate Legal Contact Documentation Donate to Weblate

Sonraki sayfada, keşfedilmiş çevrilebilir kaynakların bir listesini göreceksiniz:

 Weblate


Dashboard

Projects ▾

Languages ▾

Checks ▾

+ ▾



...

Create component

Add new translation component ⓘ

Choose translation files to import ⓘ

☐ Specify configuration manually

☐ File format Android String Resource , File mask app/src/main/res/values-*/strings.xml

☐ File format gettext PO file , File mask weblate/langdata/locale/*/LC_MESSAGES/django.po


☐ File format gettext PO file , File mask weblate/locale/*/LC_MESSAGES/django.po


☐ File format gettext PO file , File mask weblate/locale/*/LC_MESSAGES/djangojs.po

Continue

Powered by Weblate 4.17 About Weblate Legal Contact Documentation Donate to Weblate

Son olarak, çeviri bileşeni bilgilerini gözden geçirebilir ve isteğe bağlı bilgileri yazabilirsiniz:


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

Ayrıca bakınız:

Django yönetim arayüzü, Proje yapılandırması, Bileşen yapılandırması

2.7.3 Proje yapılandırması

Bir çeviri projesi oluşturun ve içine çevrilecek yeni bir bileşen ekleyin. Proje, gerçek çevirilerin konulduğu bir raf gibidir. Aynı projedeki tüm bileşenler önerileri ve sözlüğü paylaşır. Ayrıca yapılan çeviriler tek bir projedeki tüm bileşenlere kendiliğinden yayılır (bileşen yapılandırmasında kapatılmadıkça). Bilgi almak için: *Çeviri belleği*.

Ayrıca bakınız:

/devel/integration

Bu temel öznelikler kurulur ve bir projenin çevirmenlerine bildirilir:

Proje adı

Proje adını görüntülemek için kullanılan ayrıntılı proje adı.

Ayrıca bakınız:

PROJECT_NAME_RESTRICT_RE

Adres kısaltması

Projenin adreslerde kullanılmaya uygun adı.

Proje sitesi

Çevirmenlerin proje ile ilgili ayrıntılı bilgi bulabileceği adres.

Bu parametre *WEBSITE_REQUIRED* tarafından kapatılmadığı sürece zorunludur.

Ayrıca bakınız:

PROJECT_WEB_RESTRICT_HOST, *PROJECT_WEB_RESTRICT_NUMERIC,* *PRO-*
JECT_WEB_RESTRICT_RE

Çeviri yönergeleri

Projedeki yerelleştirme sürecini açıklayan metin ve çevirmenler için yararlı olabilecek diğer bilgiler. Metin biçimlendirme ya da bağlantı ekleme işlemleri için markdown kullanılabilir.

“Language-Team” üst bilgisi ayarlansın

Language-Team üst bilgisinin Weblate tarafından yönetmesinin gerekip gerekmediği (bu şu anda yalnızca bir *GNU gettext* özelliğidir).

Paylaşılan çeviri belleği kullanılsın

Paylaşılan çeviri belleğinin kullanılıp kullanılmayacağı. Bilgi almak için: *Paylaşılan çeviri belleği*.

Varsayılan değer `DEFAULT_ACCESS_CONTROL` seçeneği ile değiştirilebilir.

Paylaşılan çeviri belleğine katkıda bulunulsun

Paylaşılan çeviri belleğine katkıda bulunulup bulunulmayacağı. Bilgi almak için: *Paylaşılan çeviri belleği*.

Varsayılan değer `DEFAULT_ACCESS_CONTROL` seçeneği ile değiştirilebilir.

Not: Bu özellik, Hosted Weblate üzerinde kullanılamaz.

Erişim denetimi

Her proje için erişim denetimi yapılandırması. Bilgi almak için: `:ref:`acl``.

Varsayılan değer `DEFAULT_ACCESS_CONTROL` ile değiştirilebilir.

Onaylama kullanılsın

Çeviriler için onaylama iş akışını etkinleştirir. Bilgi almak için: *Özel onaylayanlar*.

Kaynak onaylama kullanılsın

Kaynaklar için onaylama iş akışını etkinleştirir. Bilgi almak için: *Kaynak dizge onayları*.

Ayrıca bakınız:

report-source, *Yorumlar*

Kancalar kullanılsın

Bu depo için kimliği doğrulanmamış *Bildirim kancaları* özelliğinin kullanılıp kullanılmayacağı.

Ayrıca bakınız:

Ara dil dosyası, *Kaynak dizgeleri için kalite aracı*, *İki dilli ve tek dilli biçimler*, *Dil tanımları*

Dil kod adları

Çevirileri Weblate içine aktarmak için kullanılacak dil kodları eşleştirmesini tanımlayın. Dil kodları depolarınızda tutarsız olduğunda ve Weblate üzerinde tutarlı bir görünüm elde etmek istediğinizde ya da çeviri dosyalarınızda standart olmayan adlandırma kullanmak istediğinizde bunu kullanın.

Tipik kullanım örneği, Amerikan İngilizcesi ile İngiliz İngilizcesi'ni eşleştirmek olabilir: `en_US:en`

Birden fazla eşleştirmeyi virgül ile ayırarak yazabilirsiniz: `en_GB:en,en_US:en`

Standart olmayan kod kullanmak: `ia_FOO:ia`

İpucu: Çeviri dosyaları eşleştirilirken dil kodlarına bakılır ve eşleşmeler büyük/küçük harfe duyarlıdır. Bu nedenle kaynak dil kodlarını dosya adlarındakiler ile aynı şekilde kullandığınızdan emin olun.

Ayrıca bakınız:

adding-translation, *Dil kodu*, *Dil kodlarını işlemek*

2.7.4 Bileşen yapılandırması

Bir bileşen, çevrilecek bir şeyin gruplanmasıdır. Çevrilmesini istediğiniz dosyalar için bir sürüm denetimi sistemi deposu konumu ve dosya maskesi belirtirsiniz ve Weblate otomatik olarak bu sürüm denetimi sisteminden eşleşen tüm çevrilebilir dosyaları bulup alır.

Ayrıca bakınız:

/devel/integration

Bazı yaygın yapılandırma örneklerini *Desteklenen dosya biçimleri* bölümünde bulabilirsiniz.

Not: Çeviri bileşenlerini makul bir boyutta tutmanız önerilir. Çeviriyi sizin durumunuzda anlamlı olan herhangi bir şeye göre bölün (bireysel uygulamalar ya da eklentiler, kitap bölümleri ya da siteler).

Weblate, onbinlerce dizgeden oluşan çevirileri kolayca işler. Ancak işi bölmek ve bu kadar büyük çeviri bileşenlerinin çevirmenler arasında koordinasyonunu sağlamak daha zordur.

Bir çevirinin dil tanımı eksikse, boş bir tanım oluşturulur ve “tr_TR (oluşturulmuş)” olarak adlandırılır. Tanımı ayarlamalı ve eksik dillerin bir sonraki sürüme katılabilmesi için bunu Weblate geliştiricilerine bildirmelisiniz.

Bileşende, sürüm denetim sistemi ile çalışmak ve çevirileri almak için gereken tüm önemli parametreler bulunur:

Bileşen adı

Bileşen adını görüntülemek için kullanılan ayrıntılı bileşen adı.

Bileşen kısaltması

Bileşenin adreslerde kullanılmaya uygun adı.

Bileşenin projesi

Proje yapılandırması bileşenin ait olduğu yer.

Sürüm denetimi sistemi

Kullanılacak sürüm denetimi sistemi. Bilgi almak için: *Sürüm denetimi bütünleştirilmesi*.

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri itmek

Kaynak kod deposu

Değişikliklerin alınacağı sürüm denetimi sistemi deposu.

Ayrıca bakınız:

Adresleri belirtmek ile ilgili bilgi almak için: [Depolara erişmek](#).

İpucu: Bu, gerçek bir sürüm denetimi sistemi adresi ya da deponun başka bir bileşenle paylaşılması gerektiğini belirten `weblate://project/component` olabilir. Bilgi almak için: [İç Weblate adresleri](#).

Depo itme adresi

İtme için kullanılacak depo adresi. Bu ayar yalnızca [Git](#) ve [Mercurial](#) için kullanılır ve boş olduğunda bunlar için itme desteği kapatılır.

Bağlantılı depolar için bu kullanılmaz ve bağlantılı bileşendeki ayar uygulanır.

Ayrıca bakınız:

Bir depo adresinin nasıl belirtileceği ile ilgili bilgi almak için: [Depolara erişmek](#) ve Weblate üzerinden değişiklikleri itmek ile ilgili bilgi almak için: [Weblate üzerindeki değişiklikleri itmek](#) bölümlerine bakabilirsiniz.

Depo tarayıcısı

Kaynak dosyaları görüntülemek için kullanılan depo tarayıcısının adresi (kullanılan iletilerin konumu). Boş olduğunda, böyle bir bağlantı oluşturulmaz. [Kalıp kodlaması](#) kullanabilirsiniz.

Örneğin, GitHub üzerinde şöyle bir şey kullanın: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

Yollarınızın farklı bir klasöre göre olması durumunda (yolda `..` varsa), baştaki klasörü ```parentdir` süzgeciyle atmak isteyebilirsiniz (bilgi almak için: [Kalıp kodlaması](#)): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

Dışa aktarılmış depo adresi

Weblate tarafından yapılan değişikliklerin dışa aktarıldığı adres. Bu adres, [Sürekli yerelleştirme](#) kullanılmadığında ya da değişiklikleri el ile birleştirmeye gerek duyulduğunda önemli olur. Git depoları için otomatikleştirmek için [Git dışa aktarıcı](#) kullanabilirsiniz.

Depo dalı

Sürüm denetimi sistemindeki hangi daldan çıkış yapılacağı ve çeviriler için nereye bakılacağı.

Bağlantılı depolar için bu kullanılmaz ve bağlantılı bileşendeki ayar uygulanır.

İtme işleminin yapılacağı dal

Değişikliklerin gönderileceği dal, *Depo dalı* kullanılması için boş bırakın.

Bağlantılı depolar için bu kullanılmaz ve bağlantılı bileşendeki ayar uygulanır.

Not: Bu özellik şu anda yalnızca Git, GitLab ve GitHub için destekleniyor. Diğer sürüm denetimi sistemi bütünleştirimleri yok sayılır.

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri itmek

Dosya maskesi

Yol ile birlikte çevrilecek dosyaların maskesi. Dil kodunun yerine bir “*” bulunmalıdır (bunun nasıl işlendiği ile ilgili bilgi almak için: *Dil tanımları*). Deponuzda birden fazla çeviri dosyası varsa daha fazla gettext etki alanları gibi), her biri için bir bileşen oluşturmanız gerekir.

Örneğin `po/* .po` ya da `locale/*/LC_MESSAGES/django.po`.

Dosya adınızda `[,]` gibi özel karakterler bulunması durumunda, bunlardan `[]` ya da `[]` gibi kaçınılması gerekir.

Ayrıca bakınız:

İki dilli ve tek dilli biçimler, “Tek dil (en) için daha fazla dosya var” ne anlama geliyor?

Tek dilli temel dil dosyası

Tek dilde bileşenler için dizge tanımlarını içeren temel dosya.

Ayrıca bakınız:

İki dilli ve tek dilli biçimler, “Tek dil (en) için daha fazla dosya var” ne anlama geliyor?

Temel dosyayı düzenle

Tek dilde bileşenler için temel dosyanın düzenlenmesine izin verilip verilmeyeceği.

Ara dil dosyası

Tek dilde bileşenler için ara dil dosyası. Çoğu durumda bu, geliştiriciler tarafından sağlanan bir çeviri dosyasıdır ve gerçek kaynak dizgeleri oluşturulurken kullanılır.

Ayarlandığında, kaynak dizgeler bu dosyayı temel alır. Ancak diğer tüm diller *Tek dilli temel dil dosyası* dosyasını temel alır. Dizgenin kaynak dile çevrilmemesi durumunda, diğer dillere çeviri yapmak engellenir. Bu, *Kaynak dizgeler için kalite aracısı* sağlar.

Ayrıca bakınız:

Kaynak dizgeler için kalite aracısı, İki dilli ve tek dilli biçimler, “Tek dil (en) için daha fazla dosya var” ne anlama geliyor?

Yeni çeviriler kalıbı

Yeni çeviriler oluşturmak için kullanılan temel dosya.

- Tek biçimli biçimlerin çoğu için bu alanı boş bırakın. Bunlar genellikle boş bir dosya ile başlayabilir.
- GNU gettext po dosyaları için ``.pot`` dosyası seçin.
- Varsa, çevirileri olmayan boş dosyayı seçin.
- Tam bir anahtar kümesinin bulunması gereken tek dilli biçimler için *Tek dilli temel dil dosyası* seçin.
- Belge çevirileri için *Tek dilli temel dil dosyası* seçin.
- Diğerleri için herhangi bir çeviri dosyası seçin.
- Kalıp dosyası çoğu durumda temel dosyayla aynı olabilir.

İpucu: Birçok tek dilli biçimde Weblate varsayılan olarak boş dosya ile başlar. Yeni çeviri oluştururken tüm dizgelerin değerinin boş olmasını istiyorsanız bunu kullanın.

Ayrıca bakınız:

adding-translation, *Yeni çeviri ekleme*, *İki dilli ve tek dilli biçimler*, *“Tek dil (en) için daha fazla dosya var” ne anlama geliyor?*

Dosya biçimi

Çeviri dosyası biçimi. Bilgi almak için: *Desteklenen dosya biçimleri*.

Kaynak dizge hata bildirme adresi

Yukarı akış hatalarını bildirmek için kullanılacak e-posta adresi. Bu adres, Weblate üzerinde yapılan tüm kaynak dizge yorumları ile ilgili bildirimleri de alır.

Çevirilerin yayılmasını sağlamak

Çevirilerin aynı projedeki diğer bileşenlerden bu bileşene yayılmasını kapatabilirsiniz. Bu seçenek gerçekte neyin çevirisini yaptığınıza bağlıdır, bazen bir çeviriden birden fazla kez yararlanmak isteyebilirsiniz.

Tüm projede aynı kimlikleri kullanmadığınız sürece, tek dilli çeviriler için bu seçeneği kapatmak genellikle iyi bir fikirdir.

Varsayılan değer `DEFAULT_TRANSLATION_PROPAGATION` ile değiştirilebilir.

Ayrıca bakınız:

Bileşenler arasında çevirilerin tutarlılığını sağlamak

Öneriler kullanılsın

Bu bileşen için çeviri önerilerinin kabul edilip edilmeyeceği.

Öneriler oylanabilsin

Öneriler için oy verme özelliğini açar. Bilgi almak için: [Öneriler oylanabilsin](#).

Öneriler kendiliğinden kabul edilsin

Oylanan önerilerin kendiliğinden kabul edilmesini sağlar. Bilgi almak için: [Öneriler oylanabilsin](#).

Çeviri imleri

Kalite denetimlerinin ve diğer Weblate davranışlarının özelleştirilmesi. Bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#).

Zorunlu denetimler

Yok sayılamayacak denetimlerin listesi. Bilgi almak için: [enforcecing-checks](#).

Not: Denetim dayatması kendiliğinden etkinleştirilmez. `component-check_flags` checks ya da [Kaynak dizgeleri ile ilgili ek bilgiler](#) içindeki [Parametreleri kullanarak davranışı özelleştirmek](#) ile etkinleştirmeniz gerekir.

Çeviri lisansı

Çevirinin lisansı (kaynak kodunun lisansı ile aynı olması gerekmez).

Katılımcı anlaşması

Bir kullanıcının bu bileşeni çevirmeden önce kabul etmesi gereken katılımcı anlaşması.

Yeni çeviri ekleme

Yeni dil ekleme isteklerinin nasıl işleneceği. Kullanılabilecek seçenekler:

Sorumlulara bildirim

Kullanıcı istediği dili seçebilir ve proje sorumlularına bununla ilgili bir bildirim gönderilir. Dili depoya eklemek (ya da eklememek) proje sorumlusuna kalır.

Çeviri yönergeleri adresi görüntülensin

Kullanıcıya, yeni çevirilere başlama sürecini açıklayan bir sayfa bağlantısı sunulur. Bu seçeneği daha resmi bir süreç istenmesi durumunda kullanabilirsiniz (çeviriye başlamadan önce bir ekip oluşturmak gibi).

Yeni dil dosyası oluşturulsun

Kullanıcı dili seçebilir, Weblate dosyayı kendiliğinden oluşturur ve çeviri başlayabilir.

Yeni çevirilerin eklenmesi devre dışı bırakılsın

Kullanıcıya yeni çeviriye başlaması için bir seçenek sunulmaz.

İpucu: Proje yöneticileri, burada devre dışı bırakılmış olsa bile yeni çeviriler ekleyebilir ([Yeni çeviriler kalıbı](#) ya da boş bir dosyadan başlamayı destekleyen dosya biçimi).

Ayrıca bakınız:

adding-translation, [Yeni çeviriler kalıbı](#)

Dizge yönetimi

4.5 sürümünde geldi.

Weblate üzerindeki kullanıcıların yeni dizgeleri eklemesine ve var olan dizgeleri silmesine izin verilip verilmeyeceğini yapılandırır. Bu seçeneği, yerelleştirme iş akışınıza (yeni dizgelerin nasıl tanıtılması gerektiğine) uyacak şekilde ayarlayın.

İki dilli biçimler için, dizgeler genellikle kaynak kodundan ayıklanır (örneğin, **xgettext** kullanılarak) ve Weblate üzerine yeni dizgelerin eklenmesi devre dışı bırakılmalıdır (çeviri dosyalarının bir sonraki güncellenmesinde silinirler). Weblate üzerinde her çevirinin dizgelerini yönetebilirsiniz ve tüm çevirilerdeki dizgelerin tutarlı olmasını dayatılmaz.

Tek dilli biçimlerde, dizgeler yalnızca kaynak dilde yönetilir ve çevirilerde kendiliğinden eklenir ya da silinir. Dizgeler, çevrildikten sonra çeviri dosyalarında görünür.

Ayrıca bakınız:

İki dilli ve tek dilli biçimler, `adding-new-strings`, `POST /api/translations/(string:project)/(string:component)/(string:language)/units/`

Dil kodu biçimi

Weblate tarafından oluşturulan çevirilerin dosya adını oluşturmakta kullanılacak dil kodunu özelleştirin.

Not: Weblate, çeviri dosyalarını işlerken dil kodlarını tanır. Bu ayarlar yalnızca yeni dosyaların nasıl oluşturulacağını etkiler.

Dosya biçimine göre varsayılan

Dosya biçimine bağlıdır. Çoğu için POSIX kullanılır.

Ayraç olarak alt çizgi kullanan POSIX biçimi

Genellikle gettext ve ilgili araçlar tarafından kullanılır. `pt_BR` gibi dil kodlarını üretir.

Ülke kodu ile birlikte araç olarak alt çizgi kullanan POSIX biçimi

Gerekli olmadığında bile ülke kodunu içeren POSIX biçimindeki dil kodu (`tr_TR` gibi).

Ayraç olarak tire kullanan BCP biçimi

Genellikle internet platformlarında kullanılır. “pt-BR” gibi dil kodlarını üretir.

Ülke kodu ile birlikte araç olarak tire kullanan BCP biçimi

Gerekli olmadığında bile ülke kodunu içeren BCP biçimindeki dil kodu (`tr-TR` gibi).

Ayraç olarak tire kullanan BCP biçimi, eski dil kodları

Çince ve BCP biçiminde gösterim için eski kodları kullanır.

Ayraç olarak tire kullanan BCP biçimi, küçük harfli

BCP biçiminde gösterim. Tümü küçük harfle (örneğin `tr-tr`).

Apple App Store üst veri biçimi

Üst verileri Apple App Store üzerine yüklemeye uygun biçim.

Google Play üst veri biçimi

Üst verileri Google Play Store üzerine yüklemeye uygun biçim.

Android biçimi

Yalnızca Android uygulamalarında kullanıldığında, `tr-TR` gibi dil kodları üretir.

Linux biçimi

Linux tarafından kullanılan yerel ayarlar, Çince ve POSIX biçiminde gösterim için eski kodları kullanır.

Ayrıca bakınız:

`adding-translation`, *Dil kodu*, *Dil kodlarını işlemek*, *Dil kod adları*

Birleştirme biçemi

Yukarı akış deposundaki güncellemelerin nasıl işleneceğini yapılandırabilirsiniz. Gerçek uygulama sürüm denetimi sistemine bağlıdır. Bilgi almak için: *Sürüm denetimi bütünleşmesi*.

Yeniden yerleştirme

Weblate işlemlerini güncelleme sırasında yukarı akış deposunun üstüne yerleştirir. Böylece, ek birleştirme işlemleri olmadan temiz geçmiş sağlanır.

Yeniden yerleştirme, karmaşık birleştirmeler olması durumunda sorun yaratabilir. Bu nedenle bunları etkinleştirmek isteyip istemediğinizi dikkatlice değerlendirin.

Özellikle farklı bir dala iterken *Git (itme dayatması ile)* seçeneğini *Sürüm denetimi sistemi* olarak seçerek itmeyi dayatmanız gerekebilir.

Birleştirme

Yukarı akış deposu değişiklikleri Weblate ile birleştirilir. Bu ayar yapılabildiğinde hızlı ileri atlamayı kullanır. Bu en güvenli yoldur ancak çok sayıda birleştirme işlemesi oluşturabilir.

Hızlı ilerleme olmadan birleştirme

Yukarı akış deposu değişiklikleri, her seferinde bir birleştirme işlemesi yapılarak Weblate üzerinde birleştirilir (hızlı ileri atlama yapılabilirse bile). Her Weblate değişikliği, Weblate deposunda bir birleştirme işlemi olarak görünür.

Varsayılan değer *DEFAULT_MERGE_STYLE* ile değiştirilebilir.

İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri

Bir çeviri işlenirken kullanılacak ileti. Bilgi almak için: `:ref: markup`.

Varsayılan değerler şu seçeneklerle değiştirilebilir *DEFAULT_ADD_MESSAGE*, *DEFAULT_ADDON_MESSAGE*, *DEFAULT_COMMIT_MESSAGE*, *DEFAULT_DELETE_MESSAGE*, *DEFAULT_MERGE_MESSAGE*, *DEFAULT_PULL_MESSAGE*.

İşleme ile itme

Gönderilen değişikliklerin otomatik olarak yukarı akış deposuna itilip itilmeyeceğini belirler. Etkinleştirildiğinde, Weblate değişiklikleri temel depoya kaydettikten sonra itme başlatılır (bilgi almak için: `ref:lazy-commit`). İtmeyi etkinleştirmek için *Depo itme adresi* ayarının da yapılandırılması gerekir.

İşlenecek değişikliklerin yaşı

Değişikliklerin arka plan görevi ya da *commit_pending* yönetim komutu tarafından uygulanmasından önce ne kadar eski (saat cinsinden) olması gerektiğini ayarlar. Bir bileşendeki tüm değişiklikler, bu dönemden daha eski en az bir değişiklik olduğunda gönderilir.

Varsayılan değer *COMMIT_PENDING_HOURS* seçeneği ile değiştirilebilir.

İpucu: Bekleyen değişikliklerin işlenebileceği başka durumlar da vardır. Bilgi almak için: *Lazy commit işlemleri*.

Hata durumunda kilitlenme

İlk başarısız itme ya da yukarı akış deposuna aktarma ile çekme sırasında bileşeni (ve bağlantılı bileşenleri, bilgi almak için: [İç Weblate adresleri](#)) kilitler. Böylece, el ile çözümlenmesi gereken başka çakışmaların oluşması engellenir.

Herhangi bir depo hatası kalmadığında bileşenin kilidi kendiliğinden açılır.

Kaynak dil

Kaynak dizgeleri için kullanılacak dil. İngilizce dışında bir dilden çeviri yapıyorsanız bu seçeneği değiştirin.

İpucu: İki dilli dosyaları İngilizce dilinden çevirdiğiniz halde, İngilizce çeviride de düzeltmeler yapabilmek istiyorsanız, kaynak dil adı ile var olan çeviri arasında çakışmayı önlemek için, kaynak dil olarak *İngilizce (Geliştirici)* seçeneğini seçin.

Bu durumda, tek dilli çeviriler için ara çeviri kullanabilirsiniz. Bilgi almak için: [Ara dil dosyası](#).

Dil süzgeci

Dosya maskesine göre tarama yapılırken çeviriyi süzmek için kullanılacak kurallı ifade. Weblate tarafından yönetilen dillerin listesini sınırlamak için kullanılabilir.

Not: Dil kodlarını dosya adında gördükleri şekilde listelemeniz gerekir.

Bazı süzme örnekleri:

Süzgeç açıklaması	Kurallı ifade
Yalnızca seçilmiş diller	<code>^(cs de es)\$</code>
Dilleri katmama	<code>^(?! (it fr) \$) .+\$</code>
Yalnızca iki harfli kodları süzme	<code>^[^.]\$</code>
Dil dosyası olmayanları katmama	<code>^(?! (blank) \$) .+\$</code>
Tüm dosyaları katma (varsayılan)	<code>^[^.] +\$</code>

Çeşitler kurallı ifadesi

Bir dizinin değişkenlerini belirlemek için kullanılacak kurallı ifade. Bilgi almak için: `:ref: variants`.

Not: Alanların çoğu, proje sahipleri veya yöneticiler tarafından Weblate arayüzünden düzenlenebilir.

Ayrıca bakınız:

Weblate, Git ve Mercurial dışındaki diğer sürüm denetimi sistemlerini destekliyor mu?, `alerts`

Öncelik

Yüksek öncelikli bileşenler çevirmenlere öncelikli olarak sunulur.

4.15 sürümünde değişti: Bu seçenek artık eşleşen sözlük terimlerinin sıralamasını da etkiliyor.

Kısıtlı erişim

Not: Bu özellik, Hosted Weblate üzerinde kullanılamaz.

Varsayılan olarak, bileşen, bileşende herhangi bir değişiklik yapamasa bile, projeye erişimi olan herkes tarafından görülebilir. Böylece proje içinde çeviri tutarlılığının korunması kolaylaşır.

Bir bileşene ya da bileşen listesi düzeyine erişimi kısıtlamak, proje düzeyi izinlerinden bağımsız olarak bir bileşene erişim iznini devralır. Erişim iznini açıkça vermeniz gerekir. Bunun için, yeni bir kullanıcı grubuna erişim izni verebilir ve kullanıcıları bu gruba yerleştirebilir ya da varsayılan *özel* ya da *kişisel* erişim denetimi gruplarını kullanabilirsiniz.

Varsayılan değer `DEFAULT_RESTRICTED_COMPONENT` seçeneği ile değiştirilebilir.

İpucu: Bu durum, proje yöneticileri için de geçerlidir. Lütfen durumu değiştirdikten sonra bileşene erişiminizi kaybetmeyeceğinizden emin olun.

Projelerde paylaşım

Bileşenin görünür olacağı ek projeler seçebilirsiniz. Bu özellik birkaç projede kullandığınız paylaşılan kitaplıklar için kullanışlıdır.

Not: Bir bileşenin paylaşılması, erişim denetimini değiştirmez. Yalnızca diğer projelere göz atarken görünmesini sağlar. Kullanıcıların göz atmak ya da çevirmek için gerçek bileşene erişebilmesi gerekir.

Sözlük olarak kullanılsın

4.5 sürümünde geldi.

Bu bileşenin sözlük olarak kullanılmasını sağlar. Nasıl listeleneceğini *Sözlük rengi* ile yapılandırabilirsiniz.

Sözlük, *Projelerde paylaşım* tarafından tanımlanan tüm projelerden erişilebilir olur.

Sözlüklere yeni sözcükler eklenmesine izin vermek için sözlüklerde *Dizge yönetimi* özelliğinin etkinleştirilmesi önerilir.

Ayrıca bakınız:

Sözlük

Sözlük rengi

Görüntülenen sözcüğün eşleştiği sözlüğün görüntülenme rengi.

2.7.5 Kalıp kodlaması

Weblate, metin işlemenin gerekli olduğu çeşitli yerlerde basit markup dili kullanır. [The Django template language](#) üzerine kurulu olduğundan oldukça güçlüdür.

Şu anda şuralarda kullanılabilir:

- İşleme iletisi biçimlendirmesi. Bilgi almak için: [Bileşen yapılandırması](#)
- **Çeşitli eklentiler**
 - [Bileşen keşfi](#)
 - [İstatistik oluşturma](#)
 - [Eklentiden betikleri çalıştırma](#)

Bileşen kalıplarında kullanılabilecek değişkenler şunlardır:

```
{{ language_code }}
```

Dil kodu

```
{{ language_name }}
```

Dil adı

```
{{ component_name }}
```

Bileşen adı

```
{{ component_slug }}
```

Bileşen kısaltması

```
{{ project_name }}
```

Proje adı

```
{{ project_slug }}
```

Proje kısaltması

```
{{ url }}
```

Çeviri adresi

```
{{ filename }}
```

Çeviri dosya adı

```
{{ stats }}
```

Çeviri istatistikleri, ek öznitelikleri vardır. Aşağıda örnekleri bulabilirsiniz.

```
{{ stats.all }}
```

Toplam dizge sayısı

```
{{ stats.fuzzy }}
```

Onaylanmayı bekleyen dizge sayısı

```
{{ stats.fuzzy_percent }}
```

Onaylanmayı bekleyen dizge yüzdesi

```
{{ stats.translated }}
```

Çevrilmiş dizge sayısı

```
{{ stats.translated_percent }}
```

Çevrilmiş dizge yüzdesi

```
{{ stats.allchecks }}
```

Denetim hatası olan dizgelerin sayısı

`{{ stats.allchecks_percent }}`
Denetim hatası olan dizgelerin yüzdesi

`{{ author }}`
Geçerli işleme sorumlusu. Yalnızca işleme kapsamında kullanılabilir.

`{{ addon_name }}`
Yalnızca eklenti işleme iletisinde kullanılabilen, işlenen geçerli eklentinin adı.

Depo tarayıcısında ya da düzenleyici kalıplarında kullanılabilecek değişkenler:

`{{branch}}`
geçerli dal

`{{line}}`
dosyadaki satır

`{{filename}}`
dosya adı, baştaki parçaları parentdir süzgecini kullanarak da çıkarabilirsiniz. Örneğin `{{filename|parentdir}}`

Bunları süzgeçlerle birleştirebilirsiniz:

```
{{ component|title }}
```

Koşullar kullanabilirsiniz:

```
{% if stats.translated_percent > 80 %}Well translated!{% endif %}
```

Karakterleri değiştirmek için kullanılabilecek ek bir etiket vardır:

```
{% replace component "-" " " %}
```

Bunu süzgeçlerle birleştirebilirsiniz:

```
{% replace component|capfirst "-" " " %}
```

Dosya adlarıyla değiştirmek için de ek bir süzgeç vardır:

```
Directory of a file: {{ filename|dirname }}
File without extension: {{ filename|striptext }}
File in parent dir: {{ filename|parentdir }}
It can be used multiple times: {{ filename|parentdir|parentdir }}
```

... ve diğer Django kalıbı özellikleri.

2.7.6 İçerik aktarma hızı

Sürüm denetimi sistemi deposunu almak ve çevirileri Weblate içine aktarmak, çevirilerinizin boyutuna bağlı olarak uzun bir süreç olabilir. İşte bazı ipuçları:

Yapılandırmayı iyileştirin

Varsayılan yapılandırma, Weblate denemeleri yapmak ve hata ayıklamak için kullanışlıdır. Üretim kurulumu için bazı farklı ayarlamalar yapmanız gerekir. Birçoğunun başarımlar üzerinde oldukça büyük bir etkisi vardır. Bilgi almak için: *Üretim kurulumu*, özellikle:

- Arka plan görevlerini yürütmek için Celery yapılandırın (bilgi almak için: *Celery ile arka plan görevlerini kullanmak*)
- *Ön bellek özelliğini açın*
- *Güç bir veri tabanı sunucusu kullanın*
- *Hata ayıklama kipini kapatın*

Kaynak sınırlarını denetleyin

Büyük çevirileri ya da depoları içe aktarıyorsanız, sunucunuzun kaynak sınırlamalarından etkilenebilirsiniz.

- Boş bellek miktarını denetleyin, çeviri dosyalarının işletim sistemi tarafından ön belleğe alınması başarımlar büyük ölçüde artırır.
- İşlenecek çok sayıda dizge varsa disk işlemleri darboğaz oluşturabilir. Disk hem Weblate hem de veri tabanı tarafından dayatılır.
- Ek işlemci çekirdekleri, arka plan görevlerinin başarımlarını artırmaya yardımcı olabilir (bilgi almak için *Celery ile arka plan görevlerini kullanmak*).

Gereksiz denetimleri devre dışı bırakın

Bazı kalite denetimleri için oldukça fazla kaynak gerekebilir ve gerekmeyen denetimler yok sayılırsa içe aktarım sırasında size biraz zaman kazandırabilir. Yapılandırma ile ilgili bilgi almak için: *CHECK_LIST*.

2.7.7 Bileşenleri kendiliğinden oluşturmak

Projenizde onlarca çeviri dosyası varsa (farklı gettext alanları veya Android uygulamalarının bölümleri gibi), bunları kendiliğinden içe aktarmak isteyebilirsiniz. Bunun için komut satırından *import_project* ya da *import_json* komutları kullanabilir ya da *Bileşen keşfi* eklentisini kurabilirsiniz.

Eklenti kullanmak için, önce bir çeviri dosyasına bir bileşen oluşturmanız (gelecekte yeniden adlandırılma ya da silinme olasılığı en düşük olanı seçin) ve eklenti bu bileşene kurmanız gerekir.

Yönetim komutlarını kullanmak için, tüm bileşenleri kapsayacak bir proje oluşturmanız ve ardından *import_project* ya da *import_json* komutunu yürütmeniz gerekir.

Ayrıca bakınız:

Yönetim komutları, *Bileşen keşfi*

2.8 Dil tanımları

Farklı çevirileri doğru bir şekilde sunmak için dil adı, metin yönü, çoğul tanımları ve dil kodu hakkındaki bilgiler gereklidir.

2.8.1 Hazır dil tanımları

Weblate üzerinde yaklaşık 600 dil için tanımlar bulunur ve bu liste her sürümde büyütülmüştür. Weblate her yükseltildiğinde (daha özel olarak **weblate migrate** komutu yürütüldüğünde, bilgi almak için: [Genel yükseltme yönergeleri](#)) diller veri tabanı, Weblate içinde bulunan tüm dil tanımlarını kapsayacak şekilde güncellenir.

Bu özellik `UPDATE_LANGUAGES` seçeneği ile devre dışı bırakılabilir. Ayrıca, `setuplang` seçeneğini kullanarak veri tabanının Weblate içindeki verilerle eşleşecek şekilde güncellenmesini dayatabilirsiniz.

Ayrıca bakınız:

[İç dil tanımlarını zenginleştirmek](#), Geçerli dil tanımları

2.8.2 Dil kodlarını işlemek

Weblate, çevirileri işlerken, dil kodunu (genellikle ISO 639-1 olanı) *Dosya maskesi* üzerinden var olan herhangi bir dil nesnesiyle eşleştirmeye çalışır.

Dil kod adları ile bu eşleştirmeyi proje düzeyinde daha ayrıntılı ayarlayabilirsiniz.

Tam eşleşme bulunamazsa, var olan dillerden en iyi uyanı ile eşleştirilmeye çalışılır. Şu adımlar denenir:

- Büyük/küçük harfe duyarlı olmadan bakılır.
- Alt çizgiler ve kısa çizgiler normalleştirilir.
- Yerleşik dil kod adlarına bakılır.
- Dil adına göre bakılır.
- Belirli bir dil için varsayılan ülke kodu yok sayılır. `cs_CZ` yerine `cs` seçilir.

Bunun da başarısız olması durumunda, varsayılan ayarlar kullanılarak (soldan sağa metin yönü, bir tane çoğul) yeni bir dil tanımı oluşturulur. `xx_XX` koduyla kendiliğinden oluşturulan dil `xx_XX` (*oluşturuldu*) olarak adlandırılır. Bunu daha sonra yönetici arayüzünde değiştirmek (bilgi almak için: [Dil tanımlarını değiştirmek](#)) ve sorun izleyiciye bildirmek (bilgi almak için: [Weblate için katkıda bulunmak](#)) isteyebilirsiniz. Böylece doğru tanımın gelecek Weblate sürümüne eklenmesini sağlayabilirsiniz.

İpucu: Dil olarak istenmeyen bir şey görürseniz, çevirileri işlerken bu türde bir dosyanın yok sayılması için *Dil süzgeci* seçeneğini ayarlamak isteyebilirsiniz.

Ayrıca bakınız:

[Dil kodu](#), adding-translation

2.8.3 Dil tanımlarını değiştirmek

Dil tanımlarını diller arayüzünden değiştirebilirsiniz (`/languages/` adresinden).

Düzenleme yaparken, tüm alanların doğru olduğundan emin olun (özellikle çoğullar ve metin yönü). Yoksa çevirmenler çevirileri olması gerektiği gibi düzenleyemez.

2.8.4 Belirsiz dil kodları ve makro diller

Çoğu durumda, çeviride makro dil kodu kullanmak iyi bir fikir değildir. Tipik bir soruna örnek, gerçek değişkene bağlı olarak Arapça ya da Latin alfabesiyle yazılmış olabilecek Kürtçe olabilir. Weblate üzerinde doğru davranışı elde etmek için, yalnızca tek tek dil kodlarını kullanmanız ve makro dillerden kaçınmanız önerilir.

Ayrıca bakınız:

[Makro diller tanımları](#), [Makro dillerin listesi](#)

2.8.5 Dil tanımları

Her dilde şu alanlar bulunur:

Dil kodu

Dili tanımlayan kod. Weblate, [ISO 639-1](#) ile tanımlanan iki harfli kodları kullanmayı yeğler. Ancak iki harfli kodu olmayan diller için [ISO 639-2](#) ya da [ISO 639-3](#) kodları da kullanılabilir. Ayrıca [BCP 47](#) tarafından tanımlanmış genişletilmiş kodlar da destelenir.

Ayrıca bakınız:

[Dil kodlarını işlemek](#), adding-translation

Dil adı

Dilin görüntülenecek adı. Weblate üzerinde bulunan dil adları da seçilmiş kullanıcı arayüzü diline göre yerleştirilmiş olarak görüntülenir.

Metin yönü

Dilin sağdan sola mı yoksa soldan sağa mı yazıldığını belirler. Bu özellik, dillerin çoğu için doğru şekilde kendiliğinden ayarlanır.

Çoğul sayısı

Dilde kullanılan çoğul biçim sayısı.

Çoğul formülü

Belirtilen bir sayı için hangi çoğul biçiminin kullanılacağını belirleyen gettext uyumlu çoğul formülü.

Ayrıca bakınız:

[Çoğul biçimler](#), GNU gettext araçları: Plural forms, Unikod Konsorsiyumu Dil Çoğul Biçim Kuralları

Konuşan sayısı

Dünyada bu dili konuşan kişi sayısı.

2.9 Sürekli yerelleştirme

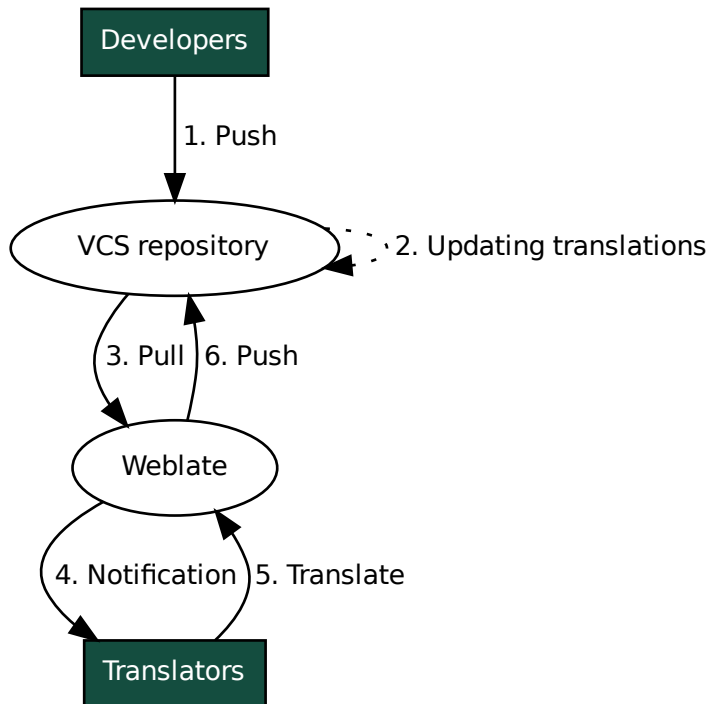
Çevirinizin gelişimi yakından izleyen hazır bir altyapı vardır. Böylece çevirmenler, yayın öncesinde büyük miktarda yeni metinler üzerinde çalışmak yerine, tüm zaman boyunca çeviriler üzerinde çalışabilirler.

Ayrıca bakınız:

/devel/integration geliştirme çalışmalarınızı Weblate ile bütünleştirmenin temel yollarını açıklar.

Süreç şu şekildedir:

1. Geliştiriciler değişiklikler yapar ve bunları sürüm denetimi deposuna gönderir.
2. İsteğe bağlı olarak çeviri dosyaları güncellenir (dosya biçimine bağlıdır, bilgi almak için: [Kalıbı güncellediğimde Weblate neden hala eski çeviri dizgelerini gösteriyor?](#)).
3. Weblate, sürüm denetimi sistemi deposundan değişiklikleri çeker, bilgi almak için: [Depoları güncellemek](#).
4. Weblate çevirilerdeki değişiklikleri algıladığında, çevirmenleri abonelik ayarlarına göre bilgilendirir.
5. Çevirmenler Weblate arayüzünü kullanarak çevirleri yapar ya da çevrimdışı yaptıkları değişiklikleri yükler.
6. Çevirmenlerin çalışması tamamlandıktan sonra, Weblate değişiklikleri yerel depoya gönderir (bilgi almak için: [Lazy commit işlemleri](#)) ve bunu yapma izni varsa bunları geri iter (bilgi almak için: [Weblate üzerindeki değişiklikleri itmek](#)).



2.9.1 Depoları güncellemek

Arka uç depolarını kaynaklarından güncellemek için bir yöntem ayarlamalısınız.

- Yaygın kullanılan kod barındırma hizmetlerinin çoğuyla bütünleştirmek için `:ref:` hooks' kullanın:
 - *GitHub değişikliklerini kendiliğinden almak*
 - *GitLab değişikliklerini kendiliğinden almak*
 - *Bitbucket değişikliklerini kendiliğinden almak*
 - *Pagure değişikliklerini kendiliğinden almak*
 - *Azure Repos değişikliklerini kendiliğinden almak*
 - *Gitea depoları değişikliklerini kendiliğinden almak*
- Depo yönetiminden ya da *Weblate REST API uygulaması* ya da *Weblate istemcisi* kullanarak güncellemeyi el ile başlatabilirsiniz
- Weblate kopyanızdaki tüm bileşenlerin kendiliğinden güncellenmesi için `AUTO_UPDATE` seçeneğini etkinleştirin
- `updategit` komutunu yürütün (projeyi seçin ya da tümünü “—güncellemek için `all` kullanın)

Weblate depoyu her güncellediğinde, güncelleme sonrası eklentileri tetiklenir. Bilgi almak için: *Eklentiler*.

Birleştirme çakışmalarından kaçınmak

Aynı dosya hem Weblate üzerinde hem de Weblate dışında değiştirildiğinde Weblate üzerinden gelen birleştirmelerde çakışmalar ortaya çıkar. Bu sorunu çözmek için kullanılabilecek iki yaklaşım vardır. Weblate dışındaki düzenlemelerden kaçınmak ya da güncelleme sürecinizi Weblate ile bütünleştirmek. Böylece Weblate dışındaki dosyalar güncellenmeden önce değişiklikler temizlenir.

Tek dilli dosyalar için ilk yaklaşım kolaydır. Weblate üzerinde yeni dizgeler ekleyebilir ve dosyaların tüm düzenleme işlemlerini orada yapabilirsiniz. İki dilli dosyalar için, kaynak kodundan çevrilebilir dosyalar oluşturmak için genellikle bir tür ileti ayıklama işlemi vardır. Bazı durumlarda bu işlem iki bölüme ayrılabilir. Birinci adım ayıklama kalıbını oluşturur (örneğin Gettext POT `xgettext` kullanılarak oluşturulur) ve ikinci adım onu gerçek çevirilerle birleştirir (Gettext PO dosyaları `msgmerge` kullanılarak güncellenir). İkinci adımı Weblate içinde yapabilirsiniz ve bu işlemden önce bekleyen tüm değişikliklerin katıldığına emin olabilirsiniz.

İkinci yaklaşım, *Weblate REST API uygulaması* ile Weblate uygulamasını bekleyen tüm değişiklikleri itmek ve kendi tarafınızda değişiklikler yaparken çeviriyi kilitlemek yoluyla uygulanabilir.

Güncelleme betiği şunun gibi görünebilir:

```
# 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
```

Aynı depoyu paylaşan birden fazla bileşeniniz varsa hepsini ayrı ayrı kilitlemeniz gerekir:

```
wlc lock foo/bar  
wlc lock foo/baz  
wlc lock foo/baj
```

Not: Örnekte, uzaktan Weblate yönetimi için yapılandırmaya (API anahtarları) gerek duyan *Weblate istemcisi* kullanılır. Bunu wlc yerine curl gibi herhangi bir HTTP istemcisini kullanarak da yapabilirsiniz. Bilgi almak için: *Weblate REST API uygulaması*.

Ayrıca bakınız:

Weblate istemcisi

GitHub değişikliklerini kendiliğinden almak

Weblate doğal GitHub desteği ile gelir.

Hosted Weblate kullanıyorsanız, *Weblate uygulaması* kurmanız önerilir. Böylece çok fazla şeyi ayarlamanız gerekmeden doğru kurulumu elde edersiniz. Değişiklikleri geri itmek için de kullanılabilir.

GitHub deposuna yapılan her itmede bildirim almak için, depo ayarlarına (*Webhooks*) aşağıdaki görseldeki gibi Weblate internet kancasını ekleyin:

The screenshot shows the GitHub Webhooks settings page for the 'hello' repository. The page has a sidebar with links to 'Options', 'Collaborators & teams', 'Branches', 'Webhooks', 'Integrations & services', 'Deploy keys', and 'Alerts'. The 'Webhooks' section is active. The main content area shows the 'Add webhook' form. It includes a 'Payload URL' field with the value 'https://hosted.weblate.org/hooks/github/'. The 'Content type' is set to 'application/x-www-form-urlencoded'. There is a 'Secret' field. Below the secret field, there is a checkbox for 'Active' which is checked, and a note 'We will deliver event details when this hook is triggered.' At the bottom of the form is a green 'Add webhook' button. The footer of the page shows the GitHub logo and links to 'Contact GitHub', 'API', 'Training', 'Shop', 'Blog', and 'About'.

Yük adresi olarak, Weblate adresinizin sonuna “/hooks/github/” ekleyin. Örneğin Hosted Weblate hizmeti için “https://hosted.weblate.org/hooks/github/” kullanabilirsiniz.

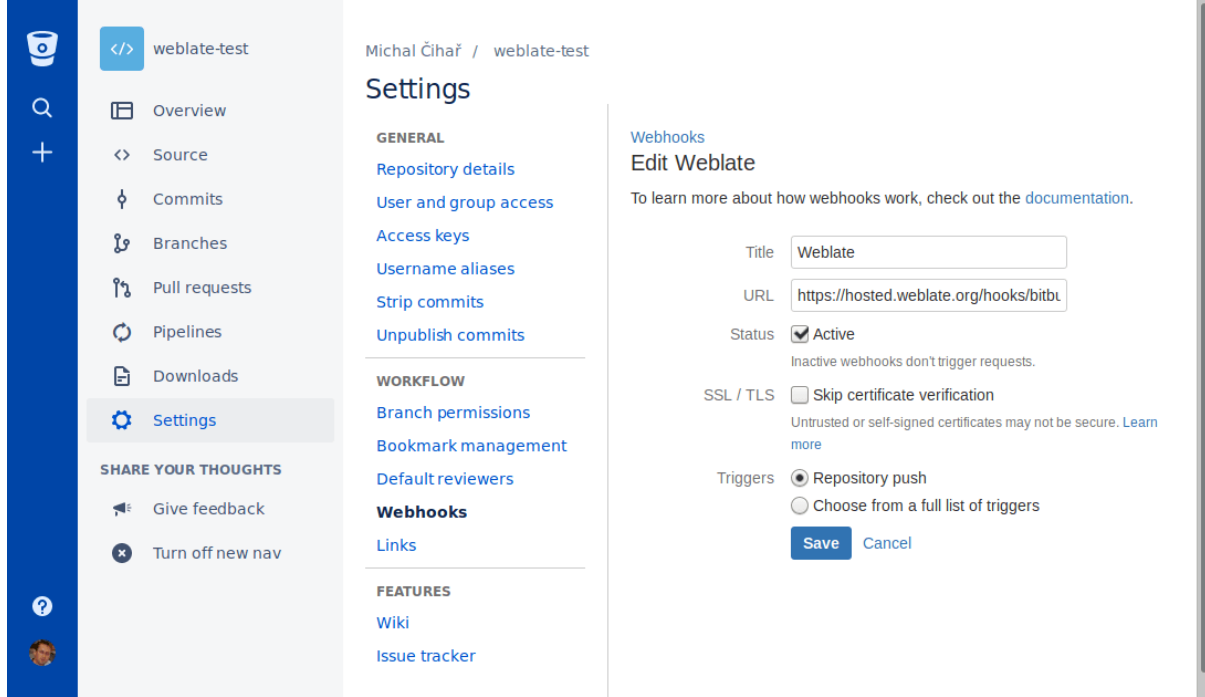
Diğer ayarları varsayılan değerlerinde bırakabilirsiniz (Weblate her iki içerik türünü de işleyebilir ve yalnızca *push* işlemine gerek duyar).

Ayrıca bakınız:

POST /hooks/github/, Hosted Weblate üzerinden depolara erişmek

Bitbucket değişikliklerini kendiliğinden almak

Weblate, Bitbucket internet kancalarını destekler. Weblate kurulumunuza hedef olarak `/hooks/bitbucket/` adresiyle depo itme işlemi sırasında tetiklenecek bir internet kancası ekleyin (`https://hosted.weblate.org/hooks/bitbucket/` gibi).

**Ayrıca bakınız:**

POST /hooks/bitbucket/, Hosted Weblate üzerinden depolara erişmek

GitLab değişikliklerini kendiliğinden almak

Weblate, GitLab kancalarını destekler. Weblate kurulumunuza hedef olarak `/hooks/gitlab/` adresiyle hedefi bir proje internet kancası ekleyin (`https://hosted.weblate.org/hooks/gitlab/` gibi).

Ayrıca bakınız:

POST /hooks/gitlab/, Hosted Weblate üzerinden depolara erişmek

Pagure değişikliklerini kendiliğinden almak

3.3 sürümünde geldi.

Weblate, Pagure kancalarını destekler. Weblate kurulumunuza hedef olarak `/hooks/pagure/` internet kancasını ekleyin (`https://hosted.weblate.org/hooks/pagure/` gibi). Bu işlem, *Proje ayarları* bölümündeki *Web kancaları kullanılsın* seçeneği ile yapılabilir:

The screenshot shows the Weblate interface for a project named 'nijel-test'. The top navigation bar includes 'Browse', 'Create', and a user profile icon. Below the project name, there are links for 'New Issue', 'Open PR', 'Fork', and 'Clone'. The main navigation menu on the left lists various settings categories: Project Settings, Project Details, Default Branch, Private Web Hook Key, API Keys, Project Options (selected), Public Notifications, Users & Groups, Deploy Keys, Hooks, Priorities, Roadmap, Close Status, Custom Issue Fields, Reports, Tags, Quick Replies, Regenerate Repos, Give Project, and Delete Project. The 'Project Options' section is active, displaying a list of checkboxes for various features. Most are unchecked, but 'Activate fedmsg notifications', 'Activate Issue tracker', 'Activate pull requests', and 'Activate stomp notifications' are checked. There is a text input field for 'Activate Minimum score to merge pull-request' with the value '-1'. Below the checkboxes, there is a section for 'Activate Web-hooks' with a text input field containing 'https://hosted.weblate.org/hooks/pagure/'. There are 'Update' and 'Test web-hook' buttons. At the bottom, there is a 'Learn more about' section with links to 'Flags', 'Tracker read-only', 'Pull-request access only', 'Roadmap on Issue page', and 'fedmsg notifications'.

Ayrıca bakınız:

POST /hooks/pagure/, *Hosted Weblate üzerinden depolara erişmek*

Azure Repos değişikliklerini kendiliğinden almak

3.8 sürümünde geldi.

Weblate, Azure Repos kancalarını destekler. Weblate kurulumunuza *İtilecek kod* işlemi için hedef olarak `/hooks/azure/` internet kancasını ekleyin (`https://hosted.weblate.org/hooks/azure/` gibi). Bu işlem, *Proje ayarları* bölümündeki *Hizmet kancaları* seçeneği ile yapılabilir.

Ayrıca bakınız:

Azure DevOps rehberinde internet kancaları, *POST /hooks/azure/*, *Hosted Weblate üzerinden depolara erişmek*

Gitea depoları değişikliklerini kendiliğinden almak

3.9 sürümünde geldi.

Weblate, Gitea kancalarını destekler. Weblate kurulumunuza *İtme işlemleri* içinden *Gitea internet kancası* işlemi için hedef olarak `/hooks/gitea/` internet kancasını ekleyin (<https://hosted.weblate.org/hooks/gitea/> gibi). Bu işlem, *Ayarlar* bölümündeki *Web kancaları* seçeneği ile yapılabilir.

Ayrıca bakınız:

Gitea rehberinde internet kancaları, *POST /hooks/gitea/*, *Hosted Weblate üzerinden depolara erişmek*

Gitee depoları değişikliklerini kendiliğinden almak

3.9 sürümünde geldi.

Weblate, Gitee kancalarını destekler. Weblate kurulumunuza *İtme* işlemi için hedef olarak `/hooks/gitee/` internet kancasını ekleyin (<https://hosted.weblate.org/hooks/gitee/> gibi). Bu işlem, *Yönetim* bölümündeki *Web kancaları* seçeneği ile yapılabilir.

Ayrıca bakınız:

Gitee rehberinde internet kancaları, *POST /hooks/gitee/*, *Hosted Weblate üzerinden depolara erişmek*

Depoları her gece kendiliğinden güncellemek

Weblate, daha sonra değişiklik birleştirme başarımını artırmak için her gece uzak depoları kendiliğinden alır. İsteğe bağlı olarak, *AUTO_UPDATE* seçeneğini etkinleştirerek bunu gecelik birleştirmeler yapmaya da dönüştürebilirsiniz.

2.9.2 Weblate üzerindeki değişiklikleri itmek

Her çeviri bileşeni için ayrı bir itme adresi ayarlanabilir (bilgi almak için *Depo itme adresi*) ve bu durumda Weblate, değişikliği uzak depoya itebilir. Weblate, değişiklikleri her işlemede otomatik olarak gönderecek şekilde de yapılandırılabilir (varsayılan davranış, bilgi almak için: *İşleme ile itme*). Değişikliklerin kendiliğinden itilmesini istemiyorsanız, bunu el ile *Depo bakımı* bölümünden ya da *wlc push* API seçeneğini kullanarak yapabilirsiniz.

İtme seçenekleri, kullanılan *Sürüm denetimi bütünleştirmesi* değerine göre farklılık gösterir. Ayrıntılı bilgileri bu bölümden alabilirsiniz.

İtme işleminin doğrudan Weblate tarafından yapılmasını istemiyorsanız, *GitHub çekme istekleri*, *GitLab birleştirme istekleri*, *Gitea çekme isteği*, *Pagure birleştirme istekleri* çekme istekleri ya da *Gerrit* onayları desteklenmektedir. Bunları *Bileşen yapılandırması* içindeki *Sürüm denetimi sistemi* bölümünden *GitHub*, *GitLab*, *Gitea*, *Gerrit* ya da *Pagure* olarak seçerek etkinleştirebilirsiniz.

Genel olarak, Git, GitHub ve GitLab ile şu seçenekler kullanılabilir:

İstenilen kurulum	Sürüm denetimi sistemi	Depo itme adresi	İtme işleminin yapılacağı dal
İtme yok	<i>Git</i>	<i>empty</i>	<i>empty</i>
Doğrudan itme	<i>Git</i>	SSH adresi	<i>empty</i>
Ayrı bir dala it	<i>Git</i>	SSH adresi	Dal adı
GitHub dalından çekme isteği	<i>GitHub çekme istekleri</i>	<i>empty</i>	<i>empty</i>
GitHub dalına itme isteği	<i>GitHub çekme istekleri</i>	SSH URL ¹	Dal adı
GitLab dalından birleştirme isteği	<i>GitLab birleştirme istekleri</i>	<i>empty</i>	<i>empty</i>
GitLab dalından birleştirme isteği	<i>GitLab birleştirme istekleri</i>	SSH URL <small>Sayfa 298, 1</small>	Dal adı
Gitea çatalından birleştirme isteği	<i>Gitea çekme isteği</i>	<i>empty</i>	<i>empty</i>
Gitea dalından birleştirme isteği	<i>Gitea çekme isteği</i>	SSH URL ¹	Dal adı
Pagure çatalından birleştirme isteği	<i>Pagure birleştirme istekleri</i>	<i>empty</i>	<i>empty</i>
Pagure dalından birleştirme isteği	<i>Pagure birleştirme istekleri</i>	SSH URL ¹	Dal adı

Not: Weblate işledikten sonra değişikliklerin kendiliğinden gönderilmesini de etkinleştirebilirsiniz. Bu işlem *İşleme ile itme* içinden yapılabilir.

Ayrıca bakınız:

SSH anahtarlarını ayarlamak için *Depolara erişmek* ve değişikliklerin Weblate tarafından ne zaman işleneceğine karar verildiği ile ilgili bilgi almak için `:ref:'lazy-commit'` bölümlerine bakabilirsiniz.

Korunmuş dallar

Weblate ile korumalı dal kullanıyorsanız, çekme isteklerini kullanacak ve çeviriler üzerinde gerçek gözden geçirme yapacak bir yapılandırma ayarlayabilirsiniz (bilmediğiniz diller için sorunlu olabilecek şeyler). Alternatif olarak, Weblate itme kullanıcısı için bu sınırlamayı kaldırabilirsiniz.

Örneğin bu işlem GitHub üzerinde, depo yapılandırmasında ayarlanabilir:

¹ *Kaynak kod deposu* itmeyi destekliyorsa boş olabilir.

☒ **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 Diğerleri ile etkileşim

Weblate, API uygulaması -başkalarıyla etkileşim kurmayı kolaylaştırır.

Ayrıca bakınız:

Weblate REST API uygulaması

2.9.4 Lazy commit işlemleri

Weblate, olabiliyorsa aynı yazardan gelen işlemleri tek bir işleme olarak gruplandırarak biçimde davranır. Böylece, işleme sayısı büyük ölçüde azaltılır. Bununla birlikte, sürüm denetimi sistemi deposunu eşitlemek isterseniz bunu açıkça belirtmeniz gerekir. Örneğin birleştirme için (varsayılan olarak *Yöneticiler* grubu için izin verilir, bilgi almak için: `:ref:privileges`).

Bu kipteki değişiklikler, aşağıdaki koşullardan herhangi biri yerine getirildiğinde işlenir:

- Başka biri zaten değiştirilmiş bir dizgeyi değiştirdiğinde.
- Yukarı akıştan bir birleştirme gerçekleştirildiğinde.
- Açık bir işleme isteği yapıldığında.
- Bir dosyanın indirilmesi istendiğinde.
- Değişiklik, *Bileşen yapılandırması* üzerinde *İşlenecek değişikliklerin yaşı* olarak tanımlanmış dönemden daha eski olduğunda.

İpucu: İşlemler her bileşen için ayrı oluşturulur. Bu nedenle, birçok bileşenin varsa, gene çok sayıda işleme görürsünüz. Bu durumda *Git işlemlerini bir araya toplar* eklentisini kullanabilirsiniz.

Değişikliklerin daha sık ve yaşları denetlemeden yapılmasını istiyorsanız, işlemeyi yapacak bir zamanlanmış görev ayarlayabilirsiniz. Bu işlemi *Django yönetim arayüzü* içinde *Zamanlanmış görevler* bölümünden yapabilirsiniz. Önce istediğiniz *Sıklık* ögesini oluşturun (120 saniye gibi). Ardından yeni bir zamanlanmış görev ekleyin ve *Görev* olarak `weblate.trans.tasks.commit_pending`, *Anahtar sözcük parametreleri* ve istenilen sıklık olarak `{"hours": 0}` yazın.

2.9.5 Betikleri kullanarak depo işlemleri yapmak

Weblate ile depo arasındaki etkileşim *Eklentiler* ile özelleştirilebilir. Eklentiler ile dış betiklerin nasıl yürütüleceği ile ilgili ayrıntılı bilgi almak için *Eklentiden betikleri çalışma* bölümüne bakabilirsiniz.

2.9.6 Bileşenler arasında çevirilerin tutarlılığını sağlamak

Birden çok çeviri bileşeniniz olduğunda, aynı dizgelerin çevirilerinin de aynı olduğundan emin olmak isteyebilirsiniz. Bu tutarlılık birkaç düzeyde sağlanabilir.

Çevirilerin yayılmasını sağlamak

Çevirilerin yayılmasını sağlamak seçeneği etkinken (varsayılan değer nedir, bilgi almak için: *Bileşen yapılandırması*), tüm yeni çeviriler dizgeleri eşleşen tüm bileşenlerde kendiliğinden yapılır. Bu tür çeviriler, tüm bileşenlerde çeviriyi yapan geçerli kullanıcının hesabına yazılır.

Not: Çeviri yayılması için, anahtarın tek dilli çeviri biçimleriyle eşleşmesi gerekir. Çeviri anahtarlarını oluştururken bunu aklınızda bulundurun.

Tutarlılık denetimi

Dizgeler farklı olduğunda *Tutarsız* denetimi tetiklenir. Bu tür farklılıkları el ile incelemek ve doğru çeviriyi seçmek için bunu kullanabilirsiniz.

Kendiliğinden çeviri

Farklı bileşenleri temel alan kendiliğinden çeviri, çevirileri bileşenler arasında eşitlemenin bir yöntemi olabilir. El ile tetikleyebilir (bilgi almak için: *Kendiliğinden çeviri*) veya eklentiye kullanarak depo güncellemelerinde kendiliğinden çalışmasını sağlayabilirsiniz (bilgi almak için: *Kendiliğinden çeviri*).

2.10 Çeviri lisanslarını belirlemek

Çevirmenlerin katkıda bulunmak için kabul etmesi gereken lisans koşullarını belirleyebilirsiniz. Bu özellik, herkese açık olan çevirilerin ne amaçla kullanılabileceğini belirlemek özellikle önemlidir.

Bileşen yapılandırması lisans bilgilerini belirtmelisiniz. Yapılabiliyor olsa da, bir katılımcı lisans anlaşmasını zorunlu kılmaktan kaçının.

2.10.1 Lisans bilgileri

Lisans bilgileri (lisans adı ve adresi) belirtildikten sonra, bu bilgiler ilgili *Bileşen yapılandırması* için çeviri bilgileri bölümünde görüntülenir.

Genellikle açık bir onay gerekmiyorsa lisans bilgilerini yayınlamak için en iyi yer burasıdır. Projeniz veya çeviriniz özgür değilse, büyük olasılıkla önceden izin almanız gerekir.

2.10.2 Katılımcı anlaşması

Bir katılımcı lisans anlaşması belirtirseniz, çeviriye yalnızca bunu kabul eden kullanıcılar katkıda bulunabilir. Katılımcı anlaşması çeviriye erişirken açıkça görülebilir:


The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, a breadcrumb shows 'WeblateOrg / Language names' and a 'translated 95%' badge. A yellow banner states: 'Contribution to this translation requires you to agree with a contributor agreement.' with a 'View contributor agreement' button. Below this is a tabbed interface with 'Languages' selected. It shows a table of languages with columns: Language, Translated, Unfinished, Unfinished words, Unfinished characters, Checks, Suggestions, and Comments. The table lists Czech, Hebrew, Hungarian, and English. Hungarian shows 81% translated, 4 unfinished items, 5 unfinished words, and 32 unfinished characters. At the bottom, there's a 'Start new translation' button and a footer with 'Powered by Weblate 4.17' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Language	Translated	Unfinished	Unfinished words	Unfinished characters	Checks	Suggestions	Comments
Czech	✓						
Hebrew	✓						
Hungarian	81%	4	5	32			
English	✓						

Yazılan metinde paragraflar kullanılabilir ve dış bağlantılar eklenebilir. HTML kod imleri kullanılamaz.

2.10.3 Kullanıcı lisansları

Herhangi bir kullanıcı, bulut sunucusunda herkese açık tüm projelerin tüm çeviri lisanslarını profilinde inceleyebilir:

 Weblate Dashboard Projects Languages Checks

Your profile

Languages Preferences Notifications Account Profile Teams Licenses Audit log API access

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 4.17 About Weblate Legal Contact Documentation Donate to Weblate

2.11 Çeviri süreci

2.11.1 Öneriler oylanabilsin

Varsayılan olarak herkes, oturum açmış kullanıcılar tarafından değerlendirilebilecek öneriler ekleyebilir. Öneri oylama özelliği açılarak oturum açmış birkaç kullanıcının bir dizgeyi değerlendirmesi sağlanabilir. `.ref:'component'` için *Öneriler oylanabilsin* seçeneğini açmak ve önerilerin onaylanması için bir eşik belirlemek üzere *Öneriler kendiliğinden onaylansın* (bu, önerilmişse öneriyi yapan kullanıcının oyunu da içerir) seçeneği ayarlanabilir.

Not: Kendiliğinden onaylama ayarlandıktan sonra, normal kullanıcılar çevirileri doğrudan kaydetme ya da önerileri onaylama yetkisini kaybeder. Bu, *Öneriler dayatıldığında dizge düzenlenebilsin* <privileges> izni ile değiştirilebilir.

Bunları *erişim denetimi* ile şu kurulumlardan birinde birleştirebilirsiniz:

- Kullanıcılar öneride bulunur ve öneriler için oy verir. Sınırlı bir grup neyin onaylanacağını denetler. - Oylamayı açın. - Kendiliğinden onaylamayı kapatın. - Kullanıcıların çevirileri kaydetmesine izin vermeyin.
- Kullanıcılar, belirtilen sayıda onay verdikten sonra öneri kendiliğinden onaylanır. - Oylamayı açın. - Kendiliğinden onay için istenilen oy sayısını ayarlayın.
- Öneriler için isteğe bağlı oylama. (İsteğe bağlı olarak, kullanıcılar bir çeviriden emin olmadıklarında birkaç öneride bulunabilir) - Yalnızca oylamayı açın.

2.11.2 Kaynak dizgeler ile ilgili ek bilgiler

Dizgelere açıklamalar, dizge öncelikleri, denetim imleri ve görsel bağlam gibi ek bilgiler ekleyerek çeviri işlemini geliştirin. Bu bilgilerin bazıları çeviri dosyalarından çıkarılabilir ve bazıları ek dizge bilgileri düzenlenerek eklenebilir:

The screenshot displays the Weblate web interface. A modal dialog titled "Edit additional string info" is open in the center. The dialog contains the following sections:

- Explanation**: A text input field with a placeholder "Additional explanation to clarify meaning or usage of the string."
- Labels**: Two checkboxes labeled "Current sprint" and "Next sprint". Below them, a note states: "Additional labels can be defined in the [project settings](#)."
- Translation flags**: A text input field with a placeholder "Additional comma-separated flags to influence Weblate behavior."
- Inherited flags**: A section showing "java-printf-format" is inherited from the Android String Resource.

The background interface shows the Weblate dashboard for a project named "Android" in the "English" language. It includes a list of source strings, a sidebar with various tools and information, and a footer with navigation links.

Bunlara doğrudan çeviri arayüzünden *Ekran görüntüsü bağlamı* ya da *İmler* yanındaki “Düzenle” simgesine tıklayarak erişebilirsiniz.

Powered by [Weblate 4.17](#) [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Dizge önceliklendirme

Öncelik imi kullanılarak daha yüksek önceliği olan dizgelerin daha önce çevrilmek üzere sunulması sağlanabilir.

İpucu: Bu seçenek, çeviri akışını belirli bir mantığa göre sıralamak için kullanılabilir.

Ayrıca bakınız:

Kalite denetimleri

Çeviri imleri

3.3 sürümünde değişti: Daha önce *Kalite denetimi imleri* olarak adlandırılan bu im, artık yalnızca denetimleri yapılandırmıyor.

Kalite denetimlerinin ve diğer Weblate davranışlarının özelleştirilmesi. Bilgi almak için: *Parametreleri kullanarak davranışı özelleştirmek*.

Dizge imleri *Bileşen yapılandırması* içindeki *Çeviri imleri* ögesinden ve çeviri dosyasındaki imlerden alınır (bilgi almak için: *Desteklenen dosya biçimleri*).

Ayrıca bakınız:

Kalite denetimleri, Parametreleri kullanarak davranışı özelleştirmek

Açıklama

4.1 sürümünde değişti: Önceki sürümlerde buna *Ek bağlam* deniyordu.

Çevirinin kapsamını veya kullanımını netleştirmek için açıklamayı kullanın. Bağlantıları ve diğer kod imlerini eklemek için Markdown kullanabilirsiniz.

Dizgeler için görsel bağlam

Uygulamanızda kullanılan belirli bir kaynak dizgeyi gösteren bir ekran görüntüsü yükleyebilirsiniz. Bu seçenek, çevirmenlerin dizgenin nerede kullanıldığını ve nasıl çevrilmesi gerektiğini anlamalarına yardımcı olur.


Yüklenen ekran görüntüsü, çeviri bağlamı yan çubuğunda görüntülenir:

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, the breadcrumb 'WeblateOrg / Django / Czech / Translate' is visible. The main area is titled 'Translation' and shows a comparison between English and Czech versions of a string. The English string is 'Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.' 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.' There are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. Below the main area, there are tabs for 'Nearby strings', 'Comments', 'Automatic suggestions', 'Other languages', and 'History'. At the bottom, there's a 'Translation memory' section with a search bar. On the right side, there's a sidebar with several sections: 'Glossary' (showing terms like 'machine translation' and 'project'), 'String information' (containing 'Screenshot context' with a screenshot upload button, 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file').

Powered by Weblate 4.17 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Kaynak dizgeler ile ilgili ek bilgiler ögesine ek olarak, ekran görüntülerinin *Araçlar* menüsü altında ayrı bir yönetim bölümü vardır. Ekran görüntülerini yükleyebilir, bunları el ile kaynak dizgelerle ilişkilendirebilir ya da optik karakter tanımadan yardım alabilirsiniz.

Bir ekran görüntüsü yüklendikten sonra, bu bölümden yönetimi ve kaynak dizge ile ilişkilendirilmesi yapılabilir:

 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 Denetimler ve düzeltmeler

2.12.1 Özel kendiliğinden düzeltmeler

Standart olanların yanında özel kendiliğinden düzeltmeleri de uygulayabilir ve bunları `:setting: AUTOFIX_LIST` içine ekleyebilirsiniz.

Kendiliğinden düzeltmeler güçlüdür, ancak bozulmaya da neden olabilir; Bunları yazarken dikkatli olun.

Örneğin, şu kendiliğinden düzeltme, bir çevirideki `foo` dizgesinin her kopyasını `bar` ile değiştirir:

```
# 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
```

Özel denetimler kurmak için, `AUTOFIX_LIST` içinde Python sınıfının tam yolunu belirtin. Bilgi almak için: `:ref: custom-check-modules`.

2.12.2 Parametreleri kullanarak davranışı özelleştirmek

İmleri kullanarak Weblate davranışını ince ayarlayabilirsiniz. Bu işlem kaynak dizge düzeyinde (bilgi almak için: *Kaynak dizgeler ile ilgili ek bilgiler*) ya da *Bileşen yapılandırması* (*Çeviri imleri*) içinde yapılabilir. Bazı dosya biçimleri, imlerin doğrudan biçimde belirtilmesini de sağlar (bilgi almak için: *Desteklenen dosya biçimleri*).

İmler virgül ile ayrılır. Parametreler iki nokta üst üste ile ayrılır. Dizgeye boşluk veya özel karakterler eklemek için tırnak işaretleri kullanabilirsiniz. Örneğin:

```
placeholders:"special:value":"other value", regex:.*
```

Hem tek hem de çift tırnak işaretleri kabul edilir. Ters bölü kullanılarak özel karakterlerden kaçılır:

```
placeholders:"quoted \"string\"":'single \'quoted\''
```

Kullanılabilecek imlerin listesi şu şekildedir:

rst-text

Bir metnin reStructuredText belgesi olarak işlenmesini sağlar. *Çeviri değiştirilmemiş* parametresini etkiler.

dos-eol

Unix satır sonu karakterleri yerine DOS karakterlerinin kullanılmasını sağlar (`\n` yerine `\r\n`).

read-only

Dizgenin salt okunur olduğunu ve Weblate üzerinde düzenlenemeyeceğini belirler. Bilgi almak için: *Salt okunur dizgeler*.

priority:N

Dizgenin önceliğini belirler. Yüksek öncelikli dizgeler daha çevrilir. Varsayılan öncelik değeri: 100. Bir dizgenin önceliği ne kadar yüksekse, o kadar önce çevrilmek üzere sunulur.

max-length:N

Bir dizgenin en fazla uzunluğunu N karakter olarak sınırlandırır. Bilgi almak için: [:ref:~check-max-length](#).

xml-text

Metnin XML belgesi olarak işlenmesini sağlar. *XML söz dizimi* ve *XML kodlaması* parametrelerini etkiler.

font-family:NAME

Görselleştirme denetimleri için yazı tipi ailesini belirler. Bilgi almak için: *Yazı tiplerini yönetmek*.

font-weight:WEIGHT

Görselleştirme denetimleri için yazı koyuluğunu belirler. Bilgi almak için: *Yazı tiplerini yönetmek*.

font-size:SIZE

Görselleştirme denetimleri için yazı boyutunu belirler. Bilgi almak için: *Yazı tiplerini yönetmek*.

font-spacing:SPACING

Görselleştirme denetimleri için harf aralığını belirler. Bilgi almak için: [:ref:~fonts](#).

icu-flags:FLAGS

ICU MessageFormat kalite denetiminin davranışını özelleştirecek imleri belirler.

icu-tag-prefix:PREFIX

check-java-messageformat kalite denetimi için XML kod imlerinin gerekli olan ön ekini belirler.

placeholders:NAME:NAME2:...

Çeviride beklenen yer belirtici dizgeleri belirler. Bilgi almak için: [:ref:~check-placeholders](#).

replacements:FROM:TO:FROM2:TO2...

Ortaya çıkan metin parametreleri denetlenirken yapılacak değişiklikleri belirler (*Çevirinin en fazla uzunluğu* ya da *Çevirinin en fazla uzunluğu* gibi). Tipik kullanım örneği, metnin uzun değerlerle bile sığıdığından emin olmak için yerleştirilebilir öğelerin genişletilmesidir. Örnek: `replacements:%s:"Ali Kaya"`.

variants:SOURCE

Bu dizgenin, eşleşen kaynağın bir dizge çeşidi olarak imlenmesini sağlar. Bilgi almak için: [:ref:~variants](#).

regex:REGEX

Çevirilerin eşleştirileceği kurallı ifadeyi belirtir. Bilgi almak için: *Kurallı ifade*.

forbidden

Bir sözlükte yasaklanmış bir çeviriyi belirtir. Bilgi almak için: *Yasaklı çeviriler*.

strict-same

“Değiştirilmemiş çeviri” içinde iç kara listedeki sözcüklerin kullanmasını engeller. Bilgi almak için: *Çeviri değiştirilmemiş*.

check-glossary

Sözlük ile uyumlu değil kalite denetimini etkinleştirir.

angularjs-format

AngularJS değer ekleme dizgesi kalite denetimini etkinleştirir.

c-format

C biçimi kalite denetimini etkinleştirir.

c-sharp-format

C# biçimi kalite denetimini etkinleştirir.

es-format

ECMAScript kalıbı sabitleri kalite denetimini etkinleştirir.

i18next-interpolation

i18next değer eklemesi kalite denetimini etkinleştirir.

icu-message-format

ICU MessageFormat kalite denetimini etkinleştirir.

java-printf-format

Java biçimi kalite denetimini etkinleştirir.

java-format

Java MessageFormat kalite denetimini etkinleştirir.

javascript-format

JavaScript biçimi kalite denetimini etkinleştirir.

lua-format

Lua biçimi kalite denetimini etkinleştirir.

object-pascal-format

Object Pascal biçimi kalite denetimini etkinleştirir.

percent-placeholders

Yüzde yer belirticileri kalite denetimini etkinleştirir.

perl-format

Perl biçimi kalite denetimini etkinleştirir.

php-format

PHP biçimi kalite denetimini etkinleştirir.

python-brace-format

Python parantez biçimi kalite denetimini etkinleştirir.

python-format

Python biçimi kalite denetimini etkinleştirir.

qt-format

Qt biçimi kalite denetimini etkinleştirir.

qt-plural-format

Qt çoğul biçimi kalite denetimini etkinleştirir.

ruby-format

Ruby biçimi kalite denetimini etkinleştirir.

scheme-format

Şema biçimi kalite denetimini etkinleştirir.

vue-format

Vue 118n biçimlendirmesi kalite denetimini etkinleştirir.

md-text

Metnin Markdown belgesi olarak işlenmesini sağlar. *Markdown bağlantıları*, *Markdown başvuruları* ve *Markdown söz dizimi* kalite denetimlerini etkinleştirin.

case-insensitive

Denetim davranışını büyük/küçük harfe duyarlı olmayacak şekilde ayarlar. Şu anda yalnızca *Yer belirticiler* kalite denetimini etkiler.

safe-html

Güvenli olmayan HTML kalite denetimini etkinleştirir.

url

Dizge yalnızca bir adresten oluşmalıdır. *Adres* kalite denetimini etkinleştirin.

ignore-all-checks

Tüm kalite denetimlerini yok sayar.

ignore-bbcode

BBCode kodlaması kalite denetimini atlar.

ignore-duplicate

Art arda yinelenen sözcükler var kalite denetimini atlar.

ignore-check-glossary

Sözlük ile uyumlu değil kalite denetimini atlar.

ignore-double-space

Çift boşluk kalite denetimini atlar.

ignore-angularjs-format

AngularJS değer ekleme dizgesi kalite denetimini atlar.

ignore-c-format

C biçimi kalite denetimini atlar.

ignore-c-sharp-format

C# biçimi kalite denetimini atlar.

ignore-es-format

ECMAScript kalıbı sabitleri kalite denetimini atlar.

ignore-i18next-interpolation

check-i18next-format kalite denetimini atlar.

ignore-icu-message-format

ICU MessageFormat kalite denetimini atlar.

ignore-java-format

Java MessageFormat kalite denetimini atlar.

ignore-java-printf-format

Java biçimi kalite denetimini atlar.

ignore-javascript-format

JavaScript biçimi kalite denetimini atlar.

ignore-lua-format

Lua biçimi kalite denetimini atlar.

ignore-object-pascal-format

Object Pascal biçimi kalite denetimini atlar.

ignore-percent-placeholders

Yüzde yer belirtecileri kalite denetimini atlar.

ignore-perl-format

Perl biçimi kalite denetimini atlar.

ignore-php-format

PHP biçimi kalite denetimini atlar.

ignore-python-brace-format

Python parantez biçimi kalite denetimini atlar.

ignore-python-format

Python biçimi kalite denetimini atlar.

ignore-qt-format

Qt biçimi kalite denetimini atlar.

ignore-qt-plural-format

Qt çoğul biçimi kalite denetimini atlar.

ignore-ruby-format

Ruby biçimi kalite denetimini atlar.

ignore-scheme-format

Şema biçimi kalite denetimini atlar.

ignore-vue-format

Vue I18n biçimlendirmesi kalite denetimini atlar.

ignore-translated

Çevrilmiş kalite denetimini atlar.

ignore-inconsistent

Tutarsız kalite denetimini atlar.

ignore-kashida

Kashida harfi kullanılmış kalite denetimini atlar.

ignore-md-link

Markdown bağlantıları kalite denetimini atlar.

ignore-md-reflink

Markdown başvuruları kalite denetimini atlar.

ignore-md-syntax

Markdown söz dizimi kalite denetimini atlar.

ignore-max-length

Çevirinin en fazla uzunluğu kalite denetimini atlar.

ignore-max-size

Çevirinin en fazla uzunluğu kalite denetimini atlar.

ignore-escaped-newline

\n aynı değil kalite denetimini atlar.

ignore-end-colon

İki nokta üst üste farklı kalite denetimini atlar.

ignore-end-ellipsis

Üç nokta karakteri farklı kalite denetimini atlar.

ignore-end-exclamation

Ünlem işareti farklı kalite denetimini atlar.

ignore-end-stop

Nokta işareti farklı kalite denetimini atlar.

ignore-end-question

Soru işareti farklı kalite denetimini atlar.

ignore-end-semicolon

Noktalı virgül farklı kalite denetimini atlar.

ignore-newline-count

Satır sonları farklı kalite denetimini atlar.

ignore-plurals

Çoğul biçim eksik kalite denetimini atlar.

ignore-placeholders

Yer belirticiler kalite denetimini atlar.

ignore-punctuation-spacing

Noktalama aralığı kalite denetimini atlar.

ignore-regex

Kurallı ifade kalite denetimini atlar.

ignore-same-plurals

Tekil ile çoğul biçim aynı kalite denetimini atlar.

ignore-begin-newline

Başta yeni satır var kalite denetimini atlar.

ignore-begin-space

Baştaki boşluk kalite denetimini atlar.

ignore-end-newline

Sondaki yeni satır kalite denetimini atlar.

ignore-end-space

Sondaki boşluk kalite denetimini atlar.

ignore-same

Çeviri değiştirilmemiş kalite denetimini atlar.

ignore-safe-html

Güvenli olmayan HTML kalite denetimini atlar.

ignore-url

Adres kalite denetimini atlar.

ignore-xml-tags

XML kodlaması kalite denetimini atlar.

ignore-xml-invalid

XML söz dizimi kalite denetimini atlar.

ignore-zero-width-space

Sıfır genişlikli boşluk kalite denetimini atlar.

ignore-ellipsis

Üç nokta kalite denetimini atlar.

ignore-icu-message-format-syntax

ICU MessageFormat söz dizimi kalite denetimini atlar.

ignore-long-untranslated

Uzun süredir çevrilmemiş kalite denetimini atlar.

ignore-multiple-failures

Birden çok denetim hatası kalite denetimini atlar.

ignore-unnamed-format

Birden çok değişken adlandırılmamış kalite denetimini atlar.

ignore-optional-plural

Çoğul değil kalite denetimini atlar.

Not: Kural genellikle, belirteci kullanılarak herhangi bir denetim için *ignore-** olarak adlandırılır. Böylece bunu kendi özel denetimleriniz için bile kullanabilirsiniz.

Bu imler hem *Bileşen yapılandırması* ayarlarında, her kaynak dizge ayarı için hem de çeviri dosyasının kendisinde (GNU gettext gibi) anlaşılır.

2.12.3 Denetimleri dayatmak

3.11 sürümünde geldi.

Bileşen yapılandırması içinde *Zorunlu denetimler* ayarını yaparak yok sayılmayacak denetimlerin listesini yapılandırabilirsiniz. Listelenen denetimler kullanıcı arayüzünden kapatılamaz ve bu denetimde hatalı bulunan dizgeler *Düzenlenmesi gerekli* olarak imlenir (bilgi almak için: :ref:`states`).

Not: Denetim dayatması kendiliğinden etkinleştirilmez. Denetim, dizge ya da bileşen imlerine karşılık gelen im eklenerek açılabilir.

Ayrıca bakınız:

Kaynak dizgeler ile ilgili ek bilgiler, Çeviri imleri

2.12.4 Yazı tiplerini yönetmek

3.7 sürümünde geldi.

İpucu: Weblate üzerine yüklenen yazı tipleri yalnızca *Çevirinin en fazla uzunluğu* denetimi için kullanılır. Weblate kullanıcı arayüzünde bir etkisi yoktur.

İşlenen metnin boyutlarını hesaplamak için kullanılan *Çevirinin en fazla uzunluğu* denetiminin yazı tipinin Weblate üzerine yüklenmesi ve bir çeviri imi kullanılarak seçilmesi gerekir (bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)).

Çeviri projenizin *Yönetim* menüsü altındaki *Yazı tipleri* içindeki Weblate yazı tipi yönetimi aracı, yazı tiplerinin yüklenip yönetilebileceği bir arayüz sağlar. TrueType veya OpenType yazı tipleri yüklenebilir, yazı tipi grupları ayarlanabilir ve bunlar denetimde kullanılabilir.

Yazı tipi grupları, genellikle Latince olmayan farklı diller için gerekli olan farklı yazı tiplerini tanımlanmanızı sağlar:

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

Yazı tipi grupları, boşluk veya özel karakterler içermeyen bir ad ile tanımlanır. Böylece denetim tanımında kolayca kullanılabilir:

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Fonts

Font groups
 Fonts

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 4.17
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

Yazı tipi ailesi ve biçemi, yüklendikten sonra kendiliğinden tanınır:

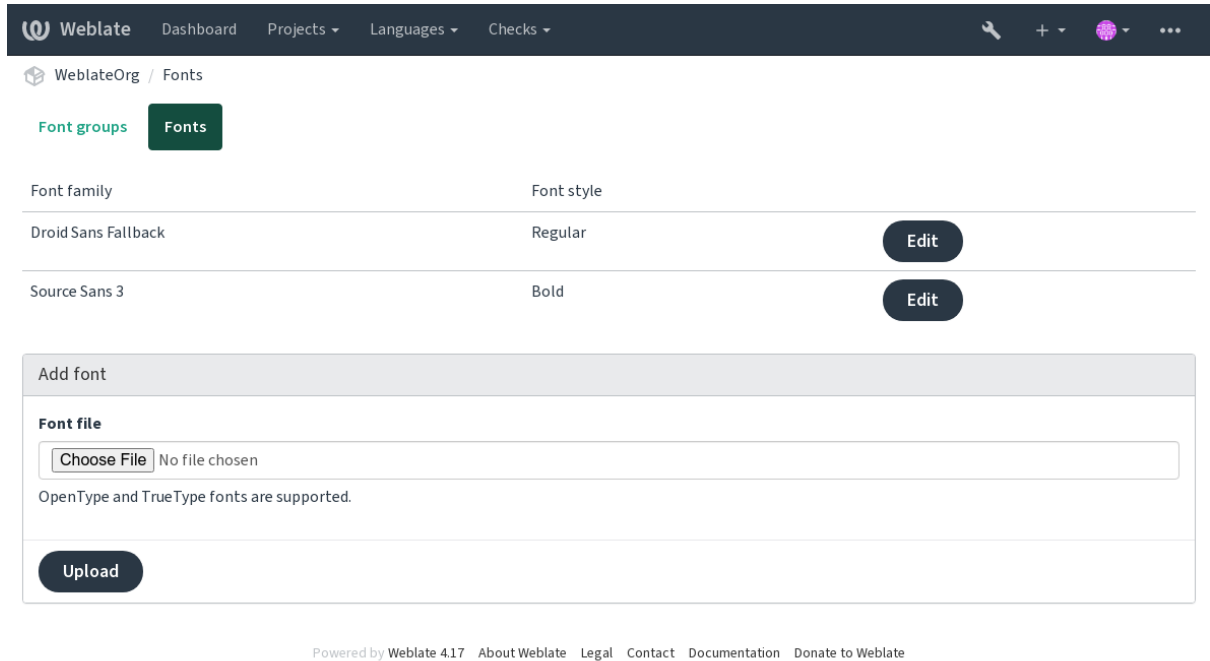
Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Fonts / Droid Sans Fallback Regular

Font	
Font family	Droid Sans Fallback
Font style	Regular
File size	3939852
Created	now
Uploaded by	testuser
Used in groups	
Delete	

Powered by Weblate 4.17
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

Weblate üzerine çeşitli yazı tipleri yükleyebilirsiniz:



Dizge uzunluğunu denetleme üzere yazı tiplerini kullanmak için, uygun imleri aktarın (bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)). Büyük olasılıkla şunlara gerek duyacaksınız:

max-size:500

En fazla genişliği piksel olarak belirler.

font-family:ubuntu

Belirteci ile kullanılacak yazı tipi grubunu tanımlar.

font-size:22

Yazı boyutunu piksel olarak tanımlar.

2.12.5 Kendi denetimlerinizi oluşturmak

Çok çeşitli kalite denetimleri hazır olarak bulunur (bilgi almak için: [Kalite denetimleri](#)). Ancak denetlemek istediğiniz her şey kapsanmamış olabilir. Yapılabilecek denetimlerin listesi [CHECK_LIST](#) kullanılarak ayarlanabilir. Ayrıca kendi özel denetimlerinizi de ekleyebilirsiniz.

1. `weblate.checks.Check` alt sınıfı
2. Birkaç öznelik ayarlayın.
3. `check` (kodunuzdaki çoğulları işlemek istiyorsanız) ya da `check_single` yöntemini uygulayın (sizin için hangisi işe yarıyorsa).

Bazı örnekler:

Özel denetimler kurmak için, [CHECK_LIST](#) içinde Python sınıfının tam yolunu belirtin. Bilgi almak için: `:ref:`custom-check-modules``.

“foo” içermeyen çeviri metinlerini denetlemek

Çeviride “foo” dizgesinin eksik olup olmadığına bakan basit bir denetim.

```
# 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
```

Çekçe çeviri metnlerinde çoğulların farklı olup olmadığı denetlemek

Çekçe iki çoğul biçimin aynı olmadığını doğrulamak için dil bilgilerini kullanan denetim.

```
# 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
```

(sonraki sayfaya devam)


(önceki sayfadan devam)





```
def check_single(self, source, target, unit):  
    """We don't check target strings here."""  
    return False
```


2.13 Kendiliğinden önerileri yapılandırmak

4.13 sürümünde değişti: Weblate 4.13 sürümünden önce, hizmetler *Yapılandırma* içinden yapılandırılıyordu.

Çeşitli makine çevirisi ve çeviri belleği hizmetlerinin desteği içeriden sağlanır. Her hizmet, yönetici tarafından tüm site için ya da projeye özel olarak ayarlanabilir:


[Weblate](#)
[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.

Not: Hizmetlerin kendi kullanım koşulları olduğundan, bunları istediğiniz gibi kullanmanıza izin verildiğinden emin olun.

Hizmetler, *Bileşen yapılandırması* konumunda yapılandırılmış kaynak dilden çeviri yapar. Bilgi almak için: `:ref:`component-source_language``.

Ayrıca bakınız:

Kendiliğinden öneriler

2.13.1 Amagama

Hizmet kimliği

amagama

Yapılandırma

Bu eklentinin yapılandırması yok.

Virtual geliştiricileri tarafından işletilen özel *tmserver* kurulumu.

Ayrıca bakınız:

Installing amaGama, Amagama, amaGama çeviri belleği

2.13.2 Apertium APy

Hizmet kimliği

apertium-apy

Yapılandırma

url	API adresi
-----	------------

Sınırlı bir dil kümesine çeviri sağlayan bir özgür yazılım makine çevirisi platformu.

Apertium kullanmak için kendi Apertium-APy sunucunuzu işletmeniz önerilir.

Ayrıca bakınız:

Apertium sitesi, Apertium APy belgeleri

2.13.3 AWS

3.1 sürümünde geldi.

Hizmet kimliği

aws

Yapılandırma

key	Erişim anahtarı kimliği
secret	API gizli anahtarı
region	Bölge adı

Amazon Translate, desteklenen çok sayıda dilde İngilizce'ye ve İngilizce'den metin çevirme için kullanılabilecek bir nöral makine çevirisi hizmetidir.

Ayrıca bakınız:[Amazon Translate belgeleri](#)**2.13.4 Baidu**

3.2 sürümünde geldi.

Hizmet kimliği

baidu

Yapılandırma

key	İstemci kimliği
secret	İstemci parolası

Baidu tarafından sunulan makine çevirisi hizmeti.

Bu hizmet bir API kullanır ve bunu kullanmak için Baidu üzerinden bir kimlik ve API anahtarı edinmeniz gerekir.

Ayrıca bakınız:[Baidu Translate API](#)**2.13.5 DeepL****Hizmet kimliği**

deepl

Yapılandırma

url	API adresi
key	API anahtarı

DeepL, birkaç dil için iyi makine çevirisi sağlayan ücretli bir hizmettir. *DeepL API* aboneliği satın almanız gerekir veya eski *DeepL Pro (klasik)* tarifesini kullanabilirsiniz.

DeepL hizmetiyle kullanılacak API adresi. Bu bilgi yazıldığında, API v1 yanında ücretsiz ve ücretli API v2 sürümleri bulunuyordu.

<https://api.deepl.com/v2/> (Weblate içinde varsayılan)

Ücretli olarak API kullanımı içindir ve abonelik yöntemi uygulanır.

<https://api-free.deepl.com/v2/>

Ücretsiz olarak API kullanımı içindir ve abonelik yöntemi uygulanır.

<https://api.deepl.com/v1/>

CAT araçları içindir ve kullanıcı başına abonelik yöntemi uygulanır.

Weblate daha önce, DeepL tarafından bir CAT aracı olarak sınıflandırılmıştı. Bu nedenle API v1 kullanması gerekiyordu. Ancak şimdi API v2 kullanması gerekiyor. Bu nedenle, v2 varsayılan değerdir ve var olan bir CAT aboneliğiniz varsa ve Weblate tarafından bunun kullanılmasını istiyorsanız bu değeri v1 olarak değiştirebilirsiniz.

Hangisini kullanacağınızı öğrenmenin en kolay yolu, tarayıcınızda aşağıdakine benzer bir adres açmaktır:

https://api.deepl.com/v2/translate?text=Hello&target_lang=TR&auth_key=XXX

XXX yerine kimlik doğrulama anahtarınızı (auth_key) yazın. “Bonjour” içeren bir JSON nesnesi alırsanız adresiniz doğrudur. Yoksa, diğer üçünü deneyin.

Weblate, DeepL resmîlik düzeyini destekler ve dile göre eşleşen birini seçer (örneğin, “**de@formal**” ve “**de@informal**” seçenekleri vardır).

Ayrıca bakınız:

[DeepL sitesi](#), [DeepL fiyatları](#), [DeepL API belgeleri](#)

2.13.6 Glosbe

Hizmet kimliği

glosbe

Yapılandırma

Bu eklentinin yapılandırması yok.

Hemen hemen yaşayan her dil için ücretsiz sözlük ve çeviri belleği.

API kullanımı ücretsizdir. Ancak çevirilerin kullanımı, kullanılan veri kaynağının lisansına bağlıdır. Kötüye kullanımı önlemek için bir IP adresinden belirli bir süre içinde yapılabilecek çağrılar sınırlıdır.

Ayrıca bakınız:

[Glosbe sitesi](#)

2.13.7 Google Translate

Hizmet kimliği

google-translate

Yapılandırma

key	API anahtarı
-----	--------------

Google tarafından sunulan makine çevirisi hizmeti.

Bu hizmet, Google Çeviri API uygulamasını kullanır. Bir API anahtarı edinmeli ve Google API konsolundan fatu-ralamayı etkinleştirmelisiniz.

Ayrıca bakınız:

[Google translate belgeleri](#)

2.13.8 Google Translate API v3

Hizmet kimliği

google-translate-api-v3

Yapılandırma

credentials	Google Çeviri hizmeti hesap bilgileri
project	Google Çeviri projesi
location	Google Çeviri konumu

Google Cloud hizmetleri tarafından sunulan makine çevirisi hizmeti.

Ayrıca bakınız:

[Google Çeviri belgeleri](#), İstemci kitaplıklarını kullanarak bulut hizmetlerinde kimlik doğrulaması, [Google Çeviri projesi oluşturmak](#), [Google Cloud App Engine konumları](#)

2.13.9 LibreTranslate

4.7.1 sürümünde geldi.

Hizmet kimliği

`libretranslate`

Yapılandırma

<code>url</code>	API adresi
<code>key</code>	API anahtarı

LibreTranslate, ücretsiz ve açık kaynaklı bir makine çevirisi hizmetidir. Herkese açık kopya için bir API anahtarı gerekir. Ancak LibreTranslate hizmetini kendiniz barındırabilirsiniz ve API uygulamasını ücretsiz olarak kullanabileceğini birkaç yansı bulunur.

`https://libretranslate.com/` (resmi herkese açık kopya)

Web sitesi dışında kullanmak için bir API anahtarı gerekir.

Ayrıca bakınız:

[LibreTranslate sitesi](#), [LibreTranslate deposu](#), [LibreTranslate yansıları](#)

2.13.10 Microsoft Terminology

Hizmet kimliği

`microsoft-terminology`

Yapılandırma

Bu eklentinin yapılandırması yok.

Microsoft Terminoloji hizmeti API uygulaması, Dil portalı üzerinde bulunan terminoloji, tanım ve kullanıcı arayüzü (UI) dizgelerine bir internet hizmeti üzerinden programlama ile erişmenizi sağlar.

Ayrıca bakınız:

[Microsoft Terminology hizmeti API uygulaması](#)

2.13.11 Microsoft Translator

Hizmet kimliği

`microsoft-translator`

Yapılandırma

key	API anah- tarı	
ba- se_1	Uy- gu- lama temel adre- si	Kullanılabilecek seçenekler: api.cognitive.microsofttranslator.com – Küresel (bölgesiz) api-apc.cognitive.microsofttranslator.com – Asya Pasifik api-eur.cognitive.microsofttranslator.com – Avrupa api-nam.cognitive.microsofttranslator.com – Kuzey Ame- rika api.translator.azure.cn – Çin api.cognitive.microsofttranslator.us – Azure ABD hükü- meti bulutu
end- po- int_	Kim- lik doğ- rula- ma hiz- meti adre- si	Bölgesel ya da çoklu hizmet, aşağıdaki bölge alanı kullanılarak belirtilebilir. Kullanılabilecek seçenekler: api.cognitive.microsoft.com – Genel api.cognitive.azure.cn – Çin api.cognitive.microsoft.us – Azure ABD hükümeti bulutu
re- gi- on	Kim- lik doğ- rula- ma hiz- meti böl- gesi	

Microsoft tarafından Azure portalında Cognitive Services altında sunulan makine çevirisi hizmeti.

Weblate, Translator API V3 kullanır.

Translator Text API V2

Translator API V2 ile kullandığınız anahtar, API 3 ile de kullanılabilir.

Translator Text API V3

Azure portalında bir hesap açmanız ve oradan alacağınız anahtar kullanmanız gerekir. Yeni Azure anahtarlarıyla, `region` seçeneği için hizmetinizin yerel ayarını belirtmeniz gerekir.

İpucu: Azure Çin için lütfen Azure Portal üzerindeki uç noktanızı kullanın.

Ayrıca bakınız:

Cognitive Services - Text Translation API, Microsoft Azure Portal, Temel adresler, "Çok hizmetli bir kaynak ile kimlik doğrulamak" "Erişim kodu ile kimlik doğrulamak" bölümü

2.13.12 ModernMT

4.2 sürümünde geldi.

Hizmet kimliği

modernmt

Yapılandırma

url	API adresi
key	API anahtarı

Ayrıca bakınız:

[ModernMT API](#),

2.13.13 MyMemory

Hizmet kimliği

mymemory

Yapılandırma

email	İletişim e-posta adresi
username	Kullanıcı adı
key	API anahtarı

Makine çevirisi ile çok büyük bir çeviri belleği.

Ücretsiz, anonim kullanım şu anda günde 100 istek ya da `email` içine bir iletişim e-posta adresi yazdığınızda günde 1.000 istek ile sınırlıdır. Ayrıca onlara yazarak daha fazlasını da isteyebilirsiniz.

Ayrıca bakınız:

[MyMemory sitesi](#)

2.13.14 Netease Sight

3.3 sürümünde geldi.

Hizmet kimliği

netease-sight

Yapılandırma

key	İstemci kimliği
secret	İstemci parolası

NetEase tarafından sunulan makine çevirisi hizmeti.

Bu hizmet bir API kullanır ve bir NetEase anahtarı ile parolası edinmeniz gerekir.

Ayrıca bakınız:

[NetEase Sight çeviri platformu](#)

2.13.15 SAP Translation Hub

Hizmet kimliği

sap-translation-hub

Yapılandırma

url	API adresi
key	API anahtarı
user-name	SAP kullanıcı adı
password	SAP parolası
enable_mt	Makine çevirisi kullanılıp-sın
do-main	Çeviri etki alanı
	Bir çeviri etki alanının kimliği. Örnek: BC. Bir etki alanı belirtmezseniz, yöntem kullanılabilir tüm etki alanlarındaki çevirileri arar.

SAP tarafından sunulan makine çevirisi hizmeti.

Bu hizmeti kullanmak için bir SAP hesabınızın (ve SAP Cloud Platform üzerinde etkinleştirilmiş SAP Translation Hub) olması gerekir.

Veri tabanı terimine ek olarak, makine çevirisi hizmetlerinin de kullanılıp kullanılmayacağını yapılandırabilirsiniz.

Not: Sandbox API erişimi için url ve key değerlerini ayarlamalısınız.

Üretken API erişimi için url, username ve password seçeneklerini ayarlamalısınız.

Ayrıca bakınız:

SAP Translation Hub API, [`SAP Translation Hub temel adresini oluşturmak<https://help.sap.com/docs/SAP_TRANSLATION_HUB/ed6ce7a29bdd42169f5f0d7868bce6eb/3a011fba82644259a2cc3`](https://help.sap.com/docs/SAP_TRANSLATION_HUB/ed6ce7a29bdd42169f5f0d7868bce6eb/3a011fba82644259a2cc3)

2.13.16 tmserver

Hizmet kimliği

tmserver

Yapılandırma

url	API adresi
-----	------------

Translate-toolkit ile birlikte geleni kullanarak kendi çeviri belleği sunucunuzu işletebilir ve Weblate ile konuşmasına izin verebilirsiniz. Ayrıca, geliştirilmiş bir tmserver sürümü olan bir amaGama sunucusu ile de kullanabilirsiniz.

- Önce çeviri belleği içine bazı verileri aktarmak isteyeceksiniz:

```
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
```

- İsteklerinizi dinlemek üzere tmserver uygulamasını başlatın:

```
tmserver -d /var/lib/tm/db
```

3. İletişim kurmak için gereken Weblate yapılandırmasını ayarlayın. Varsayılan adres `http://localhost:8888/tmserver/`.

Ayrıca bakınız:

[tmserver](#) [Installing amaGama](#), [Amagama](#), [Amagama çeviri belleği](#)

2.13.17 IBM Watson Language Translator

Hizmet kimliği

`ibm`

Yapılandırma

<code>url</code>	API adresi
<code>key</code>	API anahtarı

IBM Watson Language Translator, metni bir dilden diğerine çevirir. Hizmet, etki alanına özgü birden çok model sunar.

Ayrıca bakınız:

[Watson Language Translator](#), [IBM Cloud API belgeleri](#)

2.13.18 Weblate

Hizmet kimliği

`weblate`

Yapılandırma

Bu eklentinin yapılandırması yok.

Weblate makine çevirisi hizmeti, Weblate içinde zaten çevrilmiş olan dizgeler ile ilgili çevirileri sunabilir. Var olan dizgelerdeki tam eşleşmeleri arar.

2.13.19 Weblate Çeviri Belleği

Hizmet kimliği

`weblate-translation-memory`

Yapılandırma

Bu eklentinin yapılandırması yok.

Makine çevirisi hizmeti olarak [Çeviri belleği](#) kullanır. Geçmişte çevrilmiş (ya da çeviri belleğine yüklenmiş) herhangi bir dizge bu şekilde çevrilebilir.

2.13.20 Yandex

Hizmet kimliği

`yandex`

Yapılandırma

<code>key</code>	API anahtarı
------------------	--------------

Yandex tarafından sunulan makine çevirisi hizmeti.

Bu hizmet bir Translation API uygulaması kullanır ve Yandex üzerinden bir API anahtarı edinmeniz gerekir.

Ayrıca bakınız:

[Yandex Translate API](#), [Translate Yandex](#) tarafından sunulur

2.13.21 Youdao Zhiyun

3.2 sürümünde geldi.

Hizmet kimliği

youdao-zhiyun

Yapılandırma

key	İstemci kimliği
secret	İstemci parolası

Youdao tarafından sunulan makine çevirisi hizmeti.

Bu hizmet bir API uygulaması kullanır ve Youdao üzerinden bir kimlik ve API anahtarı edinmeniz gerekir.

Ayrıca bakınız:

[Youdao Zhiyun doğal dil çeviri hizmeti](#)

2.13.22 Özel makine çevirisi

Birkaç satır Python kodu kullanarak kendi makine çevirisi hizmetlerinizi de kullanabilirsiniz. Bu örnek, dictionary Python modülünü kullanarak sabit bir dil listesinde makine çevirisi yapar:

```
# 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"

    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,
```

(sonraki sayfaya devam)

(önceki sayfadan devam)


```
threshold: int = 75,
):
    """Return tuple with translations."""
    for t in dictionary.translate(text):
        yield {"text": t, "quality": 100, "service": self.name, "source": text}
```

Weblate `WEBLATE_MACHINERY` seçeneğinde belirteceğiniz kendi sınıfınızı kullanır.

2.14 Eklentiler

Eklentiler, çeviri iş akışını özelleştirmesi ve kendiliğinden yapılması için yöntemler sağlar. Yöneticiler, her bir çeviri bileşenine *Yönetim* ↓ *Eklentiler* menüsünden eklentiler ekleyip yönetebilir.

İpucu: Eklentileri `API`, `DEFAULT_ADDONS` veya `:d jadmin:install_addon` kullanarak da yapılandırabilirsiniz.



[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 Hazır eklentiler

Kendiliğinden çeviri

3.9 sürümünde geldi.

Eklenti kimliği

`weblate.autotranslate.autotranslate`

Yapılandırma

<code>mo-</code>	Kendi-	Kullanılabilecek seçenekler:
<code>de</code>	liğinden	<code>suggest</code> – Öneri olarak ekler
	çeviri	<code>translate</code> – Çeviri olarak ekler
	kipi	<code>fuzzy</code> – “Düzenlenmesi gerekli” olarak ekler
<code>fil-</code>	Arama	Lütfen tüm dizgeleri çevirdiğinizde var olan tüm çevirileri yok sayılacağını
<code>ter_</code>	süzgeci	unutmayın.
		Kullanılabilecek seçenekler:
		<code>all</code> – Tüm dizgeler
		<code>nottranslated</code> – Çevrilmemiş dizgeler
		<code>todo</code> – Tamamlanmamış dizgeler
		<code>fuzzy</code> – Düzenlenecek olarak imlenmiş dizgeler
		<code>check:inconsistent</code> – Denetim hatası: Tutarsız
<code>autoc</code>	Kendi-	Kullanılabilecek seçenekler:
	liğinden	<code>others</code> – Diğer çeviri bileşenleri
	yapılmış	<code>mt</code> – Makine çevirisi
	çevi-	
	rilerin	
	kaynağı	
<code>com-</code>	Bileşen-	Kaynak olarak kullanılacak bir bileşenin kısaltmasını yazın. Geçerli proje-
<code>po-</code>	ler	deki tüm bileşenlerin kullanılması için boş bırakın.
<code>nent</code>		
<code>en-</code>	Makine	
<code>gi-</code>	çevirisi	
<code>nes</code>	hizmetle-	
	ri	
<code>th-</code>	Skor eşiği	
<code>res-</code>		
<code>holc</code>		

Tetikleyiciler

bileşen güncellemesi, günlük

Makine çevirisini ya da diğer bileşenleri kullanarak dizgeleri kendiliğinden çevirir.

Ne zaman tetiklenir:

- Bileşende yeni dizgeler görüldüğünde.
- Her bileşen için ayda bir kez. `:setting:BACKGROUND_TASKS` kullanılarak yapılandırılabilir.

Ayrıca bakınız:

Kendiliğinden çeviri, Bileşenler arasında çevirilerin tutarlılığını sağlamak

JavaScript yerelleştirme CDN

4.2 sürümünde geldi.

Eklenti kimliği

`weblate.cdn.cdnjs`

Yapılandırma

thres- hold	Çeviri eşiği	Çevirilerin katılacağı eşik değeri.
css_sele	CSS seçici	Yerelleştirilebilir öğeleri algılamak için CSS seçici.
co- okie_nam	Dil tanımlama bilgisi- nin adı	Dil seçiminin kaydedildiği tanımlama bilgisinin adı.
files	Dizgeleri HTML dos- yalarından ayıklama	Geçerli depoda ya da uzak adreslerde çevrilebilir diz- geler için işlenecek dosya adlarının listesi.

Tetikleyiciler

günlük, depoya işlendikten sonra, depo güncellendikten sonra

JavaScript ya da HTML yerelleştirmesinde kullanılmak üzere içerik dağıtım ağına çevirileri yayınlar.

Statik HTML sayfalarını yerelleştirmek veya JavaScript kodunda yerelleştirmeyi yüklemek için kullanılabilir.

Bileşeniniz için, bunları yerelleştirmek amacıyla HTML sayfalarına ekleyebileceğiniz benzersiz bir adres oluşturur. Bilgi almak için `weblate-cdn`.

Ayrıca bakınız:

`cdn-addon-config`, `weblate-cdn`, `cdn-addon-extract`, `cdn-addon-html`

Boş dizgeleri silme

4.4 sürümünde geldi.

Eklenti kimliği

`weblate.cleanup.blank`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya işlendikten sonra, depo güncellendikten sonra

Çeviri dosyalarından çevirisi olmayan dizgeleri kaldırır.

Çeviri dosyalarında boş dizgelerin görünmemesi için bunu kullanın (yerelleştirme kitaplığınızın kaynak dizgeyi kullanmak yerine boş dizgeler görüntülenmesi gibi durumlarda).

Ayrıca bakınız:

Weblate, çevirilerin yanında çeviri dosyalarını da günceller mi?

Çeviri dosyalarını temizleme

Eklenti kimliği

`weblate.cleanup.generic`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya işlenmeden önce, depo güncellendikten sonra

Tüm çeviri dosyalarını tek dilli temel dosyayla eşleştirecek şekilde günceller. Bu işlem çoğu dosya biçiminde temel dosyada artık bulunmayan eski çeviri anahtarlarını kaldırır.

Ayrıca bakınız:

Weblate, çevirilerin yanında çeviri dosyalarını da günceller mi?

Eksik dilleri ekleme

Eklenti kimliği

`weblate.consistency.languages`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

günlük, depoya eklenmesinden sonra

Proje içindeki tüm bileşenler için tutarlı bir dil kümesinin kullanılmasını sağlar.

Eksik diller her 24 saatte bir ve Weblate üzerine yeni diller eklendiğinde denetlenir.

Diğerlerinden farklı olarak, bu eklenti tüm projeyi etkiler.

İpucu: Yeni eklenen dizgeler *Kendiliğinden çeviri* ile kendiliğinden çevrilir.

Bileşen keşfi

Eklenti kimliği

weblate.discovery.discovery

Yapılandırma

match	Çeviri dosyalarıyla eşleştirilecek kurallı ifade	
file_format	Dosya biçimi	
name_template	Bileşen adını özelleştirme	
base_file_type	Tek dil temel dosya adını tanımlama	İki dilli çeviri dosyaları için boş bırakın.
new_base_name	Yeni çeviriler için temel dosyayı tanımlama	Yeni çevirileri oluşturmak için kullanılan dosyanın adı. Gettext için .pot dosyası olarak seçin.
intermediate_template	Ara dil dosyası	Ara çeviri dosyasının adı. Çoğu durumda bu dosya geliştiriciler tarafından sunulan bir çeviri dosyasıdır ve gerçek kaynak dizgeler oluşturulurken kullanılır.
language_regex	Dil süzgeci	Dosya maskesi taranırken çeviri dosyalarını süzmek için kullanılan kurallı ifade.
copy_addons	Ana bileşendeki eklentileri yeni oluşturulanlara kopyalama	
remove	Var olmayan dosyalar için bileşenleri kaldırma	
confirm	Yukarıdaki eşleşmelerin doğru görüldüğünü onaylıyorum	

Tetikleyiciler

depo güncellenmesinden sonra

Sürüm denetim sistemindeki dosya değişikliklerine göre proje bileşenlerini kendiliğinden ekler ya da siler.

Sürüm denetimi sistemi her güncellendiğinde tetiklenir ve `:d jadmin:import_project` yönetim komutuna benzer. Bu şekilde, bir sürüm denetimi sistemi içinde birden çok çeviri bileşenini izleyebilirsiniz.

Eşleştirme, karmaşık yapılandırmayı etkinleştiren kurallı ifadeler kullanılarak yapılır. Ancak bunu yapmak için bazı bilgiler gerekir. Yaygın kullanım örnekleri eklenti yardımı bölümünde bulunabilir.

Kaydet üzerine tıkladığınızda, yapılandırmanın gereksinimlerinize uyup uymadığını denetleyebileceğiniz biçimde eşleşen bileşenlerin ön izlemesi sunulur:

Weblate
Dashboard
Projects
Languages
Checks

WeblateOrg / Language names / Add-ons / Component discovery

Configure add-on

Please review and confirm the matched components.

Component	Matched files
The following components would be created	
Djangojs (djangojs)	File mask: <code>weblate/locale/*/LC_MESSAGES/djangojs.po</code> <code>weblate/locale/cs/LC_MESSAGES/djangojs.po</code> (cs) <code>weblate/locale/hu/LC_MESSAGES/djangojs.po</code> (hu) <code>weblate/locale/he/LC_MESSAGES/djangojs.po</code> (he)
Django (django)	File mask: <code>weblate/locale/*/LC_MESSAGES/django.po</code> <code>weblate/locale/cs/LC_MESSAGES/django.po</code> (cs) <code>weblate/locale/hu/LC_MESSAGES/django.po</code> (hu) <code>weblate/locale/he/LC_MESSAGES/django.po</code> (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
<code>(?P<language>["/"].)*/(?P<component>["/"]*)\.po</code>	<code>cs/application.po</code> <code>cs/website.po</code> <code>de/application.po</code> <code>de/website.po</code>	One folder per language containing translation files for components.
<code>locale/(?P<language>["/"].)*LC_MESSAGES/(?P<component>["/"]*)\.po</code>	<code>locale/cs/LC_MESSAGES/application.po</code> <code>locale/cs/LC_MESSAGES/website.po</code> <code>locale/de/LC_MESSAGES/application.po</code> <code>locale/de/LC_MESSAGES/website.po</code>	Usual structure for storing gettext PO files.
<code>src/locale/(?P<component>["/"]*)\.(?P<language>["/"].)*\.po</code>	<code>src/locale/application.cs.po</code> <code>src/locale/website.cs.po</code> <code>src/locale/application.de.po</code> <code>src/locale/website.de.po</code>	Using both component and language name within filename.
<code>locale/(?P<language>["/"].)*/(?P<component>["/"]*)/(?P<language>["/"].)*\.po</code>	<code>locale/cs/application/cs.po</code> <code>locale/cs/website/cs.po</code> <code>locale/de/application/de.po</code> <code>locale/de/website/de.po</code>	Using language in both path and filename.
<code>res/values-(?P<language>["/"].)*strings-(?P<component>["/"]*)\.xml</code>	<code>res/values-cs/strings-about.xml</code> <code>res/values-cs/strings-help.xml</code> <code>res/values-de/strings-about.xml</code> <code>res/values-de/strings-help.xml</code>	Android resource strings, split into several files.
<code>(?P<originalHierarchy>.*)(?P<component>["/"]*)/src/main/resources/ApplicationResources_(?P<language>["/"].)*.properties</code>	<code>parent/module1/submodule/src/main/resources/ApplicationResources_fr.properties</code> <code>parent/module1/submodule/src/main/resources/ApplicationResource_es.properties</code> <code>parent/module2/src/main/resources/ApplicationResource_de.properties</code> <code>parent/module2/src/main/resources/ApplicationResource_ro.properties</code>	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

Powered by Weblate 4.17
About Weblate
Legal
Contact
Documentation
Donate to Weblate

İpucu: Bileşen bulma eklentisi *İç Weblate adresleri* kullanır. Sürüm denetimi sistemi kurulumunu birden çok bileşen arasında paylaşmanın kullanışlı bir yoludur. Bağlı bileşenler, ilgili her bir bileşenin *Kaynak kod deposu* alanına (Yönetim ↓ Ayarlar ↓ Sürüm denetimi sistemi) `weblate://project/main-component` doldurularak kurulan ana bileşenin yerel deposunu kullanır. Böylece yapılandırma ve sistem kaynakları açısından da zaman kazanılır.

Ayrıca bakınız:

Kalıp kodlaması

Toplu düzenleme

3.11 sürümünde geldi.

Ekenti kimliği

`weblate.flags.bulk`

Yapılandırma

q	Sorgu	
state	Ayarlanacak durum	Kullanılabilecek seçenekler: -1 – Değiştirilmesin 10 – Düzenlenmesi gerekli 20 – Çevrilmiş 30 – Onaylanmış
add_flag:	Eklenenecek çeviri imleri	
remove_flags:	Kaldırılacak çeviri imleri	
add_label:	Eklenenecek etiketler	
remove_label:	Kaldırılacak etiketler	

Tetikleyiciler

bileşen güncellemesi

Dizgelerin imlerini, etiketlerini veya durumlarını toplu düzenleyin.

NOT `has:label` arama sorgusuyla başlayarak kendiliğinden etiketleme yapın ve tüm dizgelere gerekli tüm etiketler eklenene kadar sürdürün. Weblate üst verileri için başka kendiliğinden işlemler de yapılabilir.

Örnekler:

Tablo 5: Yeni dizgeler kendiliğinden etiketlensin

Arama sorgusu	<code>NOT has:label</code>
Eklenenecek etiketler	<code>son</code>

Tablo 6: Tüm Uygulama mağazası üst veri dosyaları değişiklik günlüğü dizgelerini salt okunur olarak imleme

Arama sorgusu	<code>language:en AND key:changelogs/</code>
Eklenenecek çeviri imleri	<code>read-only</code>

Ayrıca bakınız:

Toplu düzenleme, Parametreleri kullanarak davranışı özelleştirmek, labels

Değiştirilmemiş çevirileri “Düzenlenmesi gerekli” olarak imleme

3.1 sürümünde geldi.

Eklenti kimliği

`weblate.flags.same_edit`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

birim oluşturulduktan sonra

Sürüm denetimi sisteminden yeni bir çevrilebilir dizge içe aktarıldığında ve bir kaynak dizgeyle eşleştiğinde, Weblate üzerinde düzenlenmesi gerekli olarak imlenir. Özellikle çevrilmemiş dizgeler için kaynak dizgeleri içeren dosya biçimleri için kullanışlıdır.

İpucu: Ayrıca, *Çeviri imleri* için `strict-same` imini ekleyerek *Çeviri değiştirilmemiş* denetimini sıkılaştırmak isteyebilirsiniz.

Ayrıca bakınız:

Çeviri durumları

Yeni kaynak dizgeleri “Düzenlenmesi gerekli” olarak imleme

Eklenti kimliği

`weblate.flags.source_edit`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

birim oluşturulduktan sonra

Sürüm denetimi sisteminden yeni bir kaynak dizge içe her aktarıldığında, Weblate üzerinde düzenlenmesi gerekli olarak imlenir. Böylece geliştiriciler tarafından yazılan kaynak dizgeleri kolayca süzebilir ve düzenleyebilirsiniz.

Ayrıca bakınız:

Çeviri durumları

Yeni çevirileri “Düzenlenmesi gerekli” olarak imleme

Eklenti kimliği

`weblate.flags.target_edit`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

birim oluşturulduktan sonra

Sürüm denetimi sisteminden yeni bir çevrilebilir dizge içeri her aktarıldığında, Weblate üzerinde düzenlenmesi gerekli olarak imlenir. Böylece geliştiricilerin oluşturduğu çevirileri kolayca süzebilir ve düzenleyebilirsiniz.

Ayrıca bakınız:

Çeviri durumları

İstatistik oluşturma

Eklenti kimliği

`weblate.generate.generate`

Yapılandırma

<code>filename</code>	Oluşturulacak dosyanın adı
<code>template</code>	Oluşturulacak dosyanın içeriği

Tetikleyiciler

depoya işlenmeden önce

Çeviri durumu ile ilgili ayrıntılı bilgiler içeren bir dosya oluşturur.

Hem dosya adında hem de içerikte bir Django kalıbı kullanabilirsiniz. Markup ile ilgili bilgi almak için [Kalıp kodlaması](#) bölümüne bakabilirsiniz.

Örneğin, her çeviri için bir özet dosyası oluşturma:

Oluşturulacak dosyanın adı

`locale/{{ language_code }}.json`

İçerik

```
{
  "language": "{{ language_code }}",
  "strings": "{{ stats.all }}",
  "translated": "{{ stats.translated }}",
  "last_changed": "{{ stats.last_changed }}",
  "last_author": "{{ stats.last_author }}"
}
```

Ayrıca bakınız:

[Kalıp kodlaması](#)

Çeviriyi kaynak dizgeyle önceden doldurma

4.11 sürümünde geldi.

Eklenti kimliği

`weblate.generate.prefill`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

bileşen güncellemesi, günlük

Çeviri dizgelerini kaynak dizgeler ile doldurur.

Bileşendeki tüm çevrilmemiş dizgeler kaynak dizgeyle doldurulur ve düzenlenmesi gerekli olarak imlenir. Çeviri dosyalarında boş dizge kabul edilmediğinde bunu kullanın.

Sözde yerel oluşturma

4.5 sürümünde geldi.

Ekenti kimliği

`weblate.generate.pseudolocale`

Yapılandırma

<code>source</code>	Kaynak dizgeler	
<code>target</code>	Hedef çeviri	Bu çevirideki tüm dizgelerin üzerine yazılacak
<code>prefix</code>	Sabit dizge ön eki	
<code>var_prefix</code>	Değişken dizge ön eki	
<code>suffix</code>	Sabit dizge son eki	
<code>var_suffix</code>	Değişken dizge son eki	
<code>var_multipli</code>	Değişken bölüm çarpanı	Kaynak dizgenin uzunluğuna bağlı olarak değişken bölümün kaç kez yinleneceği.
<code>include_readonly</code>	Salt okunur dizgeleri katma	

Tetikleyiciler

bileşen güncellemesi, günlük

Kaynak dizgelere kendiliğinden ön ek ve son ek ekleyerek çeviri oluşturur.

Sözde yereller, yerelleştirme için hazırlanmamış dizgeleri bulmaya yarar. Böylece, uygulamayı sözde yerel dilde çalıştırırken değiştirilmemiş dizgelerin bulunmasını kolaylaştırmak için tüm çevrilebilir kaynak dizgeleri değiştirilir.

Yerelleştirilmiş karşılıkları görünümüne uymayabilecek dizgeler de bulunabilir.

Değişken bölümlerin kullanılması, yerelleştirmeden sonra kullanıcı arabirimine sığmayabilecek dizgelerin bulunmasını sağlar. Metin kaynak dizge uzunluğuna göre genişletilir. Değişken bölümler, metnin uzunluğunun çarpanla çarpılmasıyla yinelenir. Örneğin, değişken son eki `_` olan `Merhaba dünya` ve değişken çarpanı `1` olduğunda `Merhaba dünya_____` olur. Son ek kaynak dizgedeki her karakter için bir kez yinelenir.

Dizgeler şu model kullanılarak oluşturulacak:

Sabit dizge ön eki Değişken dizge ön eki Kaynak dizge Değişken dizge son eki Sabit dizge son eki

İpucu: Deneme için gerçek dilleri kullanabilirsiniz. Bunun yanında Weblate üzerinde *en_XA* ve *ar_XB* gibi bu iş için ayrılmış sahte ayarlar vardır.

İpucu: Bu eklentiye, var olan bir dilin ya da benzer bir dilin yeni bir yerel ayarına çeviri başlatmak için kullanabilirsiniz. Çeviriyi bileşene ekledikten sonra, eklentiye izleyin. *Örnek:* *fr* diliniz varsa ve *fr_CA* çevirisini başlatmak istiyorsanız, kaynak olarak *fr*, hedef olarak *fr_CA* ayarlayın ve ön ek ile son eki boş bırakın.

Kopyalamadan sonra yapılan çevirilerin Weblate tarafından değiştirilmesini önlemek için yeni çeviriyi doldurduktan sonra eklentiye kaldırın.

Katılımcıları yorum bölümünde görüntüleme

Eklenti kimliği

weblate.gettext.authors

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya işlenmeden önce

PO dosyası üst bilgilerinin yorum bölümü, katılımcı adlarını ve katkı yıllarını içerecek biçimde güncellenir.

PO dosya üst bilgileri şöyle görünür:

```
# 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.
```

“configure” dosyasındaki ALL_LINGUAS değişkeni güncelleme

Eklenti kimliği

weblate.gettext.configure

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya eklendikten sonra, günlük

Yeni bir çeviri eklendiğinde configure, configure.in ya da herhangi bir configure.ac dosyasında ALL_LINGUAS değişkeni güncellenir.

Gettext çıktısını özelleştirme

Eklenti kimliği

weblate.gettext.customize

Yapılandırma

widt	Uzun	Varsayılan olarak gettext satırları 77 karakterden sonra ve yeni satır karakterlerinde kaydırır. –no-wrap parametresi kullanıldığında kaydırma yalnızca yeni satır karakterlerinde yapılır.
kay-	lar	Kullanılacak seçenekler:
dır-	ma	77 – Satırlar, 77 karakterden sonra ve yeni satır karakterlerinde kaydırılır (xgettext varsayılanı)
		65535 – Satırlar yalnızca yeni satırlar karakterlerinde kaydırılır ('xgettext –no-wrap' gibi)
		–1 – Satırlar kaydırılmaz

Tetikleyiciler

depolamaya yüklendikten sonra

Gettext çıktısında satır kaydırma gibi davranışların özelleştirilmesini sağlar.

Şu seçenekleri sunar:

- Satırlar 77 karakter ve yeni satır karakterlerinde kaydırılır
- Satırları yalnızca yeni satır karakterlerinde kaydırılır

- Satırlar kaydırılmaz

Not: Varsayılan olarak gettext satırları 77 karakterde ve yeni satır karakterlerinde kaydırır. `--no-wrap` parametresi kullanıldığında kaydırma yalnızca yeni satır karakterlerinde yapılır.

LINGUAS dosyasını güncelleme

Eklenti kimliği

`weblate.gettext.linguas`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya eklendikten sonra, günlük

Yeni bir çeviri eklendiğinde LINGUAS dosyası güncellenir.

MO dosyaları üretme

Eklenti kimliği

`weblate.gettext.mo`

Yapılandırma

<code>path</code>	Oluşturulan MO dosyasının yolu	Belirtilmemişse, PO dosyasının konumu kullanılır.
-------------------	--------------------------------	---

Tetikleyiciler

depoya işlenmeden önce

PO dosyası her değiştirildiğinde MO dosyası kendiliğinden oluşturulur.

Oluşturulan MO dosyasının konumu özelleştirilebilir ve bunun için *Kalıp kodlaması* alanı kullanılır.

Not: Bir çeviri silinirse, PO dosyası depodan silinir, ancak bu eklenti tarafından oluşturulmuş MO dosyası silinmez. MO dosyası yukarı akıştan el ile kaldırılmalıdır.

PO dosyalarını POT ile eşleştirerek güncelleme (msgmerge)

Eklenti kimliği

`weblate.gettext.msgmerge`

Yapılandırma

<code>previous</code>	Çevrilmiş dizgelerin önceki msgid değerleri saklansın
<code>no_location</code>	Çevrilmiş dizgelerin konumları kaldırılсын
<code>fuzzy</code>	Belirsiz eşleşme kullanılsın

Tetikleyiciler

depo güncellenmesinden sonra

Tüm PO dosyalarını (*Dosya maskesi* ile yapılandırıldığı gibi) **msgmerge** komutunu kullanarak POT dosyasıyla eşleşecek şekilde (*Yeni çeviriler kalıbı* ile yapılandırıldığı gibi) günceller.

Yukarı akış deposundan yeni değişiklikler çekildiğinde tetiklenir. Eklenti yapılandırması aracılığıyla msgmerge komut satırı seçeneklerinin çoğu yapılandırılabilir.

Ayrıca bakınız:

Weblate, çevirilerin yanında çeviri dosyalarını da günceller mi?

Git işlemlerini bir araya toplar

Eklenti kimliği

weblate.git.squash

Yapılandırma

squash	İşlemleri bir araya toplama	Kullanılabilecek seçenekler: all – Tüm işlemler bir arada yapılır language – Her dilin işlemleri ayrı yapılır file – Her dosyanın işlemleri ayrı yapılır author – Her katkıda bulunanın işlemleri ayrı yapılır
append	Bir araya toplanmış işlemlerin iletisine art bilgileri eklensin	'Co-authored-by: ...' gibi bir işleme iletisinin başka bir serbest biçim bölümünün sonunda, art bilgi satırları RFC 822 e-posta üst bilgilerine benzeyen satırlardır.
commit	İşleme iletisi	Bu işleme iletisi, bir araya toplanmış işleme iletilerinden gelen birleşik işleme iletileri yerine kullanılır.

Tetikleyiciler

depoya işlendikten sonra

Değişiklikleri yollamadan önce Git işlemlerini bir araya toplar.

Git işlemleri, değişiklikler yollanmadan önce aşağıdaki kiplerden biri kullanılarak bir araya toplanabilir:

- Tüm işlemler bir arada
- Dile göre
- Dosyaya göre
- Katkıda bulunana göre

Özgün işleme iletileri korunur. Ancak *Katkıda bulunana göre* olarak seçilmediği ya da işleme iletisi eklenecek şekilde özelleştirilmediği sürece katkıda bulunan bilgisi eksik kalır.

Özgün işleme iletileri, isteğe bağlı olarak özel bir işleme iletisiyle değiştirilebilir.

Art bilgiler (Co-authored-by: ... gibi işleme satırları) isteğe bağlı olarak özgün işleme iletilerinden kaldırılabilir ve bir araya toplanmış işleme iletisinin sonuna eklenebilir. Bu aynı zamanda her çevirmen için özel bir "Co-authored-by:" bilgisi oluşturur.

JSON çıktısını özelleştirme

Eklenti kimliği

weblate.json.customize

Yapılandırma

sort_key	JSON anahtarlarını sıralama	
indent	JSON girintisi	
style	JSON girintisinin biçimi	Kullanılabilecek seçenekler: spaces – Boşluklar tabs – Sekmeler

Tetikleyiciler

depolamaya yüklendikten sonra

JSON çıktısındaki girinti ya da sıralama gibi davranışların ayarlanmasını sağlar.

Java özellikleri dosyasını biçimlendirme**Eklenti kimliği**

weblate.properties.sort

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya işlenmeden önce

Java özellikleri dosyasını biçimlendirir ve sıralar.

- Yeni satır karakterlerini Unix karakterleri olarak değiştirir.
- Unicode kaçış dizilerini büyük harfle biçimlendirir (varsa).
- Boş satırları ve yorumları siler.
- Dizgeleri anahtarlara göre sıralar.
- Yinelenen dizgeleri kaldırır.

Eski yorumları kaldırma

3.7 sürümünde geldi.

Eklenti kimliği

weblate.removal.comments

Yapılandırma

age	Tutulacak gün sayısı
-----	----------------------

Tetikleyiciler

günlük

Yorumların kaldırılması için bir zaman dilimi ayarlayabilirsiniz.

Bu özellik, güncelliğini yitirmiş olabilecek eski yorumları kaldırmak için yararlı olabilir. Bir yorumun eski olması önemini yitirdiği anlamına gelmediğinden dikkatli kullanın.

Eski önerileri kaldırma

3.7 sürümünde geldi.

Eklenti kimliği

`weblate.removal.suggestions`

Yapılandırma

age	Tutulacak gün sayısı	
votes	Oy verme eşiği	Kaldırma eşiği. Oylama kapalıyken bu alanın herhangi bir etkisi yoktur.

Tetikleyiciler

günlük

Önerilerin kaldırılması için bir zaman dilimi ayarlayabilirsiniz.

Belirli bir zaman diliminde yeterince olumlu oy almayan önerilerin kaldırılması için öneri oylamayla (bilgi almak için [Akran değerlendirmesi](#)) bağlantılı olarak çok yararlı olabilir.

RESX dosyalarını güncelleme

3.9 sürümünde geldi.

Eklenti kimliği

`weblate.resx.update`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depo güncellenmesinden sonra

Tüm çeviri dosyalarını tek dilli temel dosya ile eşleştirecek biçimde günceller. Kullanılmayan dizgeler kaldırılır ve yeni dizgeler kaynak dizgenin kopyaları olarak eklenir.

İpucu: Yalnızca eski çeviri anahtarlarını kaldırmak istiyorsanız [Çeviri dosyalarını temizleme](#) kullanın.

Ayrıca bakınız:

Weblate, çevirilerin yanında çeviri dosyalarını da günceller mi?

XML çıktısını özelleştirme

4.15 sürümünde geldi.

Eklenti kimliği

`weblate.xml.customize`

Yapılandırma

<code>closing_tags</code>	Boş XML kod imlerine kapanış kod imini ekler
---------------------------	--

Tetikleyiciler

depolamaya yüklendikten sonra

YAML çıktısında satır uzunluğu ve yeni satır gibi davranışların ayarlanmasını sağlar.

YAML çıktısını özelleştirme

3.10.2 sürümünde geldi.

Ekenti kimliği

`weblate.yaml.customize`

Yapılandırma

<code>indent</code>	YAML girintisi	
<code>width</code>	Uzun satırları kaydırma	Kullanılabilir seçenekler: 80 – Satırlar 80 karakterde kaydırılır 100 – Satırlar 100 karakterde kaydırılır 120 – Satırlar 120 karakterde kaydırılır 180 – Satırlar 180 karakterde kaydırılır 65535 – Satırlar kaydırılmaz
<code>line_break</code>	Satır sonları	Kullanılabilir seçenekler: <code>dos</code> – DOS (\r\n) <code>unix</code> – UNIX (\n) <code>mac</code> – MAC (\r)

Tetikleyiciler

depolamaya yüklendikten sonra

YAML çıktısında satır uzunluğu ve yeni satır gibi davranışların ayarlanmasını sağlar.

2.14.2 Ekenti listesini özelleştirme

Ekenti listesi `WEBLATE_ADDONS` tarafından yapılandırılır. Başka bir ekenti eklemek için, bu seçeneğe mutlak sınıf adını eklemeniz yeterlidir.

2.14.3 Ekenti yazma

Kendi eklentilerinizi de yazabilir, ekenti üst verilerini tanımlamak için `weblate.addons.base.BaseAddon` alt sınıfını oluşturabilir ve ardından işlemi yapmak için bir geri çağırma ekleyebilirsiniz.

Ayrıca bakınız:

Ekentileri geliştirmek

2.14.4 Eklentiden betikleri çalıştırma

Eklentiler, dış betikleri yürütmek için de kullanılabilir. Bu özellik eskiden Weblate ile bütünleşikti. Ancak şimdi betikleri bir ekenti ile sarmalamak için bazı kodlar yazmanız gerekiyor.

```
# 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
```

(sonraki sayfaya devam)

```
class ExamplePreAddon(BaseScriptAddon):
    # Event used to trigger the script
    events = (EVENT_PRE_COMMIT,)
    # Name of the addon, has to be unique
    name = "weblate.example.pre"
    # Verbose name and long description
    verbose = _("Execute script before commit")
    description = _("This add-on executes a script.")

    # Script to execute
    script = "/bin/true"
    # File to add in commit (for pre commit event)
    # does not have to be set
    add_file = "po/{{ language_code }}.po"
```

Kurulum ile ilgili bilgi almak için *Özel kalite denetimleri, eklentiler ve kendiliğinden düzeltmeler*.

Betik dosyası, belirtilen bileşen için sürüm denetimi sistemi deposunun kök klasörüne ayarlanmış geçerli klasör ile yürütülür.

Ek olarak, şu ortam değişkenleri kullanılabilir:

WL_VCS

Kullanılan sürüm denetimi sistemi.

WL_REPO

Yukarı akış deposu adresi.

WL_PATH

Sürüm denetimi sistemi deposunun mutlak yolu.

WL_BRANCH

Geçerli bileşende yapılandırılmış depo dalı.

WL_FILEMASK

Geçerli bileşen için dosya maskesi.

WL_TEMPLATE

Tek dilli çeviriler için kalıbın dosya adı (boş olabilir).

WL_NEW_BASE

Yeni çeviriler oluşturmak için kullanılan dosyanın adı (boş olabilir).

WL_FILE_FORMAT

Geçerli bileşende kullanılan dosya biçimi.

WL_LANGUAGE

İşlenmekte olan çevirinin dili (bileşen düzeyinde kancalar için kullanılamaz).

WL_PREVIOUS_HEAD

Güncellemeden sonra önceki HEAD (yalnızca güncelleme sonrası kancası çalıştırıldıktan sonra kullanılabilir).

WL_COMPONENT_SLUG

3.9 sürümünde geldi.

Adresi oluşturmak için kullanılan bileşen kısaltması.

WL_PROJECT_SLUG

3.9 sürümünde geldi.

Adresi oluşturmak için kullanılan proje kısaltması.

WL_COMPONENT_NAME

3.9 sürümünde geldi.

Bileşen adı.

WL_PROJECT_NAME

3.9 sürümünde geldi.

Proje adı.

WL_COMPONENT_URL

3.9 sürümünde geldi.

Bileşen adresi.

WL_ENGAGE_URL

3.9 sürümünde geldi.

Proje etkileşim adresi.

Ayrıca bakınız:

Bileşen yapılandırması

Güncelleme sonrası depo işleme

Sürüm denetimi sistemi yukarı akış kaynağı değiştiğinde çeviri dosyalarını güncellemek için kullanılabilir. Bunu yapabilmek için lütfen Weblate tarafından yalnızca sürüm denetimi sistemine işlenmiş dosyaların görüldüğünü, bu nedenle değişiklikleri betiğin bir parçası olarak yapmanız gerektiğini unutmayın.

Örneğin Gulp ile bunu şu kodu kullanarak yapabilirsiniz:

```
#!/bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Depoya işlenmeden önce çevirilerin işlenmesi

Bir çevirinin depoya kaydedilmesinden önce kendiliğinden değiştirilmesi için işleme betiğini kullanın.

Geçerli bir çevirinin dosya adından oluşan tek bir parametre olarak aktarılır.

2.15 Çeviri belleği

Weblate üzerinde şu yetenekleri olan bir çeviri belleği özelliği bulunur:

- El ile içe aktarılmış çeviri belleği (bilgi almak için *Kullanıcı arayüzü*).
- Weblate üzerinde yapılan ve otomatik olarak kaydedilen çeviriler (*Çeviri belleği kapsamları* kapsamında).
- Geçmiş çeviriler otomatik olarak içe aktarılır.

Çeviri belleğindeki içerik iki yoldan biriyle uygulanabilir:

- El ile, çeviri yapılırken *Kendiliğinden öneriler* görünümü ile.
- Kendiliğinden, dizgeleri *Kendiliğinden çeviri* ya da *Kendiliğinden çeviri* eklentisi ile çevirerek.

Kurulum ipuçları için, varsayılan olarak açık olan *Weblate Çeviri Belleği* bölümüne bakın.

2.15.1 Çeviri belleği kapsamları

3.2 sürümünde geldi: Daha önceki sürümlerde, çeviri belleği yalnızca geçerli içe aktarılmış çeviri belleği kapsamına karşılık gelen bir dosyadan yüklenebiliyordu.

Çeviri belleği kapsamları, istenen davranışa uyacak şekilde çevirilerin hem gizliliğine hem de paylaşılmasına izin vermek için vardır.

İçe aktarılmış çeviri belleği

`import_memory` komutunu kullanarak isteğe bağlı çeviri belleği verilerinin içe aktarılması, bellek içeriğini tüm kullanıcılar ve projeler için kullanılabilir kılar.

Kullanıcıların bireysel çeviri belleği

Tüm kullanıcı çevirilerini otomatik olarak ilgili her kullanıcının kişisel çeviri belleğine kaydeder.

Projelerin bireysel çeviri belleği

Bir projedeki tüm çeviriler, otomatik olarak yalnızca bu proje için kullanılabilen bir proje çeviri belleğine kaydedilir.

Paylaşılan çeviri belleği

Paylaşılan çeviri belleği açık olan projelerdeki tüm çeviriler, tüm projeler için kullanılabilen paylaşılan bir çeviri belleğine kaydedilir.

Ciddi sonuçları olabileceğinden, bu özelliği paylaşılan Weblate kurulumları için açıp açmayacağınızı dikkatlice değerlendirin:

- Çeviriler başkaları tarafından kullanılabilir.
- Bu seçim, gizli bilgilerin açığa çıkmasına yol açabilir.

2.15.2 Çeviri belleğini yönetmek

Kullanıcı arayüzü

3.2 sürümünde geldi.

Temel kullanıcı arayüzünden, her kullanıcı için ve her proje için çeviri belleklerini yönetebilirsiniz. Çeviri belleğini indirme, silme ya da içe aktarma işlemleri yapılabilir.

İpucu: JSON çeviri belleği Weblate içine aktarılabilir. Diğer araçlarla birlikte çalışabilirlik için TMX biçimi de sağlanır.

Ayrıca bakınız:

Weblate çeviri belleği şeması

testuser / Translation memory

Translation memory status ⓘ

Number of your entries	0	Download as JSON	Download as TMX	Delete
Total number of entries	0			

Import translation memory

File

Choose File No file chosen

You can upload a TMX or JSON file.

Upload

Powered by Weblate 4.17 About Weblate Legal Contact Documentation Donate to Weblate

Yönetim arayüzü

Çeviri belleği içeriğini değiştirmek için çeşitli yönetim komutları bulunur. Bunlar, kapsamlara göre süzülmeden bir bütün olarak çeviri belleği üzerinde çalışır (parametreler ile istenmedikçe):

`dump_memory`

Çeviri belleğini JSON dosyası olarak dışa aktarır

`import_memory`

TMX ya da JSON dosyalarını çeviri belleği içine aktarır

2.16 Yapılandırma

Tüm ayarlar `settings.py` dosyasında (Django için her zamanki gibi) tutulur.

Not: Bu ayarlardan herhangi birini değiştirdikten sonra, Weblate WSGI ve Celery işlemlerini yeniden başlatmanız gerekir.

`mod_wsgi` olarak çalıştırılması durumunda, yapılandırmayı yeniden yüklemek için Apache sunucusunu yeniden başlatmanız gerekir.

Ayrıca bakınız:

Django yapılandırma parametreleri için [Django belgelerine](#) bakabilirsiniz.

2.16.1 AKISMET_API_KEY

Weblate, istenmeyen ileti olarak gelen anonim önerileri denetlemek için Akismet kullanabilir. Bir API anahtarı satın almak ve bunu bir siteyle ilişkilendirmek için 'akismet.com <<https://akismet.com/>>' adresine bakabilirsiniz.

2.16.2 ANONYMOUS_USER_NAME

Oturum açmamış kullanıcılar için kullanıcı adı.

Ayrıca bakınız:

Erişim denetimi

2.16.3 AUDITLOG_EXPIRY

3.6 sürümünde geldi.

Weblate hesap etkinliği ile ilgili bilgi içeren denetim günlüklerinin tutulacağı gün sayısı.

Varsayılan değer: 180 gün.

2.16.4 AUTH_LOCK_ATTEMPTS

Hız sınırlaması uygulanmadan önce başarısız olmasına izin verilecek en fazla kimlik doğrulama girişimi sayısı.

Bu seçenek şu anda şu konumlarda uygulanmaktadır:

- Oturum aç. Hesap parolasını silerek kullanıcının yeni bir parola isteği yapmadan oturum açmasını engeller.
- Parola sıfırlama. Çok fazla parola sıfırlama girişimi ile istenmeyen trafik oluşturan kullanıcılara yeni e-posta gönderilmesini engeller.

Varsayılan değer: 10.

Ayrıca bakınız:

Hızı sınırlamak

2.16.5 AUTO_UPDATE

3.2 sürümünde geldi.

3.11 sürümünde değişti: Özgün açma/kapama seçeneği, hangi dizgelerin kabul edildiğini ayırt edecek biçimde değiştirildi.

Tüm depoları günlük olarak günceller.

İpucu: Weblate depolarını kendiliğinden güncellemek için *Bildirim kancaları* kullanmıyorsanız yararlıdır.

Not: Geriye dönük uyumluluk için açma/kapama seçenekleri dizge seçimine ek olarak bulunur.

Seçenekler:

"none"

Günlük güncelleme yapılmaz.

"remote" ayrıca False

Yalnızca uzak tarafı günceller.

"full" ayrıca True

Uzak tarafı günceller ve çalışan kopya ile birleştirir.

Not: Bunun için *Celery ile arka plan görevlerini kullanmak* çalışıyor olmalıdır. Yeniden başlatıldıktan sonra etkili olur.

2.16.6 AVATAR_URL_PREFIX

Avatar adreslerini şu biçimde oluşturmak için ön ek: `${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`. Şu hizmetlerin çalıştığı biliniyor:

Gravatar (varsayılan), <https://gravatar.com/> üzerinden

```
AVATAR_URL_PREFIX = 'https://www.gravatar.com/'
```

Libravatar, <https://www.libravatar.org/> üzerinden

```
AVATAR_URL_PREFIX = 'https://www.libravatar.org/'
```

Ayrıca bakınız:

Avatar ön belleği, `ENABLE_AVATARS`, [Avatarlar](#)

2.16.7 AUTH_TOKEN_VALID

Kimlik doğrulama belirticinin ve parola sıfırlama e-postalarındaki geçici parolanın geçerli olacağı süre. Varsayılan olarak 172800 saniye (2 gün).

2.16.8 AUTH_PASSWORD_DAYS

Bir kullanıcının daha önce kullandığı bir parolayı yeniden kullanabilmesi için geçmesi gereken gün sayısı.

Bu işlem denetim günlüğü ile yapılır. `AUDITLOG_EXPIRY` seçeneğinin değeri en az bu değer ile aynı olmalıdır.

Not: Weblate 2.15 sürümünden önce yapılan parola değişiklikleri bu ilkede dikkate alınmaz.

Varsayılan değer: 180 gün.

2.16.9 AUTOFIX_LIST

Bir çeviri kaydedilirken uygulanacak kendiliğinden düzeltmelerin listesi.

Not: Kendiliğinden düzeltme arayüzünü uygulayan Python sınıfının tam yolunu belirtin.

Kullanılabilecek düzeltmeler:

`weblate.trans.autofixes.whitespace.SameBookendingWhitespace`

Çevirinin başındaki ve sonundaki boşlukları kaynak ile eşleştirir.

`weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis`

Kaynakta üç nokta karakteri (...) varsa çevirinin sonundaki üç ayrı noktayı (...) değiştirir.

`weblate.trans.autofixes.chars.RemoveZeroSpace`

Kaynakta bulunmuyorsa çevirideki sıfır genişlikli boşluk karakterlerini kaldırır.

`weblate.trans.autofixes.chars.RemoveControlChars`

Kaynakta bulunmuyorsa denetim karakterlerini kaldırır.

`weblate.trans.autofixes.chars.DevanagariDanda`

Bangla yazımındaki cümle sonunu Devanagari danda ile değiştirir.

`weblate.trans.autofixes.html.BleachHTML`

Güvenli olmayan HTML kod imlerini `safe-html` olarak imlenmiş çevirilerden kaldırır (bilgi almak için: [Güvenli olmayan HTML](#)).

Hangilerinin kullanılacağını seçebilirsiniz:

```
AUTOFIX_LIST = (  
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",  
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",  
)
```

Ayrıca bakınız:

Kendiliğinden düzeltmeler, Özel kendiliğinden düzeltmeler

2.16.10 BACKGROUND_TASKS

4.5.2 sürümünde geldi.

Bir bileşen için uzun bakım görevlerinin tetiklenme sıklığını belirtir.

Şu anda bunun denetlediği:

- *Kendiliğinden çeviri* eklentisi
- *Denetimler ve düzeltmeler* yeniden hesaplama

Olası seçenekler:

- aylık (varsayılan)
- weekly
- daily
- never

Not: Weblate üzerinde binlerce bileşen bulunuyorsa sıklığın artırılması önerilmez.

2.16.11 BASIC_LANGUAGES

4.4 sürümünde geldi.

Kullanıcılara yeni çeviriye başlamaları için sunulacak dillerin listesi. Belirtilmediğinde, yaygın olarak kullanılan tüm dilleri içeren, ancak ülkeye özgü lehçeleri olmayan iç liste kullanılır.

Bu seçenek, yalnızca yetki verilmemiş kullanıcıların istenmeyen diller eklemesini engeller. Proje yöneticilerine Weblate üzerinde tanımlanmış dillerin tümü sunulur.

Not: Bu seçenek, Weblate üzerinde yeni diller tanımlamaz, yalnızca veri tabanında var olan dilleri süzer.

Örnek:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

Ayrıca bakınız:

Dil tanımları

2.16.12 BORG_EXTRA_ARGS

4.9 sürümünde geldi.

İç yedeklemeler tetiklendiğinde **borg create** komutuna ek parametreler iletebilirsiniz.

Örnek:

```
BORG_EXTRA_ARGS = ["--exclude", "vcs/"]
```

Ayrıca bakınız:

Weblate yedeğini alma ve taşıma, `borg create`

2.16.13 CACHE_DIR

4.16 sürümünde geldi.

Weblate ön bellek dosyalarının tutulacağı klasör. Varsayılan olarak `DATA_DIR` içindeki `cache` alt klasörü kullanılır.

`DATA_DIR` bir ağ dosya sisteminde bulunuyorsa bunu yerel ya da geçici bir dosya sistemi olarak değiştirin.

Docker kapsayıcısı bunun için ayrı bir birim kullanır. Bilgi almak için: `:ref:`docker-volume``.

2.16.14 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Weblate için Content-Security-Policy üst bilgisini özelleştirin. Başlık, üçüncü taraf hizmetlerle (Matomo, Google Analytics, Sentry, ...) etkin bütünleştirmelere göre kendiliğinden oluşturulur.

Tüm bunlar varsayılan olarak boş listedir.

Örnek:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Ayrıca bakınız:

İçerik güvenliği ilkesi, İçerik güvenliği ilkesi (CSP)

2.16.15 CHECK_LIST

Bir çeviride yapılacak hata denetimlerinin listesi.

Not: Denetim arayüzünü uygulayan Python sınıfının tam yolunu yazın.

Denetim listesini, isteğinize göre ayarlayın.

Tüm iç *Kalite denetimleri* varsayılan olarak açıktır ve buradan bu ayarları değiştirebilirsiniz. Varsayılan olarak, kullanılacak değerler *Örnek yapılandırma* dosyası içinde yorum satırı olarak bulunurlar. Daha sonra her yeni Weblate sürümü için yeni denetimler yapıldı.

Tüm denetimleri kapatabilirsiniz:

```
CHECK_LIST = ()
```

Yalnızca birkaçını açabilirsiniz:

```
CHECK_LIST = (  
    "weblate.checks.chars.BeginNewlineCheck",  
    "weblate.checks.chars.EndNewlineCheck",  
    "weblate.checks.chars.MaxLengthCheck",  
)
```

Not: Bu ayarın değiştirilmesi yalnızca yeni değiştirilen çevirileri etkiler. Var olan denetimlerin veri tabanında tutulması sürdürülür. Değişiklikleri kaydedilmiş çevirilere de uygulamak için `updatechecks` komutunu yürütün.

Ayrıca bakınız:

Kalite denetimleri, Parametreleri kullanarak davranışı özelleştirmek

2.16.16 COMMENT_CLEANUP_DAYS

3.6 sürümünde geldi.

Belirli günden sonra yorumları siler. Varsayılan değer Yok. Yani bir silme işlemi yapılmaz.

2.16.17 COMMIT_PENDING_HOURS

Bekleyen değişikliklerin gönderilmesi ile arka plan görevi tarafından işlenmesi arasındaki saat sayısı.

Ayrıca bakınız:

Bileşen yapılandırması, İşlenecek değişikliklerin yaşı, Bakım görevlerini yürütmek, `commit_pending`

2.16.18 CONTACT_FORM

4.6 sürümünde geldi.

İletişim formu e-postalarının nasıl gönderileceğini yapılandırır. E-posta sunucusu yapılandırmanıza uygun bir yapılandırma seçin.

"reply-to"

Gönderici *Yanıt* olarak kullanılır. Varsayılan davranış budur.

"from"

Gönderici *Kimden* olarak kullanılır. E-posta sunucunuz bu tür e-postaların gönderilmesine izin vermelidir.

2.16.19 DATA_DIR

Weblate klasöründe tüm veriler bulunur. Sürüm denetimi sistemi depo bağlantılarını, tam metin dizinini ve dış araçlar için çeşitli yapılandırma dosyalarını içerir.

Genellikle aşağıdaki alt klasörler de bulunur:

home

Betikleri çağırmak için kullanılan giriş klasörü.

ssh

SSH anahtarları ve yapılandırması.

static

Durağan Django dosyalarının varsayılan konumu `STATIC_ROOT` tarafından belirlenir. Bilgi almak için: `:ref:`static-files``.

Docker kapsayıcısı bunun için ayrı bir birim kullanır. Bilgi almak için: `:ref:`docker-volume``.

media

Django ortam dosyalarının varsayılan konumu `MEDIA_ROOT` tarafından belirlenir. Yüklenen ekran görüntülerini içerir. Bilgi almak için: [:ref:`screenshots`](#).

vcs

Çeviriler için sürüm denetimi depoları.

backups

Günlük yedekleme verileri. Bilgi almak için: [Yedeklemeler için dökümü yapılan veriler](#).

fonts:

Kullanıcı tarafından yüklenen yazı tipleri. Bilgi almak için: [:ref:`fonts`](#).

cache

Çeşitli ön bellekler, `CACHE_DIR` seçeneği kullanılarak başka bir konuma taşınabilir.

Docker kapsayıcısı bunun için ayrı bir birim kullanır. Bilgi almak için: [:ref:`docker-volume`](#).

Not: Bu klasör Weblate tarafından yazılabilir olmalıdır. uWSGI olarak çalıştırıldığında, `www-data` kullanıcısının bu klasöre yazma erişimi olmalıdır.

Bunu yapmanın en kolay yolu, kullanıcıyı klasörün sahibi yapmaktır:

```
sudo chown www-data:www-data -R $DATA_DIR
```

Varsayılan değer: `/home/weblate/data`. Ancak yapılandırılması beklenir.

Ayrıca bakınız:

[Dosya sistemi izinleri](#), [Weblate yedeğini alma ve taşıma](#), [CACHE_DIR](#)

2.16.20 DATABASE_BACKUP

3.1 sürümünde geldi.

Veri tabanı yedeklemelerinin düz metin ya da sıkıştırılmış olarak depolanacağını ya da depolanmayacağını belirler. Şu değerler kullanılabilir:

- "plain"
- "compressed"
- "none"

Ayrıca bakınız:

[Weblate yedeğini alma ve taşıma](#)

2.16.21 DEFAULT_ACCESS_CONTROL

3.3 sürümünde geldi.

Yeni projeler için varsayılan erişim denetimi ayarı:

0

Herkese açık

1

Korunmuş

100

Kişisel

200

Özel

Erişim denetimi listesini el ile yönetiyorsanız :guilabel:`Özel` olarak ayarlayın. Bu seçenek iç Weblate yönetimine güvenilmemesi anlamına gelir.

Ayrıca bakınız:

Proje erişim denetimi, Erişim denetimi

2.16.22 DEFAULT_AUTO_WATCH

4.5 sürümünde geldi.

Katkıda bulunulan projeler kendiliğinden izlensin seçeneğinin yeni kullanıcılar için açık olup olmayacağını belirler. Varsayılan değer: True.

Ayrıca bakınız:

Bildirimler

2.16.23 DEFAULT_RESTRICTED_COMPONENT

4.1 sürümünde geldi.

Varsayılan bileşen kısıtlaması değeri.

Ayrıca bakınız:

Kısıtlı erişim, Takımların kapsamı

2.16.24 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DEFAULT_COMMIT_MESSAGE, DEFAULT_DELETE_MESSAGE, DEFAULT_MERGE_MESSAGE

Farklı işlemler için varsayılan işleme iletileri. Bilgi almak için: ref: *component*.

Ayrıca bakınız:

Kalıp kodlaması, Bileşen yapılandırması, İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri

2.16.25 DEFAULT_ADDONS

Oluşturulan her bileşene varsayılan olarak kurulacak eklentiler.

Not: Bu ayar yalnızca yeni oluşturulan bileşenleri etkiler.

Örnek:

```
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"],
    },
}
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    "threshold": "80",
  },
}
```

Ayrıca bakınız:*install_addon, Eklentiler, WEBLATE_ADDONS***2.16.26 DEFAULT_COMMITER_EMAIL**

Varsayılan işleyenin e-posta adresi: noreply@weblate.org.

Ayrıca bakınız:*DEFAULT_COMMITER_NAME***2.16.27 DEFAULT_COMMITER_NAME**

Varsayılan işleyenin adı: Weblate.

Ayrıca bakınız:*DEFAULT_COMMITER_EMAIL***2.16.28 DEFAULT_LANGUAGE**

4.3.2 sürümünde geldi.

Kaynak dil içinde örnek olarak kullanılacak varsayılan kaynak dil.

Varsayılan değer: *en*. Eşleşen dil nesnesinin veri tabanında bulunması gerekir.

Ayrıca bakınız:*Dil tanımları, Kaynak dil***2.16.29 DEFAULT_MERGE_STYLE**

3.4 sürümünde geldi.

Tüm yeni bileşenler için birleştirme biçemi.

- *rebase* - default
- *merge*

Ayrıca bakınız:*Bileşen yapılandırması, Birleştirme biçemi*

2.16.30 DEFAULT_SHARED_TM

3.2 sürümünde geldi.

Paylaşılan çeviri belleği kullanılsın ve Paylaşılan çeviri belleğine katkıda bulunulsun varsayılan değerlerini yapılandırır.

2.16.31 DEFAULT_TRANSLATION_PROPAGATION

Varsayılan çeviri yayılması ayarı. Varsayılan değer `True`.

Ayrıca bakınız:

Bileşen yapılandırması, Çevirilerin yayılmasını sağlamak

2.16.32 DEFAULT_PULL_MESSAGE

Çekme isteklerinin varsayılan başlığını ve iletisini yapılandırır.

2.16.33 ENABLE_AVATARS

Kullanıcıların Gravatar avatarlarının kullanılıp kullanılmayacağını belirler. Varsayılan olarak kullanılabilir.

Avatarlar sunucu tarafından alınır ve ön belleğe eklenir. Böylece kişisel bilgilerin sızma riski azaltılır ve kullanıcı deneyimi hızlandırılır.

Ayrıca bakınız:

Avatar ön belleği, AVATAR_URL_PREFIX, Avatarlar

2.16.34 ENABLE_HOOKS

Anonim uzak kancaların etkinleştirilmesini sağlar.

Ayrıca bakınız:

Bildirim kancaları

2.16.35 ENABLE_HTTPS

HTTPS ya da HTTP olarak kullanılacak Weblate bağlantısı iletişim kuralı. Bu seçenek, gönderilen e-postaları ve oluşturulan mutlak adresleri etkiler.

Varsayılan yapılandırmada bu seçenek, HTTPS ile ilgili çeşitli Django ayarları için de kullanılır. Güvenli çerezleri etkinleştirir, HSTS seçeneğini açar ya da HTTPS adresine yönlendirmeyi etkinleştirir.

HTTPS yönlendirmesi bazı durumlarda sorunlu olabilir ve iletişim kuralı üst bilgilerini Django üzerine doğru şekilde iletmeyen SSL sonlandırma işlemi yapan ters bir vekil sunucu kullanmanız durumunda, sonsuz yönlendirme sorunu yaşayabilirsiniz. Lütfen ters vekil sunucu yapılandırmanızı `X-Forwarded-Proto` veya `Forwarded` üst bilgilerini yayacak şekilde ayarlayın ya da SSL durumunun Django tarafından doğru bir şekilde algılanmasını sağlamak için `SECURE_PROXY_SSL_HEADER` seçeneğini yapılandırın.

Ayrıca bakınız:

`SESSION_COOKIE_SECURE`, `CSRF_COOKIE_SECURE`, `SECURE_SSL_REDIRECT`, `SECURE_PROXY_SSL_HEADER` *Doğru site etki alanını ayarlayın*

2.16.36 ENABLE_SHARING

Kullanıcıların çeviri ilerlemesini sosyal ağlarda paylaşabilmesi için *Paylaş* menüsünü açar ya da kapatır.

2.16.37 EXTRA_HTML_HEAD

4.15 sürümünde geldi.

HTML üst bilgisine ek biçimlendirme ekler. Sitenin sahipliğini doğrulamak için kullanılabilir. Örneğin:

```
EXTRA_HTML_HEAD = '<link href ="https://fosstodon.org/@weblate" rel ="me">'
```

Uyarı: Dizge üzerinde herhangi bir temizleme işlemi yapılmaz. HTML üst bilgisine olduğu gibi eklenir.

2.16.38 GET_HELP_URL

4.5.2 sürümünde geldi.

Weblate kopyanızın destek bilgilerinin bulunduğu adres.

2.16.39 GITEA_CREDENTIALS

4.12 sürümünde geldi.

Gitea sunucuları için kimlik doğrulama bilgileri listesi.

```
GITEA_CREDENTIALS = {
  "try.gitea.io": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitea.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Ayrıca bakınız:

Gitea çekme isteği, Gitea kişisel erişim kodu oluşturmak

2.16.40 GITLAB_CREDENTIALS

4.3 sürümünde geldi.

GitLab sunucuları için kimlik doğrulama bilgileri listesi.

```
GITLAB_CREDENTIALS = {
  "gitlab.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitlab.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
} ,  
}
```

Ayrıca bakınız:

GitLab birleştirme istekleri, GitLab: Kişisel erişim kodu

2.16.41 GITHUB_CREDENTIALS

4.3 sürümünde geldi.

GitHub sunucuları için kimlik doğrulama bilgileri listesi.

```
GITHUB_CREDENTIALS = {  
  "api.github.com": {  
    "username": "weblate",  
    "token": "your-api-token",  
  },  
  "github.example.com": {  
    "username": "weblate",  
    "token": "another-api-token",  
  },  
}
```

Ayrıca bakınız:

GitHub çekme istekleri, GitHub kişisel erişim kodu oluşturmak

2.16.42 BITBUCKETSERVER_CREDENTIALS

4.16 sürümünde geldi.

Bitbucket sunucuları için kimlik doğrulama bilgileri listesi.

```
BITBUCKETSERVER_CREDENTIALS = {  
  "git.self-hosted.com": {  
    "username": "weblate",  
    "token": "http-access-token",  
  },  
}
```

Ayrıca bakınız:

Bitbucket sunucusu çekme isteği, Bitbucket: HTTP erişim kodu

2.16.43 GOOGLE_ANALYTICS_ID

Google Analytics ile Weblate izlemesini açmak için Google Analytics kimliği.

2.16.44 HIDE_REPO_CREDENTIALS

Depo kimlik doğrulama bilgilerini site arayüzünde gizler. İçinde kullanıcı adı ve parolası geçen bir depo adresiniz varsa, Weblate ilgili bilgileri kullanıcılara görüntülerken bunları gizler.

Örneğin, ``https://kullanici:parola@git.ornek.com/depo.git`` yerine yalnızca ``https://git.ornek.com/depo.git`` görüntülenir. Sürüm denetimi sistemi hata iletilerinde de benzer ayıklama yapılmaya çalışılır.

Not: Bu seçenek varsayılan olarak kapalıdır.

2.16.45 HIDE_VERSION

4.3.1 sürümünde geldi.

Sürüm bilgilerini kimliği doğrulanmamış kullanıcılardan gizler. Bu seçenek aynı zamanda tüm belge bağlantılarının kurulu sürüm belgeleri yerine son sürümün belgelerinin göstermesini sağlar.

Sürümün gizlenmesi bazı kuruluşlar için önerilen bir güvenlik önlemidir. Ancak saldırganın uygulamanın davranışını inceleyerek sürümü öğrenmesini engellemez.

Not: Bu seçenek varsayılan olarak kapalıdır.

2.16.46 INTERLEDGER_PAYMENT_POINTERS

4.12.1 sürümünde geldi.

Web üzerinden para kazanmak için genel muhasebe ödeme göstergeleri (ILP) listesi.

Birden fazla belirtilirse, rastgele biri seçilerek olasılıksal gelir paylaşımı sağlanır.

Ayrıntılı bilgi almak için <<https://webmonetization.org/>> sitesine bakabilirsiniz.

İpucu: Varsayılan değer, kullanıcıların Weblate için fon sağlamasına yardımcı olur.

2.16.47 IP_BEHIND_REVERSE_PROXY

Weblate platformunun ters vekil sunucu arkasında çalışıp çalışmadığını gösterir.

True olarak ayarlanırsa, Weblate IP adresini `IP_PROXY_HEADER` seçeneğindeki bir üst bilgidен alır.

Uyarı: Gerçekten bir ters vekil sunucu kullandığınızdan ve bu üst bilginin ayarlandığından emin olun. Yoksa kullanıcılar IP adresini taklit edebilir.

Not: Bu seçenek, varsayılan olarak kapalıdır.

Ayrıca bakınız:

Ters vekil sunucu arkasında çalıştırmak, Hızı sınırlamak, `IP_PROXY_HEADER`, `IP_PROXY_OFFSET`

2.16.48 IP_PROXY_HEADER

`IP_BEHIND_REVERSE_PROXY` seçeneği açıldığında hangi Weblate üst bilgisinde IP adresinin bulunduğunu gösterir.

Varsayılan değer: `HTTP_X_FORWARDED_FOR`.

Ayrıca bakınız:

Ters vekil sunucu arkasında çalıştırmak, Hızı sınırlamak, SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY, IP_PROXY_OFFSET

2.16.49 IP_PROXY_OFFSET

`:setting:'IP_PROXY_HEADER'` ögesinin istemci IP adresi olarak kullanılacak bölümünü gösterir.

Kurulumunuza bağlı olarak, bu üst bilgi birkaç IP adresinden oluşabilir (örneğin, `X-Forwarded-For: a, b, client-ip`) ve üst bilgiden hangi adresin istemci IP adresi olarak kullanılacağını bu bölümden yapılandırabilirsiniz.

Uyarı: Bunu ayarlamak kurulumunuzun güvenliğini etkiler. Yalnızca IP adresini belirlemek için güvenilen vekil sunucuları kullanacak şekilde yapılandırmanız gerekir.

Varsayılan değer: 0.

Ayrıca bakınız:

Ters vekil sunucu arkasında çalıştırmak, Hızı sınırlamak, SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY, IP_PROXY_HEADER

2.16.50 LEGAL_TOS_DATE

4.15 sürümünde geldi.

Not: Bunun çalışması için *Yasal* kurulmuş olmalıdır.

Hizmet koşulları belgelerinin son güncellenme tarihi. Tarih değiştiğinde, kullanıcıların hizmet koşullarını kabul etmeleri gerekir.

```
from datetime import date

LEGAL_TOS_DATE = date(2022, 2, 2)
```

2.16.51 LEGAL_URL

3.5 sürümünde geldi.

Weblate kopyanızın yasal belgelerinin bulunduğu adres.

İpucu: Yasal belgelerinizi Weblate dışında bulunduruyor ve Weblate içine eklemek istiyorsanız yararlıdır. Bilgi almak için: *Yasal*.

Örnek:

```
LEGAL_URL = "https://weblate.org/terms/"
```

Ayrıca bakınız:[*PRIVACY_URL*](#)

2.16.52 LICENSE_EXTRA

Lisans seçeneklerine katılacak ek lisanslar.

Not: Her lisans tanımında bir kısa ad, bir uzun ad ve bir adres olmalıdır.

Örnek:

```
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

4.3 sürümünde değişti: Bu seçenek boş olarak ayarlandığında lisans uyarısını kapatır.

Görüntülenecek lisans listesini süzer. Bu seçenek, boş olarak ayarlandığında lisans uyarısını da kapatır.

Not: Bu süzgeç kısa lisans adlarını kullanır.

Örnek:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

Şu lisans uyarısını kapatır:

```
LICENSE_FILTER = set()
```

Ayrıca bakınız:

[alerts](#)

2.16.54 LICENSE_REQUIRED

Bileşen yapılandırması içindeki lisans özneliğinin zorunlu olup olmadığını belirler.

Not: Bu seçenek varsayılan olarak kapalıdır.

2.16.55 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Belirtilen bir çevirinin uzunluğunun sınırlı olmasının gerekip gerekmediği. Kısıtlama, kaynak dizgenin uzunluğu x 10 karakterdir.

İpucu: Kaynak dizge uzunluğuna bakılmaksızın daha uzun çevrilere (10.000 karaktere kadar) izin vermek için bu seçeneği `False` olarak ayarlayın.

Not: Varsayılan değer: `True`.

2.16.56 LOCALIZE_CDN_URL ve LOCALIZE_CDN_PATH

Bu ayarlar *JavaScript yerelleştirme CDN* eklentisini yapılandırır. `LOCALIZE_CDN_URL`, yerelleştirme CDN sunucusunun kullanılabildiği kök adresini belirler ve `LOCALIZE_CDN_PATH` seçeneği, `LOCALIZE_CDN_URL` adresinde sunulan oluşturulmuş dosyaların kaydedileceği yolu belirler.

İpucu: Hosted Weblate üzerinde bu seçenek, `https://weblate-cdn.com/` adresini kullanır.

Ayrıca bakınız:

JavaScript yerelleştirme CDN

2.16.57 LOGIN_REQUIRED_URLS

Oturum açmanın dayatılmasını istediğiniz adreslerin listesi. (Weblate üzerindeki iç standart kuralların yanı sıra).

İpucu: Bu seçenek, şunu kullanarak tüm kurulumu parola ile korumanızı sağlar:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

İpucu: Yukarıdaki örnekte gösterildiği gibi API erişiminin de kilitlenmesi istenir.

Ayrıca bakınız:

`REQUIRE_LOGIN`

2.16.58 LOGIN_REQUIRED_URLS_EXCEPTIONS

`LOGIN_REQUIRED_URLS` için istisna listesi. Belirtilmediğinde kullanıcıların oturum açma sayfasına erişmesine izin verilir.

Ekleme isteyebileceğiniz bazı istisnalar:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r"/accounts/(.*)$", # Required for sign in
    r"/static/(.*)$", # Required for development mode
    r"/widgets/(.*)$", # Allowing public access to widgets
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
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

İzlemek istediğiniz Matomo (eski adıyla Piwik) üzerindeki bir site kimliği.

Not: Bu bütünleştirme Matomo Etiket Yöneticisi özelliğini desteklemez.

Ayrıca bakınız:

MATOMO_URL

2.16.60 MATOMO_URL

Weblate kullanımını izlemek için kullanmak istediğiniz bir Matomo (eski adıyla Piwik) kurulumunun tam adresi (sondaki bölü karakteri ile birlikte). Bilgi almak için: <<https://matomo.org/>>.

İpucu: Bu bütünleştirme Matomo Etiket Yöneticisi özelliğini desteklemez.

Örnek:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

Ayrıca bakınız:

MATOMO_SITE_ID

2.16.61 NEARBY_MESSAGES

Çevrilen dizgenin yakınında görüntülenecek dizge sayısı. Bu yalnızca varsayılan bir değerdir. Kullanıcılar bu değeri *Kullanıcı profili* bölümünden ayarlayabilir.

2.16.62 DEFAULT_PAGE_LIMIT

4.7 sürümünde geldi.

Sayfalama etkin olduğunda bir sayfada varsayılan olarak görüntülenecek öge sayısı.

2.16.63 PAGURE_CREDENTIALS

4.3.2 sürümünde geldi.

Pagure sunucuları için kimlik doğrulama bilgilerinin listesi.

```
PAGURE_CREDENTIALS = {
  "pagure.io": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "pagure.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Ayrıca bakınız:

Pagure birleştirme istekleri, *Pagure API*

2.16.64 PRIVACY_URL

4.8.1 sürümünde geldi.

Weblate kopyanızın gizlilik ilkesinin bulunduğu adres.

İpucu: Yasal belgelerinizi Weblate dışında bulunduruyor ve Weblate içine eklemek istiyorsanız yararlıdır. Bilgi almak için: *Yasal*.

Örnek:

```
PRIVACY_URL = "https://weblate.org/terms/"
```

Ayrıca bakınız:

LEGAL_URL

2.16.65 PRIVATE_COMMIT_EMAIL_OPT_IN

4.15 sürümünde geldi.

Kişisel işleme e-postasının türünü opt-in ya da opt-out olarak yapılandırır (varsayılan değer: opt-out).

Ayrıca bakınız:

Profil, *PRIVATE_COMMIT_EMAIL_TEMPLATE*

2.16.66 PRIVATE_COMMIT_EMAIL_TEMPLATE

4.15 sürümünde geldi.

Bir kullanıcının kişisel işleme e-postasını oluşturan kalıp. Varsayılan değer: "{username}@users.noreply.{site_domain}".

Devre dışı bırakılması için boş bırakın.

Not: Farklı işleme e-postası kullanmak, *PRIVATE_COMMIT_EMAIL_OPT_IN* seçeneği ile yapılandırılmadığı sürece kullanıcılar için abone olmak anlamına gelir. Kullanıcılar işleme e-postasını *Profil* içinde yapılandırabilir.

2.16.67 PROJECT_BACKUP_KEEP_COUNT

4.14 sürümünde geldi.

Her proje için sunucuda tutulacak yedek sayısını belirler. Varsayılan değer: 3.

Ayrıca bakınız:

Proje düzeyinde yedekler

2.16.68 PROJECT_BACKUP_KEEP_DAYS

4.14 sürümünde geldi.

Proje yedeklerinin sunucuda tutulacağını süreyi belirler. Varsayılan değer: 30 gün.

Ayrıca bakınız:

Proje düzeyinde yedekler

2.16.69 PROJECT_NAME_RESTRICT_RE

4.15 sürümünde geldi.

Proje adlandırmasını kısıtlayacak kurallı ifadeyi belirler. Kurallı ifadeye uyan adlar reddedilir.

Ayrıca bakınız:

Proje adı

2.16.70 PROJECT_WEB_RESTRICT_HOST

4.16.2 sürümünde geldi.

Proje sitesinde belirli sunucuların kullanmasını reddeder. Tüm alt etki alanları eşleşir. Bu nedenle `example.com` katıldığında `test.example.com` alt etki alanı da engellenir. Listede yalnızca küçük harfli dizgeler bulunmalıdır. Ayırıştırılan etki alanı eşleştirmeden önce küçük harflidir.

Varsayılan yapılandırma:

```
PROJECT_WEB_RESTRICT_HOST = {"localhost"}
```

Ayrıca bakınız:

Proje sitesi `PROJECT_WEB_RESTRICT_NUMERIC`, `PROJECT_WEB_RESTRICT_RE`,

2.16.71 PROJECT_WEB_RESTRICT_NUMERIC

4.16.2 sürümünde geldi.

Proje sitesinde sayısal IP adresi kullanarak reddedilir. Varsayılan olarak etkindir.

Ayrıca bakınız:

Proje sitesi `PROJECT_WEB_RESTRICT_HOST`, `PROJECT_WEB_RESTRICT_RE`,

2.16.72 PROJECT_WEB_RESTRICT_RE

4.15 sürümünde geldi.

Proje sitelerini kısıtlayacak kurallı ifadeyi belirler. Kurallı ifadeye uyan adresler reddedilir.

Ayrıca bakınız:

Proje sitesi `PROJECT_WEB_RESTRICT_HOST`, `PROJECT_WEB_RESTRICT_NUMERIC`

2.16.73 RATELIMIT_ATTEMPTS

3.2 sürümünde geldi.

Hız sınırlaması uygulanmadan önce yapılabilecek en fazla kimlik doğrulama girişimi sayısı.

Varsayılan değer: 5.

Ayrıca bakınız:

Hızı sınırlamak, `RATELIMIT_WINDOW`, `RATELIMIT_LOCKOUT`

2.16.74 RATELIMIT_WINDOW

3.2 sürümünde geldi.

Hız sınırlaması uygulandıktan sonra kimlik doğrulamasının kabul edileceği süre.

Varsayılan değer: 300 saniye (5 dakika).

Ayrıca bakınız:

Hızı sınırlamak, `RATELIMIT_ATTEMPTS`, `RATELIMIT_LOCKOUT`

2.16.75 RATELIMIT_LOCKOUT

3.2 sürümünde geldi.

Hız sınırlaması uygulandıktan sonra kimlik doğrulamasının kilitleneceği süre.

Varsayılan değer: 600 saniye (10 dakika).

Ayrıca bakınız:

Hızı sınırlamak, `RATELIMIT_ATTEMPTS`, `RATELIMIT_WINDOW`

2.16.76 REGISTRATION_ALLOW_BACKENDS

4.1 sürümünde geldi.

Hesap açılmasına izin verilecek kimlik doğrulama arka uçlarının listesi. Bu seçenek yalnızca yeni kayıtları sınırlar. Kullanıcılar yapılandırılmış tüm kimlik doğrulama arka uçları ile kimlik doğrulaması yapabilir ve kimlik doğrulaması ekleyebilir.

Hesap açma arka uçlarını sınırlarken `REGISTRATION_OPEN` seçeneğinin etkin tutulması önerilir. Yoksa kullanıcılar hesap açabilir, ancak Weblate kullanıcı arayüzüne hesap açma bağlantılarını görüntüleyemez.

Örnek:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

İpucu: Arka uç adları, kimlik doğrulaması için adres içinde kullanılan adlarla eşleşir.

Ayrıca bakınız:

REGISTRATION_OPEN, Kimlik doğrulama

2.16.77 REGISTRATION_CAPTCHA

Yeni hesap açılışlarının CAPTCHA tarafından korunup korunmadığını gösteren `True` ya da `False` değeri. Bu ayar isteğe bağlıdır ve belirtilmezse varsayılan olarak `True` olduğu varsayılır.

Açıkça, kullanıcıların e-posta adreslerini yazdığı tüm sayfalara bir CAPTCHA eklenir:

- Yeni hesap açılışı.
- Parola kurtarma.
- Bir hesaba e-posta ekleme.
- Oturum açmamış kullanıcılar için iletişim formu.

2.16.78 REGISTRATION_EMAIL_MATCH

Hangi e-posta adreslerinin kaydedilebileceğini süzmenizi sağlar.

Varsayılan değer: `. *`. Herhangi bir e-posta adresi ile hesap açılmasına izin verir.

Bu seçeneği, hesap açma işlemi tek bir e-posta etki alanıyla sınırlamak için kullanabilirsiniz:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.79 REGISTRATION_OPEN

Şu anda yeni hesap açılmasına izin verilip verilmediği. Bu ayar isteğe bağlıdır ve varsayılan `True` değerinde kalabileceği gibi `False` olarak da değiştirilebilir.

Bu ayar, e-posta adresi ya da Python Social Auth ile yapılan iç kimlik doğrulamasını etkiler (*REGISTRATION_ALLOW_BACKENDS* komutunu kullanarak belirli arka uçları beyaz listeye ekleyebilirsiniz).

Not: *LDAP kimlik doğrulaması* gibi üçüncü taraf kimlik doğrulama yöntemleri kullanılıyorsa bu seçenek yalnızca kayıt formunu gizler. Ancak yeni kullanıcılar yine de oturum ve hesap açabilir.

Ayrıca bakınız:

REGISTRATION_ALLOW_BACKENDS, REGISTRATION_EMAIL_MATCH, Kimlik doğrulama

2.16.80 REGISTRATION_REBIND

4.16 sürümünde geldi.

Var olan kullanıcılar için kimlik doğrulama hizmeti arka uçlarının yeniden bağlanmasını sağlar. Kimlik doğrulama hizmeti sağlayıcıları arasında geçiş yaparken bu seçeneği etkinleştirin.

Not: Bu seçenek varsayılan olarak, var olan hesaba başka kimlik doğrulama hizmeti arka uçlarının eklenmesini engellemek için devre dışı bırakılmıştır. Yeniden bağlama seçeneği, daha fazla üçüncü taraf kimlik doğrulama hizmeti arka ucunun kullanılmasını sağlarken hesabın ele geçirilme riskini artırır.

2.16.81 REPOSITORY_ALERT_THRESHOLD

4.0.2 sürümünde geldi.

Eski ya da çok fazla değişiklik içeren depolar için bir uyarı tetikleme eşiği. Varsayılan değer: 25.

Ayrıca bakınız:

alerts

2.16.82 REQUIRE_LOGIN

4.1 sürümünde geldi.

Bu seçenek, `LOGIN_REQUIRED_URLS` özelliğini etkinleştirir ve REST çatısını tüm API uç noktaları için kimlik doğrulaması gerekecek şekilde yapılandırır.

Not: Bu seçenek, *Örnek yapılandırma* içine eklenmiştir. Docker için `WEBLATE_REQUIRE_LOGIN` kullanın.

2.16.83 SENTRY_DSN

3.9 sürümünde geldi.

Hata raporlarını derlemek için kullanılacak Sentry DSN.

Ayrıca bakınız:

Sentry için Django bütünleştirilmesi

2.16.84 SESSION_COOKIE_AGE_AUTHENTICATED

4.3 sürümünde geldi.

Kimliği doğrulanmış kullanıcılar için oturum geçerlilik süresini ayarlayın. Bu seçenek, kimliği doğrulanmamış kullanıcılar için kullanılan `SESSION_COOKIE_AGE` seçeneğini tamamlar.

Ayrıca bakınız:

`SESSION_COOKIE_AGE`

2.16.85 SIMPLIFY_LANGUAGES

Varsayılan dil/tülke kombinasyonları için basit dil kodlarının kullanılmasını sağlar. Örneğin, bir “fr_FR” çevirisi “fr” dil kodunu kullanır. Varsayılan kombinasyonlar için dillerin listelenmesini basitleştirdiğinden genellikle bu davranış istenir.

Her çeşit için farklı çeviriler yapmak istiyorsanız bu seçeneği kapatın.

2.16.86 SITE_DOMAIN

Site etki alanını yapılandırır. Bu seçenek, birçok kapsamda doğru mutlak bağlantılar üretmek için gereklidir (etkinleştirme e-postaları, bildirimler ya da RSS akışlarındaki gibi).

Weblate standart olmayan bir bağlantı noktası kullanıyorsa, buraya da ekleyin.

Örnekler:

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

Not: Bu ayarda yalnızca etki alanı adı bulunmalıdır. İletişim kuralını yapılandırmak için (HTTPS kullanımını etkinleştirmek ve dayatmak) `ENABLE_HTTPS` seçeneğini ve adresi değiştirmek için `URL_PREFIX` seçeneğini kullanın.

İpucu: Bir Docker kapsayıcısında, site etki alanı `WEBLATE_ALLOWED_HOSTS` seçeneği ile yapılandırılır.

Ayrıca bakınız:

Doğru site etki alanını ayarlayın, Allowed hosts kurulumu, HTTPS ayarını düzgün biçimde yapın `WEBLATE_SITE_DOMAIN`, `ENABLE_HTTPS`

2.16.87 SITE_TITLE

Web sitesi ve gönderilecek e-postalar için kullanılacak site başlığı.

2.16.88 SPECIAL_CHARS

Görsel klavyeye eklenecek karakterler. *Sanal klavye*.

Varsayılan değer:

```
SPECIAL_CHARS = ("\t", "\n", "\u00a0", "...")
```

2.16.89 SINGLE_PROJECT

3.8 sürümünde geldi.

Kullanıcıları pano yerine doğrudan bir proje ya da bileşene yönlendirir. `True` olarak ayarlandığında yalnızca Weblate üzerinde tek bir proje olması durumunda çalışır. Alternatif olarak, proje kısaltmasını ayarlarsanız koşulsuz olarak bu projeye yönlendirir.

3.11 sürümünde değişti: Artık tek bir projenin görüntülenmesini dayatmak için bir proje kısaltması da ayarlanabiliyor.

Örnek:

```
SINGLE_PROJECT = "test"
```

2.16.90 SSH_EXTRA_ARGS

4.9 sürümünde geldi.

Weblate SSH çağırırken özel parametreler eklenmesine izin verir. Bu seçenek, eski şifrelemeyi veya diğer standart dışı özellikleri kullanan sunucularla bağlantı kurarken yardımcı olur.

Örneğin, Weblate SSH bağlantısı *Legacyhost ile iletişim kurulamıyor: Uygun bir anahtar değişimi yöntemi bulunamadı.* Önerilen: *diffie-hellman-group1-sha1* hatası ile kurulamadığında şunu kullanarak etkinleştirebilirsiniz:

```
SSH_EXTRA_ARGS = "-oKexAlgorithms =+diffie-hellman-group1-sha1"
```

İpucu: Dizge kabuk tarafından değerlendirilir. Bu nedenle boşluk ve özel karakterleri tırnak arasına aldığınızdan emin olun.

Ayrıca bakınız:

[Eski OpenSSH seçenekleri](#)

2.16.91 STATUS_URL

Weblate kopyanızın durumunu bildireceği adres.

2.16.92 SUGGESTION_CLEANUP_DAYS

3.2.1 sürümünde geldi.

Belirtilen sayıda gün sonra önerileri kendiliğinden siler. Varsayılan değer: Yok herhangi bir silme işlemi yapmaz.

2.16.93 UNUSED_ALERT_DAYS

4.17 sürümünde geldi.

Configures when *Component seems unused* alert is triggered.

Varsayılan değer 365 gündür. Devre dışı bırakmak için 0 olarak ayarlayın.

2.16.94 UPDATE_LANGUAGES

4.3.2 sürümünde geldi.

Veri tabanı aktarımı çalıştırılırken dil veri tabanının güncellenmesinin gerekip gerekmediğini denetler ve varsayılan olarak etkindir. Bu ayarın :d jadmin:setuplang ögesinin çağrılması üzerinde herhangi bir etkisi yoktur.

Uyarı: Bu seçenek diller görünümünü tutarsız kılabilir. Weblate dil tanımları zaman içinde zenginleşir ve tanımlanan dillerin dil kodu görüntülenmez.

Ayrıca bakınız:

[Hazır dil tanımları](#)

2.16.95 URL_PREFIX

Bu ayar, Weblate platformunun belirli bir yol altında çalışmasını sağlar (yoksa site sunucusunun kök klasöründen çalıştırılır).

Not: Bu ayarı kullanmak için, sunucunuzu bu ön eki kaldırarak şekilde yapılandırmanız gerekir. BU işlem örneğin, WSGI ile `WSGIScriptAlias` seçeneği ayarlanarak elde edilebilir.

İpucu: Ön ek / karakteri ile başlamalıdır.

Örnek:

```
URL_PREFIX = "/translations"
```

Not: Bu ayar iç Django sunucusuyla çalışmaz. `urls.py` dosyasını bu ön eki içerecek şekilde ayarlamanız gerekir.

2.16.96 VCS_API_DELAY

4.15.1 sürümünde geldi.

GitHub çekme istekleri, *GitLab birleştirme istekleri*, *Gitea çekme isteği* ve *Pagure birleştirme istekleri* içindeki üçüncü taraf API çağrılarında saniye olarak en az gecikmeyi yapılandırır.

Bu seçenek, aşırı yüklenmelerini önlemek için Weblate üzerinden bu hizmetlere yapılan API çağrılarının hızını sınırlar.

GitHub üzerinde ikincil hız sınırlayıcısına takılıyorsanız, bu değeri artırmak yardımcı olabilir.

Varsayılan değer: 10.

2.16.97 VCS_BACKENDS

Kullanılabilecek sürüm denetimi sistemi arka uçlarının yapılandırması.

Not: Weblate, araçlarının bulunduğu desteklenen tüm arka uçları kullanmaya çalışır.

İpucu: Bu ayar ile seçenekleri sınırlayabilir ya da özel sürüm denetimi sistemi arka uçları ekleyebilirsiniz.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository", )
```

Ayrıca bakınız:

Sürüm denetimi bütünleştirilmesi

2.16.98 VCS_CLONE_DEPTH

3.10.2 sürümünde geldi.

Weblate depolarının hangi alt düzeye kadar kopyalanacağını yapılandırır.

Not: Bu seçenek şu anda yalnızca *Git* için destekleniyor. Varsayılan olarak Weblate, kopyalamayı hızlandırmak ve disk alanını korumak için sığ depo kopyaları oluşturur. Kullanımınıza bağlı olarak (örneğin custom *Eklentiler* kullanırken), kopya derinliği artırmak ya da 0 olarak ayarlayarak tümüyle kapatmak isteyebilirsiniz.

İpucu: Weblate üzerinden iterken ``fatal: protocol error: expected old/new/ref, got 'shallow <commit hash>`` hatası alırsanız, sığ kopyaları tümüyle kapatarak şunları ayarlayın:

```
VCS_CLONE_DEPTH = 0
```

2.16.99 WEBLATE_ADDONS

Kullanılabilecek eklentilerin listesi. Bunların kullanılabilmesi için, belirli bir çeviri bileşeninde etkinleştirilmeleri gerekir. Varsayılan olarak bu, tüm iç eklentileri içerir. Listeyi genişletirken büyük olasılıkla var olanları etkin tutmak isteyeceksiniz. Örneğin:

```
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.LanguangeConsistencyAddon",  
    "weblate.addons.discovery.DiscoveryAddon",  
    "weblate.addons.flags.SourceEditAddon",  
    "weblate.addons.flags.TargetEditAddon",  
    "weblate.addons.flags.SameEditAddon",  
    "weblate.addons.flags.BulkEditAddon",  
    "weblate.addons.generate.GenerateFileAddon",  
    "weblate.addons.json.JSONCustomizeAddon",  
    "weblate.addons.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",  
)
```

Not: Eklentiği listeden kaldırmak, bileşenlerden kaldırmaz. Bu durumda Weblate çöker. Lütfen eklentiği bu listeden kaldırmadan önce tüm bileşenlerden kaldırın.

Ayrıca bakınız:*Eklentiler, DEFAULT_ADDONS***2.16.100 WEBLATE_EXPORTERS**

4.2 sürümünde geldi.

Çevirilerin veya sözlüklerin çeşitli dosya biçimlerinde indirilmesini sağlayan dış aktarıcıların listesi.

Ayrıca bakınız:*Desteklenen dosya biçimleri***2.16.101 WEBLATE_FORMATS**

3.0 sürümünde geldi.

Kullanılabilecek dosya biçimlerinin listesi.

Not: Varsayılan listede yaygın kullanılan biçimler bulunur.

Ayrıca bakınız:*Desteklenen dosya biçimleri***2.16.102 WEBLATE_MACHINERY**

4.13 sürümünde geldi.

Kullanılabilecek makine hizmetlerinin listesi.

Ayrıca bakınız:*Kendiliğinden önerileri yapılandırmak***2.16.103 WEBLATE_GPG_IDENTITY**

3.1 sürümünde geldi.

Weblate tarafından Git işlemlerini imzalamak için kullanılacak kimlik. Örneğin:

```
WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"
```

Weblate GPG anahtarlığında, uygun bir anahtar aranır (home/ .gnupg altında *DATA_DIR*). Bulunamazsa, bir anahtar oluşturulur. Bilgi almak için: *Git işlemlerini GnuPG ile imzalamak*.

Ayrıca bakınız:*Git işlemlerini GnuPG ile imzalamak*

2.16.104 WEBSITE_REQUIRED

Bir proje oluşturulurken *Proje sitesi* ögesinin belirtilmesinin gerekip gerekmediğini belirler. Varyılan olarak, herkese açık sunucu kurulumlarına uygun olarak açıktır.

2.17 Örnek yapılandırma

Bu örnek, Weblate ile birlikte gelen `weblate/settings_example.py` dosyasıdır:

```
# 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"

# 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",
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

        # 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
# 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"),
    ("cy", "Cymraeg"),
    ("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"),

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    ("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", "中文"),
    ("zh-hant", "中文"),
)

SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

# If you set this to False, Django will not format dates, numbers and
# calendars according to the current locale.
USE_L10N = True

# If you set this to False, Django will not use timezone-aware datetimes.
USE_TZ = True

# Type of automatic primary key, introduced in Django 3.2
DEFAULT_AUTO_FIELD = "django.db.models.AutoField"

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(CACHE_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 = (

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    "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",
            ],
        },
        "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"]

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

SOCIAL_AUTH_GITHUB_ORG_KEY = ""
SOCIAL_AUTH_GITHUB_ORG_SECRET = ""
SOCIAL_AUTH_GITHUB_ORG_NAME = ""

SOCIAL_AUTH_GITHUB_TEAM_KEY = ""
SOCIAL_AUTH_GITHUB_TEAM_SECRET = ""
SOCIAL_AUTH_GITHUB_TEAM_ID = ""

SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True

SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}

SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# Social auth settings
SOCIAL_AUTH_PIPELINE = (
    "social_core.pipeline.social_auth.social_details",
    "social_core.pipeline.social_auth.social_uid",
    "social_core.pipeline.social_auth.auth_allowed",
    "social_core.pipeline.social_auth.social_user",
    "weblate.accounts.pipeline.store_params",
    "weblate.accounts.pipeline.verify_open",
    "social_core.pipeline.user.get_username",
    "weblate.accounts.pipeline.require_email",
    "social_core.pipeline.mail.mail_validation",
    "weblate.accounts.pipeline.revoke_mail_code",
    "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

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

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"
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.CharsPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.PastPasswordsValidator"},
    # Optional password strength validation by django-zxcvbn-password
    # {
    #     "NAME": "zxcvbn_password.ZXCVBNValidator",
    #     "OPTIONS": {
    #         "min_score": 3,
    #         "user_attributes": ("username", "email", "full_name")
    #     }
    # },
]

# Password hashing (prefer Argon)
PASSWORD_HASHERS = [
    "django.contrib.auth.hashers.Argon2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher",
    "django.contrib.auth.hashers.BCryptSHA256PasswordHasher",
]

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [
    "weblate.middleware.RedirectMiddleware",
    "weblate.middleware.ProxyMiddleware",
    "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",

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    "social_django.middleware.SocialAuthExceptionMiddleware",
    "weblate.accounts.middleware.RequireLoginMiddleware",
    "weblate.api.middleware.ThrottlingMiddleware",
    "weblate.middleware.SecurityMiddleware",
    "weblate.wladmin.middleware.ManageMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.formats",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",
    "weblate.screenshots",
    "weblate.fonts",
    "weblate.accounts",
    "weblate.configuration",
    "weblate.utils",
    "weblate.vcs",
    "weblate.wladmin",
    "weblate.metrics",
    "weblate",
    # Optional: Git exporter
    "weblate.gitexport",
    # 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)

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# - 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
# 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,

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    },
    # Logging to a file
    # "logfile": {
    #     "level": "DEBUG",
    #     "class": "logging.handlers.RotatingFileHandler",
    #     "filename": "/var/log/weblate/weblate.log",
    #     "maxBytes": 100000,
    #     "backupCount": 3,
    #     "formatter": "logfile",
    # },
},
"loggers": {
    "django.request": {
        "handlers": ["mail_admins", DEFAULT_LOG],
        "level": "ERROR",
        "propagate": True,
    },
    "django.server": {
        "handlers": ["django.server"],
        "level": "INFO",
        "propagate": False,
    },
    # Logging database queries
    # "django.db.backends": {
    #     "handlers": [DEFAULT_LOG],
    #     "level": "DEBUG",
    # },
    "redis_lock": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    "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",

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

#      "weblate.machinery.saptranslationhub.SAPTranslationHub",
#      "weblate.machinery.youdao.YoudaoTranslation",
"weblate.machinery.weblatetm.WeblateTranslation",
"weblate.memory.machine.WeblateMemory",
)

# Machine translation API keys

# URL of the Apertium APY server
MT_APERTIUM_APY = None

# DeepL API key
MT_DEEPL_KEY = None

# LibreTranslate
MT_LIBRETRANSLATE_API_URL = None
MT_LIBRETRANSLATE_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

# ModernMT
MT_MODERNMT_KEY = None

# MyMemory identification email, see
# https://mymemory.translated.net/doc/spec.php
MT_MYMEMORY_EMAIL = None

# Optional MyMemory credentials to access private translation memory
MT_MYMEMORY_USER = None
MT_MYMEMORY_KEY = None

# 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

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

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
SECURE_SSL_HOST = SITE_DOMAIN
# Sent referrrrer only for same origin links
SECURE_REFERRER_POLICY = "same-origin"
# SSL redirect URL exemption list
SECURE_REDIRECT_EXEMPT = (r"healthz/$",) # Allowing HTTP access to health check
# Session cookie age (in seconds)
SESSION_COOKIE_AGE = 1000
SESSION_COOKIE_AGE_AUTHENTICATED = 1209600
SESSION_COOKIE_SAMESITE = "Lax"
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 50000000
# Allow more fields for case with a lot of subscriptions in profile
DATA_UPLOAD_MAX_NUMBER_FIELDS = 2000

# Apply session cookie 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/"

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow longer
# translations (up to 10.000 characters)
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True

# Use simple language codes for default language/country combinations
SIMPLIFY_LANGUAGES = True

# Render forms using bootstrap
CRISPY_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",

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# "weblate.checks.format.JavaMessageFormatCheck",
# "weblate.checks.format.PercentPlaceholdersCheck",
# "weblate.checks.format.VueFormattingCheck",
# "weblate.checks.format.I18NextInterpolationCheck",
# "weblate.checks.format.ESTemplateLiteralsCheck",
# "weblate.checks.angularjs.AngularJSInterpolationCheck",
# "weblate.checks.icu.ICUMessageFormatCheck",
# "weblate.checks.icu.ICUSourceCheck",
# "weblate.checks.qt.QtFormatCheck",
# "weblate.checks.qt.QtPluralCheck",
# "weblate.checks.ruby.RubyFormatCheck",
# "weblate.checks.consistency.PluralsCheck",
# "weblate.checks.consistency.SamePluralsCheck",
# "weblate.checks.consistency.ConsistencyCheck",
# "weblate.checks.consistency.TranslatedCheck",
# "weblate.checks.chars.EscapedNewlineCountingCheck",
# "weblate.checks.chars.NewLineCountCheck",
# "weblate.checks.markup.BBCodeCheck",
# "weblate.checks.chars.ZeroWidthSpaceCheck",
# "weblate.checks.render.MaxSizeCheck",
# "weblate.checks.markup.XMLValidityCheck",
# "weblate.checks.markup.XMLTagsCheck",
# "weblate.checks.markup.MarkdownRefLinkCheck",
# "weblate.checks.markup.MarkdownLinkCheck",
# "weblate.checks.markup.MarkdownSyntaxCheck",
# "weblate.checks.markup.URLCheck",
# "weblate.checks.markup.SafeHTMLCheck",
# "weblate.checks.placeholders.PlaceholderCheck",
# "weblate.checks.placeholders.RegexCheck",
# "weblate.checks.duplicate.DuplicateCheck",
# "weblate.checks.source.OptionalPluralCheck",
# "weblate.checks.source.EllipsisCheck",
# "weblate.checks.source.MultipleFailingCheck",
# "weblate.checks.source.LongUntranslatedCheck",
# "weblate.checks.format.MultipleUnnamedFormatsCheck",
# "weblate.checks.glossary.GlossaryCheck",
# )

# 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.LangaueConsistencyAddon",
#     "weblate.addons.discovery.DiscoveryAddon",
#     "weblate.addons.autotranslate.AutoTranslateAddon",
#     "weblate.addons.flags.SourceEditAddon",
#     "weblate.addons.flags.TargetEditAddon",

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# "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"

# 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(CACHE_DIR, "avatar"),
        "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.

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

"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
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"
]

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# 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"},
}

# 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 Yönetim komutları

Not: Yönetim komutlarını site sunucunuzu çalıştıran kullanıcıdan başka bir kullanıcı ile çalıştırmak, dosya izinlerinin yanlış olmasına neden olabilir. Bilgi almak için: [Dosya sistemi izinleri](#).

Temel yönetim komutlarını (Django kaynaklarında `./manage.py` dosyasında ya da genişletilmiş bir küme olarak Weblate üzerine kurulabilen **weblate** adlı betik dosyasında bulabilirsiniz).

2.18.1 Yönetim komutlarını çağırmak

Daha önce de belirtildiği gibi, çağrı yöntemi Weblate kurulumunuza bağlıdır.

Weblate için virtualenv kullanıyorsanız, **weblate** komutunun tam yolunu belirtebilir ya da çağırmadan önce virtualenv etkinleştirebilirsiniz:

```
# Direct invocation
~/weblate-env/bin/weblate

# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate
```

Kaynak kodunu doğrudan kullanıyorsanız (tarball ya da Git checkout menüsünden), Weblate kaynaklarındaki yönetim betiği `./manage.py` dosyasıdır. Çalıştırmak için:

```
python ./manage.py list_versions
```

Weblate kurulumu için pip kurucusunu ya da `./setup.py` betiğini kullandıysanız, Weblate denetimi için kullanılacak **weblate** yolunuza (ya da kullandığınız virtualenv yoluna) kurulum:

```
weblate list_versions
```

Docker kalıbında betik yukarıdaki gibi kurulur ve **docker exec** komutuyla çalıştırılabilir:

```
docker exec --user weblate <container> weblate list_versions
```

docker-compose için de işlem benzerdir. Yalnızca **docker-compose exec** kullanmanız gerekir:

```
docker-compose exec --user weblate weblate weblate list_versions
```

Bir dosya belirtilmezse, geçici olarak bir birim ekleyebilirsiniz:

```
docker-compose exec --user weblate /tmp:/tmp weblate weblate importusers /tmp/
↪users.json
```

Ayrıca bakınız:

Docker ile kurmak, Debian ve Ubuntu üzerine kurmak, SUSE ve openSUSE üzerine kurmak, RedHat, Fedora ve CentOS üzerine kurmak, Kaynaklardan kurulum

2.18.2 add_suggestions

weblate add_suggestions <project> <component> <language> <file>

Dosyadaki çeviriyi belirtilen çeviri için öneri olarak içe aktarır. Aynı olan çevirileri atlar. Yalnızca farklı olanları ekler.

--author USER@EXAMPLE.COM

Önerileriniz için sorumlu e-posta adresi. Bu kullanıcının içe aktarma işleminden önce var olması gerekir (gerekirse yönetici arayüzünden bir kullanıcı oluşturabilirsiniz).

Örnek:

```
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>

4.6 sürümünde değişti: Çeviri kipi parametresi eklenir.

Diğer bileşen çevirilerini temel alarak kendiliğinden çeviri yapar.

--source PROJECT/COMPONENT

Çeviri için kaynak olarak kullanılacak bileşeni belirtir. Belirtilmediğinde, projedeki tüm bileşenler kullanılır.

--user USERNAME

Çevirilerin yazarı olarak görüntülenecek kullanıcı adını belirtir. Belirtilmediğinde “Anonim” olarak görüntülenir.

--overwrite

Var olan çevirilerin üzerine yazılıp yazılmayacağını belirler.

--inconsistent

Tutarsız olan var olan çevirilerin üzerine yazılıp yazılmayacağını belirler (bilgi almak için *Tutarsız*).

--add

Belirtilen bir çeviri bulunmuyorsa dili kendiliğinden ekler.

--mt MT

Makine çevirisi olarak diğer bileşenler yerine makine çevirisini kullanır.

--threshold THRESHOLD

Makine çevirisi için benzerlik eşiği, varsayılan değer: 80.

--mode MODE

Çeviri kipini belirtir. Varsayılan değer: translate. fuzzy ya da suggest seçenekleri de kullanılabilir.

Örnek:

```
weblate auto_translate --user nijel --inconsistent --source weblate/application_
↪weblate website cs
```

Ayrıca bakınız:

Kendiliğinden çeviri

2.18.4 celery_queues

weblate celery_queues

3.7 sürümünde geldi.

Celery görev kuyruklarının uzunluğunu görüntüler.

2.18.5 checkgit

weblate checkgit <project|project/component>

Arka uç Git deposunun geçerli durumunu yazdırır.

Hangi projenin veya bileşenin güncelleneceğini tanımlayabilirsiniz (weblate/application gibi) ya da var olan tüm bileşenleri güncellemek için --all seçeneğini kullanabilirsiniz.

2.18.6 commitgit

weblate commitgit <project|project/component>

Arka uç Git deposunda bekliyor olabilecek değişiklikleri işler.

Hangi projenin veya bileşenin güncelleneceğini tanımlayabilir (`weblate/application` gibi), var olan tüm bileşenleri güncellemek için `--all` seçeneğini kullanabilir ya da dosya biçimine göre süzmek için `--file-format` seçeneğini kullanabilirsiniz.

2.18.7 commit_pending

weblate commit_pending <project|project/component>

Belirli bir süreden daha önce bekleyen değişiklikleri işler.

Hangi projenin veya bileşenin güncelleneceğini tanımlayabilirsiniz (`weblate/application` gibi) ya da var olan tüm bileşenleri güncellemek için `--all` seçeneğini kullanabilirsiniz.

--age HOURS

İşleme için saat olarak yaş. Belirtilmezse, *Bileşen yapılandırması* içinde yapılandırılmış değer kullanılır.

Not: Bu işlem, Weblate tarafından arka planda kendiliğinden yapılır. Bu nedenle *Bileşen yapılandırması* tarafından belirtilenden daha erken bir işlemeyi dayatmanın yanında, bunu el ile çağırmaya gerek yoktur.

Ayrıca bakınız:

Bakım görevlerini yürütmek, `COMMIT_PENDING_HOURS`

2.18.8 cleanuptrans

weblate cleanuptrans

Arta kalmış denetimleri ve çeviri önerilerini temizler. Normalde bunu el ile çalıştırmaya gerek yoktur. Temizlik arka planda kendiliğinden yapılır.

Ayrıca bakınız:

Bakım görevlerini yürütmek

2.18.9 cleanup_ssh_keys

weblate cleanup_ssh_keys

4.9.1 sürümünde geldi.

Depolanmış SSH sunucu anahtarlarını temizler:

- GitHub tarafından kullanımdan kaldırılmış RSA anahtarlarını GitHub ile bağlantı kurma sorunlarına yol açmaması için kaldırır.
- Çift sunucu anahtarı kayıtlarını kaldırır.

Ayrıca bakınız:

SSH depoları

2.18.10 createadmin

weblate createadmin

Parola belirtilmemişse rastgele bir parolay ile bir admin hesabı oluşturur.

--password PASSWORD

Rastgele bir parola oluşturulmaması için komut satırından bir parola belirtilmesini sağlar.

--no-password

Parola ayarlanmaz. Bu kullanım *--update* seçeneği ile yararlı olabilir.

--username USERNAME

Admin yerine belirtilen ad kullanılır.

--email USER@EXAMPLE.COM

Yönetici e-posta adresini belirtir.

--name

Yönetici adını belirtir (gözümlenecek).

--update

Var olan kullanıcıyı günceller (parolaları değıştirmek için bunu kullanabilirsiniz).

2.18.11 dump_memory

weblate dump_memory

Weblate çeviri belleği içeriğini bir JSON dosyası olarak dışa aktarır.

Ayrıca bakınız:

Çeviri belleği, Weblate çeviri belleği şeması

2.18.12 dumpuserdata

weblate dumpuserdata <file.json>

Kullanıcı verilerini daha sonra *importuserdata* tarafından kullanılabilir biçimde bir dosyaya kaydeder.

İpucu: Bu seçenek, Weblate kopyalarını taşıırken ya da birleştiren işe yarar.

2.18.13 import_demo

weblate import_demo

4.1 sürümünde geldi.

[<https://github.com/WeblateOrg/demo>](https://github.com/WeblateOrg/demo) bileşenlerini temel alan bir tanıtım projesi oluşturur. Bu komutu çalıştırmadan önce Celery görevlerinin çalıştığından emin olun.

Bu seçenek, Weblate geliştirme çalıştırmaları için yararlı olabilir.

2.18.14 import_json

weblate import_json <json-file>

JSON verileri kullanan bileşenleri toplu olarak içe aktarır.

İçe aktarılan JSON dosya yapısı aşağı yukarı bileşen nesnesine karşılık gelir (bilgi almak için: [GET /api/components/\(string:project\)/\(string:component\)/](#)). name ve filemask alanlarını katmanız gerekir.

--project PROJECT

Bileşenlerin nereden içe aktarılacağını belirtir.

--main-component COMPONENT

Bunların tümü için bu bileşenden verilen sürüm denetimi sistemi deposunu kullanır.

--ignore

İçe aktarılmış (zaten) bileşenleri atlar.

--update

İçe aktarılmış (zaten) bileşenleri günceller.

JSON dosyası örneği:

```
[
  {
    "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"
  }
]
```

Ayrıca bakınız:

[import_memory](#)

2.18.15 import_memory

weblate import_memory <file>

TMX ya da JSON dosyasını Weblate çeviri belleği içine aktarır.

--language-map LANGMAP

TMX dosyasındaki dillerin Weblate çeviri belleği ile eşlenmesini sağlar. Dil kodları genellikle Weblate tarafından yapılan normalleştirmeden sonra eşlenir.

Örneğin, `--language-map en_US:en`, tüm en_US dizgelerini en dizgeleri olarak içe aktarır.

Bu seçenek, TMX dosyasının yerel ayarlarının Weblate üzerindekiyle eşleşmemesi durumunda yararlı olabilir.

Ayrıca bakınız:

[Çeviri belleği](#), [Weblate çeviri belleği şeması](#)

2.18.16 import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

3.0 sürümünde değişti: import_project komutu artık *Bileşen keşfi* eklentisini temel aldığından davranışında ve kabul ettiği parametrelerde bazı değişiklikler vardır.

Toplu işlem, bileşenleri dosya maskesine göre projeye aktarır.

<project>, bileşenlerin içe aktarılacağı var olan bir projeyi adlandırır.

<gitrepo>, kullanılacak Git deposu adresini tanımlar ve <branch> Git dalını belirtir. Var olan bir Weblate bileşeninden ek çeviri bileşenlerini içe aktarmak için, <gitrepo> için bir `weblate://<project>/<component>` adresi kullanın.

<filemask>, depoda aranacak dosyaları tanımlar. Genel arama karakterleri kullanılarak basitleştirilebilir ya da kurallı ifadelerin tüm gücünü kullanabilir.

Basit eşleştirmede, bileşen adı için `**` ve dil için `*` kullanılır. Örnek: `**/*.po`

Kurallı ifadede *component* ve *language* adlı gruplar bulunmalıdır. Örneğin: `(?P<language>[^\s]*)/(?P<component>[^\s]*)\.po`

İçe aktarma, dosyalara bakarak var olan bileşenleri bulur ve var olmayanları ekler. Zaten var olanları değiştirmez.

--name-template TEMPLATE

Django kalıp söz dizimini kullanarak bir bileşenin adını özelleştirin.

Örnek: `Belgeler: {{ component }}`

--base-file-template TEMPLATE

Tek dilli çeviriler için temel dosyayı özelleştirin.

Örnek: `{{ component }}/res/values/string.xml`

--new-base-template TEMPLATE

Yeni çevirilerin eklenmesi için temel dosyayı özelleştirin.

Örnek: `{{ component }}/ts/en.ts`

--file-format FORMAT

Kullanılacak dosya biçimini de belirtebilirsiniz (bilgi almak için: *Desteklenen dosya biçimleri*), varsayılan değer: otomatik algılama.

--language-regex REGEX

Bu parametreyle süzülecek dili belirtebilirsiniz (bilgi almak için: *Bileşen yapılandırması*). Kurallı ifade geçerli olmalıdır.

--main-component

Hangi bileşenin ana bileşen olarak seçileceğini belirtebilirsiniz. VCS deposunu içeren bileşen olmalıdır.

--license NAME

Genel, proje veya bileşen çeviri lisansını belirtin.

--license-url URL

Çeviri lisansının bulunduğu adresi belirtin.

--vcs NAME

Hangi sürüm denetim sistemini kullanacağınızı belirtmeniz gerekirse, buradan yapabilirsiniz. Varsayılan sürüm denetimi: Git.

Örnek olması için, iki projeyi içe aktarmayı deneyelim.

İlk olarak, her dilin her bölümün çevirilerinin ayrı bir klasörde bulunan Debian el kitabı çevirileri:

```
weblate import_project \
  debian-handbook \
  git://anonscm.debian.org/debian-handbook/debian-handbook.git \
  squeeze/master \
  '*/**.po'
```

Ardından, temel dosya kalıbıyla birlikte dosya biçiminin belirtilmesi gereken Tanaguru aracı ve tüm bileşenler ile çevirilerin tek bir klasörde nasıl bulunduğu:

```
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
```

Daha karmaşık bir örnek olarak, `src/security/Numerous_security_holes_in_0.10.1.de.po` gibi bir dosya adından doğru bileşeni ve dili ayıklamak:

```
weblate import_project \
  tails \
  git://git.tails.boum.org/tails master \
  'wiki/src/security/(?P<component>.*)\.(?P<language>[^\.]*)\.po$'
```

Yalnızca seçilmiş bir dildeki çevirileri süzme:

```
./manage import_project \
  --language-regex '^(cs|sk)$' \
  weblate \
  https://github.com/WeblateOrg/weblate.git \
  'weblate/locale/*/LC_MESSAGES/**/*.po'
```

Birden çok dosyaya ayrılmış Sphinx belgelerini içe aktarmak:

```
$ weblate import_project --name-template 'Documentation: %s' \
  --file-format po \
  project https://github.com/project/docs.git master \
  'docs/locale/*/LC_MESSAGES/**/*.po'
```

Birden çok dosya ve klasöre ayrılmış Sphinx belgelerini içe aktarmak:

```
$ 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'
```

Ayrıca bakınız:

Daha ayrıntılı örnekler `starting` bölümünde bulunabilir. Alternatif olarak `import_json` kullanmak isteyebilirsiniz.

2.18.17 importuserdata

weblate importuserdata <file.json>

dumpuserdata ile oluşturulmuş bir dosyadan kullanıcı verilerini içe aktarır.

2.18.18 importusers

weblate importusers --check <file.json>

Django auth_users veri tabanının JSON dökümünden kullanıcıları içe aktarır.

--check

Bu seçenekle, yalnızca belirli bir dosyanın içe aktarılıp aktarılamayacağı denetlenir ve kullanıcı adlarından ya da e-posta adreslerinden kaynaklanan olası çakışmalar bildirilir.

Şunu kullanarak kullanıcıları var olan Django kurulumundan dönebilirsiniz:

```
weblate dumpdata auth.User > users.json
```

2.18.19 install_addon

3.2 sürümünde geldi.

weblate install_addon --addon ADDON <project|project/component>

Bir bileşen kümesine bir eklenti kurar.

--addon ADDON

Kurulacak eklentinin adı. Örneğin `weblate.gettext.customize`.

--configuration CONFIG

Bir eklentinin JSON olarak kodlanmış yapılandırması.

--update

Var olan eklenti yapılandırmasını güncelle.

Eklentinin hangi projeye ya da bileşene kurulacağını belirtebilir (`weblate/application` gibi) ya da var olan tüm bileşenleri katmak için `--all` kullanabilirsiniz.

Tüm bileşenler için *Gettext çıktısını özelleştirme* kurmak için:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --
↪update --all
```

Ayrıca bakınız:

Eklentiler

2.18.20 list_languages

weblate list_languages <locale>

MediaWiki kodlarında desteklenen dilleri listeler. Dil kodları, İngilizce adlar ve yerelleştirilmiş adlar.

Bu, https://wiki.110n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF oluşturmak için kullanılır.

2.18.21 list_translators

weblate list_translators <project|project/component>

Belirtilen projeye katkıda bulunan çevirmenleri dile göre listeler:

```
[French]
Jean Dupont <jean.dupont@example.com>
[English]
John Doe <jd@example.com>
```

--language-code

Adları dil adı yerine dil koduna göre listeler.

Hangi projenin ya da bileşenin kullanılacağını belirtebilirsiniz (weblate/application gibi) ya da var olan tüm bileşenlerin çevirmenlerini listelemek için **--all** seçeneğini kullanabilirsiniz.

2.18.22 list_versions

weblate list_versions

Tüm Weblate bağımlılıklarını ve sürümlerini listeler.

2.18.23 loadpo

weblate loadpo <project|project/component>

Çevirileri diskten yeniden yükler (örneğin, sürüm denetimi sistemi deposunda bazı güncellemeler yaptıysanız).

--force

Dosyalar güncel olsa bile güncellemeye dayatın.

--lang LANGUAGE

İşlemi tek bir dille sınırlandırın.

Hangi projenin veya bileşenin güncelleneceğini tanımlayabilirsiniz (weblate/application gibi) ya da var olan tüm bileşenleri güncellemek için **--all** seçeneğini kullanabilirsiniz.

Not: Bunu nadiren çağırmanız gerekir, Weblate her sürüm denetimi sistemi güncellemesi için değiştirilen dosyaları otomatik olarak yükler. Bu işlem, temel alınan bir Weblate sürüm denetimi sistemi deposunu el ile değiştirmeniz durumunda ya da yükseltmenin ardından bazı özel durumlarda gerekir.

2.18.24 lock_translation

weblate lock_translation <project|project/component>

Bir bileşen üzerinde çeviri yapılmasını önler.

İpucu: Temel alınan depoda bazı bakım işlemleri yapmak istediğinizde yararlıdır.

Hangi projenin veya bileşenin güncelleneceğini tanımlayabilirsiniz (weblate/application gibi) ya da var olan tüm bileşenleri güncellemek için **--all** seçeneğini kullanabilirsiniz.

Ayrıca bakınız:

[unlock_translation](#)

2.18.25 move_language

weblate move_language source target

3.0 sürümünde geldi.

Dil içeriğini birleştirmenizi sağlar. Bu seçenek, (*generated*) son ekiyle oluşturulmuş daha önce bilinmeyen dillerin diğer adlarını içeren yeni bir sürüme güncellerken yararlıdır. Tüm içeriği *kaynak* dilden *hedef* dile taşır.

Örnek:

```
weblate move_language cze cs
```

İçeriği taşıdıktan sonra, geride bir şey kalıp kalmadığını denetleyip (bu arada birisi depoyu güncellerse yarış koşulları geçerli olur) ve (*generated*) dili kaldırmalısınız.

2.18.26 pushgit

weblate pushgit <project|project/component>

İşlenen değişiklikleri yukarı akış sürüm denetimi deposuna iter.

--force-commit

Bekleyen değişikliklerin itilmeden önce işlenmesini dayatır.

Hangi projenin veya bileşenin güncelleneceğini tanımlayabilirsiniz (*weblate/application* gibi) ya da var olan tüm bileşenleri güncellemek için *--all* seçeneğini kullanabilirsiniz.

Not: Weblate *Bileşen yapılandırması* içindeki *İşleme ile itme* seçeneği etkinleştirilmiş ise değişiklikler otomatik olarak gönderilir ve bu varsayılan değerdir.

2.18.27 unlock_translation

weblate unlock_translation <project|project/component>

Belirli bir bileşenin kilidini açarak çeviri yapılmasına izin verir.

İpucu: Temel alınan depoda bazı bakım işlemleri yapmak istediğinizde yararlıdır.

Hangi projenin veya bileşenin güncelleneceğini tanımlayabilirsiniz (*weblate/application* gibi) ya da var olan tüm bileşenleri güncellemek için *--all* seçeneğini kullanabilirsiniz.

Ayrıca bakınız:

lock_translation

2.18.28 setupgroups

weblate setupgroups

Varsayılan grupları yapılandırır ve isteğe bağlı olarak tüm kullanıcıları bu varsayılan gruba atar.

--no-privs-update

Var olan grupların otomatik olarak güncellenmesini devre dışı bırakır (yalnızca yeniler eklenir).

--no-projects-update

Var olan projeler için grupların otomatik olarak güncellenmesini engeller. Bu seçenek, var olan projelere yeni grupların eklenmesine izin verir. Bilgi almak için: *Proje erişim denetimi*.

Ayrıca bakınız:

Yetkiler ve hazır rollerin listesi

2.18.29 setuplang

weblate setuplang

Weblate üzerinde tanımlanmış dillerin listesini günceller.

--no-update

Var olan dillerin otomatik olarak güncellenmesini devre dışı bırakır (yalnızca yeniler eklenir).

2.18.30 updatechecks

weblate updatechecks <project|project/component>

Tüm dizgeler için tüm denetimleri günceller.

İpucu: Denetimlerde büyük değişiklikler yapılmış olan yükseltmelerde yararlıdır.

Hangi projenin veya bileşenin güncelleneceğini tanımlayabilirsiniz (`weblate/application` gibi) ya da var olan tüm bileşenleri güncellemek için `--all` seçeneğini kullanabilirsiniz.

2.18.31 updategit

weblate updategit <project|project/component>

Uzak sürüm denetimi sistemi depolarını getirir ve iç ön belleği günceller.

Hangi projenin veya bileşenin güncelleneceğini tanımlayabilirsiniz (`weblate/application` gibi) ya da var olan tüm bileşenleri güncellemek için `--all` seçeneğini kullanabilirsiniz.

Not: Genellikle depodaki kancaları, *updategit* ile düzenli olarak yoklamak yerine, *Bildirim kancaları* ögesini tetikleyecek şekilde yapılandırmak daha iyidir.

2.19 Duyurular

4.0 sürümünde değişti: Önceki sürümlerde bu özelliğe beyaz tahta iletileri adı veriliyordu.







Site genelinde, projeye, bileşene ya da dile özel duyurular yayınlayarak çevirmenlerinize bilgi sağlayın.


Amacı, bitiş tarihlerini ve durumu duyurun ya da çeviri hedeflerini belirtin.

Kullanıcılar, izledikleri projelerin duyuruları ile ilgili bir bildirim alır (devre dışı bırakmadıkları sürece).

Bu özellik, sitenin amacını duyurmaktan çeviriler için hedefler belirlemeye kadar çeşitli konularda yararlı olabilir.

Duyurular, *Yönetim* menüsündeki *Duyuru yayınla* seçeneği ile istenilen düzeyde yayınlanabilir:

 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 4.17
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

Yönetim arayüzü kullanılarak da eklenebilir:

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](#)

Duyurular daha sonra belirtilen bağlamlarına göre görüntülenir:

Herhangi bir bağlam belirtilmediğinde

Panoda görüntülenir (giriş sayfası).

Proje belirtildiğinde

Projenin tüm bileşenleri ve çevirilerinde görüntülenir.

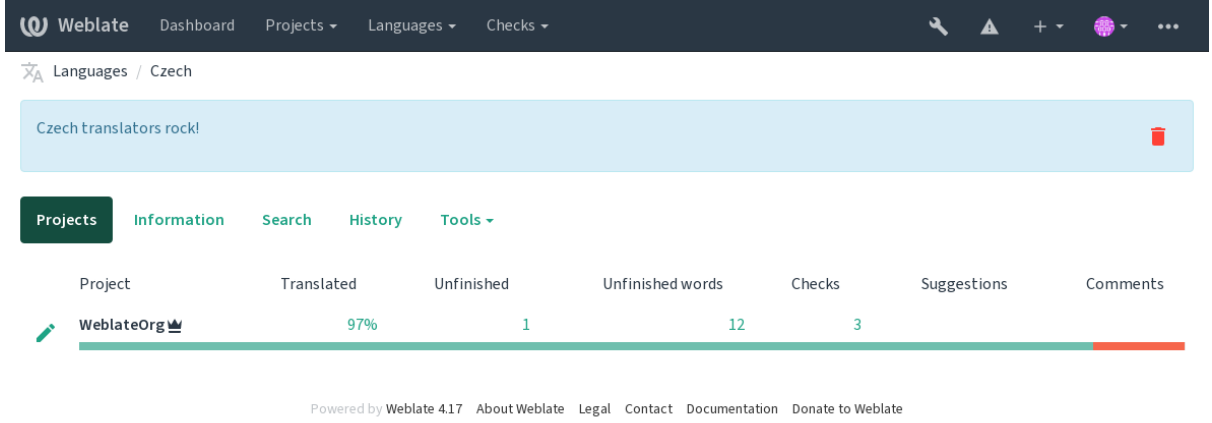
Bileşen belirtildiğinde

Belirtilen bileşen ve tüm çevirilerinde görüntülenir.

Dil belirtildiğinde

Dil özeti ve bu dildeki tüm çevirilerde görüntülenir.

Dil özet sayfası şöyle görüntülenir:



2.20 Bileşen listeleri

Kullanıcı panosunda seçenек olarak görüntülenecek ve kullanıcıların varsayılan görünümüleri olarak birini seçebileceği birkaç bileşen listesi belirtin. Bilgi almak için [Pano](#).

İpucu: Panoda sunulan her bileşen listesi için bir durum görüntülenir.

Bileşen listelerinin adları ve içeriği, yönetim arayüzünde, *Bileşen listeleri* bölümünde belirtilebilir. Her bileşen listesinin kullanıcıya görüntülenen bir adı ve adresi ile onu temsil eden bir kısaltması olmalıdır.

İpucu: Yönetim bölümünden anonim kullanıcılar için pano ayarlarını değiştirerek kimliği doğrulanmamış kullanıcılara hangi panonun sunulacağını belirtin.

2.20.1 Kendiliğinden bileşen listeleri

Kendiliğinden bileşen listesi ataması kuralları oluşturarak bileşenleri kısaltmalarına göre kendiliğinden listeye ekleyebilirsiniz.

- Büyük kurulumlarda bileşen listelerini korumak için ya da Weblate kurulumunuzdaki tüm bileşenleri içeren tek bir bileşen listesi elde etmek istediğinizde kullanışlıdır.

İpucu: Weblate kurulumunuzdaki tüm bileşenlerin bulunduğu bir bileşen listesi oluşturun.

1. Define *Automatic component list assignment* with `^.*$` as regular expression in both the project and the component fields, as shown on this image:

Weblate administration
WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Home · Weblate translations · Component lists · Add Component list

Add Component list

Required fields are marked in bold.

Component list name: Display name

URL slug: Name used in URLs and filenames.

☒ Show on dashboard
When enabled this component list will be shown as a tab on the dashboard

Components:

Available components ⓘ

- WeblateOrg/Django
- WeblateOrg/Language names
- WeblateOrg/WebblateOrg

Chosen components ⓘ

Choose all ⓘ Remove all ⓘ

Hold down "Control", or "Command" on a Mac, to select more than one.

AUTOMATIC COMPONENT LIST ASSIGNMENTS

PROJECT REGULAR EXPRESSION ⓘ	COMPONENT REGULAR EXPRESSION ⓘ	DELETE? ⓘ
<input type="text" value="^.*\$"/>	<input type="text" value="^.*\$"/>	<input type="button" value="X"/>

[+ Add another Automatic component list assignment](#)

2.21 İsteğe bağlı Weblate modülleri

Kurulumunuz için isteğe bağlı çeşitli modüller vardır.

2.21.1 Git dışı aktarıcı

HTTP(S) kullanarak temel alınan Git deposuna salt okunur erişim sağlar.

Kurulum

1. `settings.py` dosyasındaki kurulu uygulamalara `weblate.gitexport` ekleyin:

```
INSTALLED_APPS += ("weblate.gitexport",)
```

2. Kurulumdan sonra veri tabanınızı aktararak var olan depoları dışı aktarın:

```
weblate migrate
```

Kullanım

Modül otomatik olarak Weblate bağlantısı kurar ve *Bileşen yapılandırması* içinde dışa aktarılan depo adresini ayarlar. Depolara Weblate adresinin /git/ bölümünden erişilebilir. örneğin `https://site.org/git/weblate/main/`.

Herkese açık projelerin depoları kimlik doğrulaması olmadan kopyalanabilir:

```
git clone 'https://example.org/git/weblate/main/'
```

Sınırlanmış erişimi olan depolara göz atmak için erişim (*Gizli erişim denetimi* ile ya da `REQUIRE_LOGIN` seçeneği etkinleştirilerek), <user-profile> kullanıcı profilinizden alabileceğiniz bir API kodu ile sağlanır:

```
git clone 'https://user:KEY@example.org/git/weblate/main/'
```

İpucu: Varsayılan olarak, üyeler veya *Kullanıcılar* grubu ve anonim kullanıcı, herkese açık projeler için depolara *Depo erişimi* ve *Uzman kullanıcı* rolleri ile erişebilir.

2.21.2 Faturalama

Bu seçenek, faturalama tarifelerini tanımlamak, faturaları ve kullanım sınırlarını izlemek için *Hosted Weblate* üzerinde kullanılır.

Kurulum

1. Add `weblate.billing` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.billing",)
```

2. İsteğe bağlı olarak modül için ek veri tabanı yapıları kurmak üzere veri tabanı aktarımını çalıştırın:

```
weblate migrate
```

Kullanım

Kurulundan sonra faturalamayı yönetici arayüzünden yönetebilirsiniz. Faturalamanın etkin olduğu kullanıcılara *Kullanıcı profili* içinde *Faturalama* sekmesi görüntülenir.

Faturalama modülü ayrıca proje yöneticilerinin süper kullanıcı olmadan yeni projeler ve bileşenler oluşturmasını sağlar (bilgi almak için: *Çeviri projelerini ve bileşenleri eklemek*). Bunun için şu koşullar yerine getirilmelidir:

- Faturalama yapılandırılmış sınırlar içindedir (aşırı kullanım, proje/bileşen oluşturulmasını engeller) ve ödenmiştir (fiyatı sıfır değilse)
- Kullanıcı, faturalama ile var olan projenin yöneticisidir ya da kullanıcı faturalamanın sahibidir (ikincisi, kullanıcıların yeni projeleri içe aktarabilmesi için yeni faturalama oluştururken gereklidir).

Proje oluşturulduktan sonra kullanıcı, daha fazla özelliğe erişebilmesi durumunda proje için hangi fatura ücretlendirmesinin uygulanacağını seçebilir.

2.21.3 Yasal

Bu seçenek, gerekli yasal belgeleri sağlamak için [Hosted Weblate](#) üzerinde kullanılır. Boş belgelerle gelir ve belgelerde aşağıdaki kalıpları doldurmanız beklenir:

legal/documents/tos.html

Hizmet koşulları belgesi

legal/documents/privacy.html

Gizlilik ilkesi belgesi

legal/documents/summary.html

Hizmet koşulları ve gizlilik ilkeleri metinlerinin özeti

Hizmet koşulları belgelerini değiştirirken, lütfen `LEGAL_TOS_DATE` ayarını yapın, böylece kullanıcılara güncellenmiş belgeleri kabul etmeleri dayatılır.

Not: Hosted Weblate hizmeti için yasal belgeler <https://github.com/WeblateOrg/wllegal/tree/main/wllegal/templates/legal/documents> Git deposunda bulunabilir.

Büyük olasılıkla bunları olduğu gibi kullanamazsınız. Ancak gereksinimlerinize uygun olarak ayarlamakta başlangıç noktası olarak yararlı olabilir.

Kurulum

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. İsteğe bağlı olarak modül için ek veri tabanı yapıları kurmak üzere veri tabanı aktarımını çalıştırın:

```
weblate migrate
```

3. `weblate/legal/templates/legal/` klasöründeki yasal belgeleri size uygun şekilde düzenleyin.

Kullanım

Kurulum ve düzenlemeden sonra, yasal belgeler Weblate arayüzünde görüntülenir.

2.21.4 Avatarlar

Avatarlar, varsayılan olarak hizmet verdikleri sitelerden bilgi sızıntılarını azaltmak için sunucu tarafına indirilir ve ön belleğe alınır. Bunun için yapılandırılmış e-posta adreslerinden avatarları almayı sağlama özelliği `ENABLE_AVATARS` seçeneği kullanılarak kapatılabilir.

Weblate şu anda şunları destekliyor:

- Gravatar
- Libravatar

Ayrıca bakınız:

Avatar ön belleği, `AVATAR_URL_PREFIX`, `ENABLE_AVATARS`

2.21.5 İstenmeyen gönderim koruması

Akismet hizmetiyle kullanıcıların istenmeyen gönderimler yapmasını engelleyebilirsiniz.

1. *Akismet* Python modülünü kurun (resmi Docker kalıbında zaten vardır).
2. *Akismet* API anahtarını edinin.
3. `WEBLATE_AKISMET_API_KEY` ayarına ya da Docker üzerinde `AKISMET_API_KEY` ayarına ekleyin.

Şu içerikler denetlenmesi için *Akismet* üzerine gönderilir:

- Kimliği doğrulanmamış kullanıcılardan öneriler
- Proje ve bileşen açıklamaları ve bağlantıları

Not: Bu (diğer şeylerin yanında) istemcinin IP adresini kullanır. Lütfen bu yapılandırmayı düzgün bir şekilde yapmak için *Ters vekil sunucu arkasında çalıştırmak* bölümüne bakın.

Ayrıca bakınız:

Ters vekil sunucu arkasında çalıştırmak, `AKISMET_API_KEY`, `WEBLATE_AKISMET_API_KEY`

2.21.6 Git işlemlerini GnuPG ile imzalamak


3.1 sürümünde geldi.

Tüm işlemler Weblate kopyasının GnuPG anahtarı tarafından imzalanabilir.

1. Turn on `WEBLATE_GPG_IDENTITY`. (Weblate will generate a GnuPG key when needed and will use it to sign all translation commits.)

Bu özellik için GnuPG 2.1 ya da üzerindeki bir sürüm kurulu olmalıdır.

Anahtarı `DATA_DIR` içinde bulabilirsiniz. Herkese açık anahtar “Hakkında” sayfasında görüntülenir:


[Dashboard](#)
[Projects](#)
[Languages](#)
[Checks](#)
[Register](#)
[Sign in](#)

[About Weblate](#) / [Weblate keys](#)

[About Weblate](#)
[Statistics](#)
[Keys](#)

Public RSA 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
AAAAB3NzaC1yc2EAAAADAQABAAQADg1iQ7xGwqmOPyvPZhlIDtwM/5jP4PZEqsPU5Da1u+SY6hnT+MjRqAyf9EXctGqHvKjHW7MhdPu+JIITZDKIhQ9A6RebGqEFKI
Weblate
```

Commit signing

All commits made with Weblate are signed with the GPG key 02FD604282D10E708EA2A4232728FF81E12CA381, for which the corresponding public key is found below.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGNBGGQ9HxcBDACoK7rkWGS56+w8oAxYjhgNFrFZamNQKyclp247WonVntyVEXsg
3u3nRgo97oL4Kf6tNe0K5L63CgF9aPLNPu9j8VQVW3iX2LO04d1A15qdZ35Ko/wl
GIIZZg5K/kBffHrmGo+fbNZR5Jj3CAQwVZihsPrhbO6YMm8gKKatE3P4p4CD+T11
x7nfxT6EZb/nenKdtB1XULPT65srkiDkoUXN2eWWIBqymXu2dTL+xqLmdZITZ9Ac
qvHgQrZXV141BC6JGqW663bOSUnsRxms5dwGQCrrn0ZWn342eV2nISBg/WrN+s9tm
LLYjLYw9v4CpbLtu78bi9IAG4AsNv3jogyUI5sn2EOHk66qzQIW0JKgK4x8ftjee
VsacOmF24PQ0sbI07t/W3Hg8zCcB7JCaWXPmhfyF6pBIECNhjyuyG59zV1OJ/f3l
8M0FyMq1Q1stS8GZN4FblLqGLfbdLh0NEiejU8x/ARK8saEw3w6bPHXvpAhO+sSm
ICV3N5Dt7c6SIVEAEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlLmNvbT6J
Ac4EEwEKADgWIQQC/WBCgtEOcl6ipCMnKP+B4SyjgQUUCZD0fFwlbAwULCQgHAgYV
CgkICwIEFglDAQIeAQIXgAAKCRAnKP+B4SyjgT0vDACZvWQogJSMshF5hsbrecZR
H1itkrLMRUrNE/HH2mZ+3FBPy3r6SIKYk+WjJjEkAWDZWQtyuj5VdZgLSprCUhyo
-----
```

Powered by Weblate 4.17

[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`.

Ayrıca bakınız:

`WEBLATE_GPG_IDENTITY`

2.21.7 Hızı sınırlamak

3.2 sürümünde değişti: Hız sınırlaması artık daha ayrıntılı yapılandırabiliyor.

4.6 sürümünde değişti: Hız sınırlaması artık süper kullanıcılara uygulanmıyor.

Weblate üzerinde bazı işlemlere hız sınırlaması uygulanır. `RATELIMIT_WINDOW` saniye içinde en fazla `RATELIMIT_ATTEMPTS` girişim yapılmasına izin verilir. Kullanıcı daha sonra `RATELIMIT_LOCKOUT` süreyle engellenir. Kapsamlara özgü ayarlar da vardır. Örneğin `RATELIMIT_CONTACT_ATTEMPTS` ya da `RATELIMIT_TRANSLATE_ATTEMPTS`. Aşağıdaki tabloda, kullanılabilecek kapsamların tam listesini görebilirsiniz.

Hız sınırlaması uygulanan işlemler şunlardır:

Ad	Kapsam	İzin verilen girişimler	Hız sınırlaması	Kilitleme süresi
Hesap açılışı	REGISTRATION	5	300	600
Yöneticilere ileti göndermek	MESSAGE	2	300	600
Oturum açarken parola kimlik doğrulaması	LOGIN	5	300	600
Site genelinde arama	SEARCH	6	60	60
Çeviri	TRANSLATE	30	60	600
Sözlüğe ekleme	GLOSSARY	30	60	600
Yeni bir dil çevirisi başlatma	LANGUAGE	2	300	600
Yeni proje oluşturma	PROJECT	5	600	600

Bir kullanıcı `AUTH_LOCK_ATTEMPTS` kez oturum açamazsa, parola sıfırlama işlemini yapana kadar hesabın parola kimlik doğrulaması kapatılır.

Ayarlar, seçenek adına `WEBLATE_` ön eki eklenerek Docker kapsayıcısına da uygulanabilir. Örneğin `RATELIMIT_ATTEMPTS` seçeneği `WEBLATE_RATELIMIT_ATTEMPTS` olur.

API için ayrı hız sınırı ayarları vardır. Bilgi almak için: [API hız sınırı](#).

Ayrıca bakınız:

[Hızı sınırlamak](#), [Ters vekil sunucu arkasında çalıştırmak](#), [API hız sınırı](#)

2.21.8 Fedora iletileri bütünleştirilmesi

Fedora Messaging, Weblate üzerinde yapılan tüm değişiklikleri ileten AMQP temelli bir yayıncıdır. Bunu kullanarak Weblate üzerindeki değişikliklere ek hizmetler bağlayabilirsiniz.

Fedora Messaging bütünleştirilmesi, `weblate-fedora-messaging` ile ayrı bir Python modülü olarak kullanılabilir. Kurulum yönergeleri için https://github.com/WeblateOrg/fedora_messaging/ adresine bakabilirsiniz.

2.22 Weblate platformunu özelleştirmek

Django ve Python kullanarak zenginleştirilebilir ve özelleştirilebilirsiniz. Herkesin yararlanabilmesi için değişikliklerinizi yukarı akışa katın. Bu yöntem, bakım maliyetlerinizi azaltır. İç arayüzler değiştirilirken veya kod yeniden düzenlenirken Weblate üzerindeki kod ele alınır.

Uyarı: Ne iç arayüzler ne de kalıplar kararlı bir API olarak kabul edilmez. Lütfen her yükseltme için kendi özelleştirmelerinizi gözden geçirin. Arayüzler veya bağlamları önceden bildirilmeksizin değişebilir.

Ayrıca bakınız:

[Weblate için katkıda bulunmak](#)

2.22.1 Bir Python modülü oluşturmak

Python ile ilgili değilseniz, temel bilgileri açıklayan ve eğitimlerin bulunduğu ‘Yeni başlayanlar için Python <<https://www.python.org/about/gettingstarted/>>’ bölümüne bakmak isteyebilirsiniz.

Özel Python kodu (modül olarak adlandırılır) içeren bir dosya yazmak için, sistem yolunda (genellikle `/usr/lib/python3.9/site-packages/` gibi bir şey) veya yorumlayıcı arama yoluna da eklenen Weblate klasöründe bir depolama alanına gerek duyulur.

3.8-5 sürümünde geldi: *Docker kullanıldığında*, Python modüllerini `/app/data/python/` (see *Docker kapsayıcısı birimleri*) içine yerleştirebilirsiniz. Böylece Weblate bunları yükleyebilir. Örneğin bir *ayar değiştirme dosyası* kullanabilirsiniz.

Daha da iyisi, özelleştirmenizi uygun bir Python paketine dönüştürebilirsiniz:

1. Paketiniz için bir klasör oluşturun (`weblate_customization` kullanacağız).
2. İçinde paketi açıklayan bir `setup.py` dosyası oluşturun:

```
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. Özelleştirme kodunun bulunduğu Python modülü için bir klasör oluşturun (`weblate_customization` olarak da adlandırılır).
4. İçinde bir `__init__.py` dosyası oluşturarak modülün Python tarafından içe aktarabildiğinden emin olun.
5. Bu paket artık **`pip install -e`** kullanılarak kurulabilir. Bilgi almak için: [Editable installs](#).
6. Kurulduktan sonra, modül Weblate yapılandırmasında kullanılabilir (`weblate_customization.checks.FooCheck` gibi).

Paket yapınız şöyle görünmelidir:

```
weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py
```

<<https://github.com/WeblateOrg/customize-example>> adresinde bir Weblate özelleştirme örneği bulabilirsiniz. Bu örnek aşağıda açıklanan tüm konuları kapsar.

2.22.2 Logoyu değiştirmek

1. Üzerine yazmak istediğiniz durağan dosyaların bulunduğu basit bir Django uygulaması oluşturun (bilgi almak için: *Bir Python modülü oluşturmak*).

Marka şu dosyalarda bulunur:

icons/weblate.svg

Gezinme çubuğunda görüntülenecek logo.

logo-*.png

Ekran çözünürlüğüne ve tarayıcıya bağlı olarak internet simgeleri.

favicon.ico

Eski tarayıcılar tarafından kullanılan internet simgesi.

weblate-*.png

Botlar ve anonim kullanıcılar için avatarlar. Bazı tarayıcılar bunları kısayol simgeleri olarak kullanır.

email-logo.png

Bildirim e-postalarında kullanılır.

2. `INSTALLED_APPS` içine ekleyin:

```
INSTALLED_APPS = (
    # Add your customization as first
    "weblate_customization",
    # Weblate apps are here...
)
```

3. İstemcilere sunulacak durağan dosyaları derlemek için `weblate collectstatic --noinput` komutunu yürütün.

Ayrıca bakınız:

How to manage static files (e.g. images, JavaScript, CSS), *Durağan dosyalar sunmak*

2.22.3 Özel kalite denetimleri, eklentiler ve kendiliğinden düzeltmeler

Weblate üzerine *Özel kendiliğinden düzeltmeler*, *Kendi denetimlerinizi oluşturmak* ya da *Eklenti yazma* kodunuzu kurmak için:

1. Dosyaları Weblate özelleştirmesinin bulunduğu Python modülünüze yerleştirin (bilgi almak için: *Bir Python modülü oluşturmak*).
2. Python sınıfının tam yolunu özel ayarlara ekleyin (`WEBLATE_ADDONS`, `CHECK_LIST` ya da `AUTO-FIX_LIST`):

```
# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)

# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)

# Add-ons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Ayrıca bakınız:

Özel kendiliğinden düzeltmeler, *Kendi denetimlerinizi oluşturmak*, *Eklenti yazma*, *Eklentiden betikleri çalıştırma*

2.23 Yönetim arayüzü

Yönetim bölümü, `/manage/` adresinden yönetim ayarlarının yapılmasını sağlar. Yönetici yetkileri ile oturum açmış kullanıcılar tarafından kullanılabilir ve sağ üstteki İngiliz anahtarı simgesi kullanılarak erişilebilir:

The screenshot shows the Weblate management interface. The top navigation bar includes 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a 'Manage' section with tabs for 'Weblate status', 'Backups', 'Translation memory', 'Performance report', 'SSH keys', 'Alerts', 'Repositories', 'Users', and 'Teams'. The 'Weblate status' tab is active, showing 'Weblate version' as 4.17, 'Support status' as Community support, and buttons for 'Purchase support package' and 'Donate to Weblate'. Below this is an 'Activate support package' section with an 'Activation token' input field and buttons for 'Activate' and 'Purchase support package'.

Weblate ile ilgili temel bilgileri içerir:

- Destek durumu. Bilgi almak için [Weblate için destek almak](#)
- Yedekler. Bilgi almak için [Weblate yedeğini alma ve taşıma](#)
- Paylaşılan çeviri belleği. Bilgi almak için [Çeviri belleği](#)
- Weblate sağlığı ve Celery kuyruklarının uzunluğunu gösteren başarımlar raporu
- SSH anahtar yönetimi. Bilgi almak için [SSH depoları](#)
- Tüm bileşenlerin uyarı özeti. Bilgi almak için alerts

2.23.1 Django yönetim arayüzü

Uyarı: Bu arayüz düşük düzeyde olduğundan kullanırken dikkatli olun. Genellikle bunu kullanmaya gerek duymamalıdır. Çoğu şeye Weblate kullanıcı arayüzü veya API üzerinden rahatça ulaşılabilir.

Bu bölümden kullanıcılar, çeviriler gibi veri tabanında depolanan nesneleri ve diğer ayarları yönetebilirsiniz:

Weblate administration

WELCOME **WEBLATE TEST** [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Site administration

REPORTS

[Weblate support status](#)[Status of repositories](#)[SSH keys](#)[Performance report](#)[Translation memory](#)

ACCOUNTS

[Audit log entries](#)[+ Add](#)[Change](#)[User profiles](#)[+ Add](#)[Change](#)[Verified e-mails](#)[+ Add](#)[Change](#)

AUTH TOKEN

[Tokens](#)[+ Add](#)[Change](#)

AUTHENTICATION

[Groups](#)[+ Add](#)[Change](#)[Roles](#)[+ Add](#)[Change](#)[Users](#)[+ Add](#)[Change](#)

BILLING

[Billing plans](#)[+ Add](#)[Change](#)[Customer billings](#)[+ Add](#)[Change](#)[Invoices](#)[+ Add](#)[Change](#)

FONTS

[Font groups](#)[+ Add](#)[Change](#)[Fonts](#)[+ Add](#)[Change](#)

LEGAL

[TOS agreements](#)[+ Add](#)[Change](#)

PERIODIC TASKS

[Clocked](#)[+ Add](#)[Change](#)[Crontabs](#)[+ Add](#)[Change](#)[Intervals](#)[+ Add](#)[Change](#)[Periodic tasks](#)[+ Add](#)[Change](#)[Solar events](#)[+ Add](#)[Change](#)

PYTHON SOCIAL AUTH

[Associations](#)[+ Add](#)[Change](#)[Nonces](#)[+ Add](#)[Change](#)[User social auths](#)[+ Add](#)[Change](#)

SCREENSHOTS

[Screenshots](#)[+ Add](#)[Change](#)

TRANSLATION MEMORY

[Translation memory entries](#)[+ Add](#)[Change](#)

WEBLATE CONFIGURATION

[Settings](#)[+ Add](#)[Change](#)

WEBLATE LANGUAGES

[Languages](#)[+ Add](#)[Change](#)

WEBLATE TRANSLATIONS

[Announcements](#)[+ Add](#)[Change](#)[Component lists](#)[+ Add](#)[Change](#)[Components](#)[+ Add](#)[Change](#)[Contributor agreements](#)[+ Add](#)[Change](#)[Projects](#)[+ Add](#)[Change](#)

Recent actions

[My actions](#)

None available

In the *Raporlar* bölümünden, sitenizin durumuna bakabilirsiniz. *Üretim kurulumu* kipine geçebilir ya da *Depolara erişmek* erişimi için SSH anahtarlarını yönetebilirsiniz.

Bölümlerden herhangi birinin altındaki veri tabanı nesnelerini yönetin. Büyük olasılıkla en ilginç olanı, çevrilebilir projeleri yönetebileceğiniz *Weblate çevirileri*. Bilgi almak için: *Proje yapılandırması* ve *Bileşen yapılandırması*.

Weblate dilleri bölümünde dil tanımları bulunur. Bilgi almak için: *Dil tanımları*.

Proje ekleme

Eklenen bir proje, tüm bileşenler için bir kapsayıcı görevi görür. Genellikle bir yazılım ya da kitap parçası için bir proje oluşturursunuz (kullanılacak parametreler ile ilgili bilgi almak için: *Proje yapılandırması* bölümüne bakabilirsiniz):

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:	<input type="text" value="WeblateOrg"/>
	Display name
URL slug:	<input type="text" value="weblateorg"/>
	Name used in URLs and filenames.
Project website:	<input type="text" value="https://weblate.org/"/>
	Main website of translated project.
Translation instructions:	<div><input type="text" value="https://weblate.org/contribute/"/></div>
	You can use Markdown and mention users by @username.
<input checked="" type="checkbox"/> Set "Language-Team" header	Lets Weblate update the "Language-Team" file header of your project.
<input checked="" type="checkbox"/> Use shared translation memory	Uses the pool of shared translations between projects.
<input checked="" type="checkbox"/> Contribute to shared translation memory	Contributes to the pool of shared translations between projects.
Access control:	<div>Protected ▼</div>
	How to restrict access to this project is detailed in the documentation.
<input type="checkbox"/> Enable reviews	Requires dedicated reviewers to approve translations.
<input type="checkbox"/> Enable source reviews	Requires dedicated reviewers to approve source strings.
<input checked="" type="checkbox"/> Enable hooks	Whether to allow updating this repository by remote hooks.
Language aliases:	<div><input type="text"/></div>
	Comma-separated list of language code mappings, for example: en_GB:en,en_US:en
Machinery settings:	<div><input type="text" value="0"/></div>

Ayrıca bakınız:

Proje yapılandırması

Çift dilde bileşenler

Bir proje ekledikten sonra, ona çeviri bileşenleri ekleyebilirsiniz. (kullanılabilecek parametrelerle ilgili bilgi almak için *Bileşen yapılandırması* bölümüne bakabilirsiniz):

Website administration

HomeWebsiteAdministrationComponentsAdd Component

CREATE NEW COMPONENT

Add Component

Required fields are marked in bold.

Component name

Language names

Required name

URL slug

Language names

Name used in URLs and filenames

Project

Multi-selecting

Version control system

Git

Version control system is used to access your repository containing translations. This can also connect additional integration with third party providers to submit or request updates.

Source code repository

https://github.com/WordPress/po4j-core.git

URL of repository, can include a commit reference to check it with other components

Repository push URL

URL of a push repository pointing to a remote if it exists

Repository browser

https://github.com/WordPress/po4j-core/branches/WordPress441

URL to repository browser (Browsers, Browsers) and (Git) as filename and file placeholders. You might want to strip leading directory by using {filename}/{name}

Exported repository URL

URL of repository where assets can fetch changes from website

Domain string being reported

address

Email address for reports or errors in report output. Leave empty for no email.

Repository branch

Repository branch to translate

Push branch

Branch for pushing changes, leave empty to use repository branch

File mask

wp-content/languages/*/*-*.mo

Place of file in translation where is repository url used - instead of language code. For example you can have wp-content/languages/*/*-*.mo

Modeling build language file

Elements of translation base file, consisting of string and their source, it is recommended for modeling translation formats.

Get base file

Wherever you will be able to use the base file for modeling translations.

Intermediate language file

Elements of intermediate translations. It must pass this is a translation file provided to developers and is used when creating actual source file.

Template for new translation

wp-content/languages/*/*-*.mo

Elements of the used for creating new translations. For precise choose per file.

File format

gettext file

☐ Locked

Locked components will not get any translation updates.

☒ Allow translation suggestions

Enable translation updates in other components and enable automatic translation in this one

☐ Turn on suggestions

Whether to allow translation suggestions or not

☐ Suggestion policy

Users can only vote for suggestions and can make direct translations.

Automatic suggestion:

0

Approximately control suggestions with the number of votes, use 0 for no auto.

Translation flags

Additional control flags and flags to influence website behavior

Inflected details

0

List of details which can not be ignored

Translation license

GNU General Public License v3.0 or later

Contributor agreement

Your agreement which needs to be approved before a user can translate this component.

Adding new translation

Click new language file

How to handle requests for creating new translations

Language code style

Default based on the file format

Customize language code used to generate the filenames for translations created by Poedit.

☐ Message editing

Enables using existing message strings from website. If your strings are extracted from the source code or managed externally you probably want to keep it disabled.

Merge style

Polish

Other member website should merge the upstream repository or release changes with it.

Current message when translating

Translated using Website {0} language {name} {1}

Currently translated {0} from {language_name} {1} to {site_language} {2} {3} {4} {5} {6} {7} {8} {9} {10} {11} {12} {13} {14} {15} {16} {17} {18} {19} {20} {21} {22} {23} {24} {25} {26} {27} {28} {29} {30} {31} {32} {33} {34} {35} {36} {37} {38} {39} {40} {41} {42} {43} {44} {45} {46} {47} {48} {49} {50} {51} {52} {53} {54} {55} {56} {57} {58} {59} {60} {61} {62} {63} {64} {65} {66} {67} {68} {69} {70} {71} {72} {73} {74} {75} {76} {77} {78} {79} {80} {81} {82} {83} {84} {85} {86} {87} {88} {89} {90} {91} {92} {93} {94} {95} {96} {97} {98} {99} {100} {101} {102} {103} {104} {105} {106} {107} {108} {109} {110} {111} {112} {113} {114} {115} {116} {117} {118} {119} {120} {121} {122} {123} {124} {125} {126} {127} {128} {129} {130} {131} {132} {133} {134} {135} {136} {137} {138} {139} {140} {141} {142} {143} {144} {145} {146} {147} {148} {149} {150} {151} {152} {153} {154} {155} {156} {157} {158} {159} {160} {161} {162} {163} {164} {165} {166} {167} {168} {169} {170} {171} {172} {173} {174} {175} {176} {177} {178} {179} {180} {181} {182} {183} {184} {185} {186} {187} {188} {189} {190} {191} {192} {193} {194} {195} {196} {197} {198} {199} {200} {201} {202} {203} {204} {205} {206} {207} {208} {209} {210} {211} {212} {213} {214} {215} {216} {217} {218} {219} {220} {221} {222} {223} {224} {225} {226} {227} {228} {229} {230} {231} {232} {233} {234} {235} {236} {237} {238} {239} {240} {241} {242} {243} {244} {245} {246} {247} {248} {249} {250} {251} {252} {253} {254} {255} {256} {257} {258} {259} {260} {261} {262} {263} {264} {265} {266} {267} {268} {269} {270} {271} {272} {273} {274} {275} {276} {277} {278} {279} {280} {281} {282} {283} {284} {285} {286} {287} {288} {289} {290} {291} {292} {293} {294} {295} {296} {297} {298} {299} {300} {301} {302} {303} {304} {305} {306} {307} {308} {309} {310} {311} {312} {313} {314} {315} {316} {317} {318} {319} {320} {321} {322} {323} {324} {325} {326} {327} {328} {329} {330} {331} {332} {333} {334} {335} {336} {337} {338} {339} {340} {341} {342} {343} {344} {345} {346} {347} {348} {349} {350} {351} {352} {353} {354} {355} {356} {357} {358} {359} {360} {361} {362} {363} {364} {365} {366} {367} {368} {369} {370} {371} {372} {373} {374} {375} {376} {377} {378} {379} {380} {381} {382} {383} {384} {385} {386} {387} {388} {389} {390} {391} {392} {393} {394} {395} {396} {397} {398} {399} {400} {401} {402} {403} {404} {405} {406} {407} {408} {409} {410} {411} {412} {413} {414} {415} {416} {417} {418} {419} {420} {421} {422} {423} {424} {425} {426} {427} {428} {429} {430} {431} {432} {433} {434} {435} {436} {437} {438} {439} {440} {441} {442} {443} {444} {445} {446} {447} {448} {449} {450} {451} {452} {453} {454} {455} {456} {457} {458} {459} {460} {461} {462} {463} {464} {465} {466} {467} {468} {469} {470} {471} {472} {473} {474} {475} {476} {477} {478} {479} {480} {481} {482} {483} {484} {485} {486} {487} {488} {489} {490} {491} {492} {493} {494} {495} {496} {497} {498} {499} {500} {501} {502} {503} {504} {505} {506} {507} {508} {509} {510} {511} {512} {513} {514} {515} {516} {517} {518} {519} {520} {521} {522} {523} {524} {525} {526} {527} {528} {529} {530} {531} {532} {533} {534} {535} {536} {537} {538} {539} {540} {541} {542} {543} {544} {545} {546} {547} {548} {549} {550} {551} {552} {553} {554} {555} {556} {557} {558} {559} {560} {561} {562} {563} {564} {565} {566} {567} {568} {569} {570} {571} {572} {573} {574} {575} {576} {577} {578} {579} {580} {581} {582} {583} {584} {585} {586} {587} {588} {589} {590} {591} {592} {593} {594} {595} {596} {597} {598} {599} {600} {601} {60

Ayrıca bakınız:

Bileşen yapılandırması, İki dilli ve tek dilli biçimler

Tek dilde bileşenler

Bunların daha kolay çevrilmesi için, ileti kimliklerinin ilgili kaynak diliyle (genellikle İngilizce) eşlenebileceği bir kalıp dosyası sağlayın (kullanılacak parametrelerle ilgili bilgi almak için *Bileşen yapılandırması* bölümüne bakabilirsiniz):

Weblate administration
[Help](#)
[Weblate 1.10.0](#)
[Return to Weblate](#)
[Documentation](#)
[Feedback](#)
[Feedback](#)

[Home](#)
[Recent translations](#)
[Components](#)
[Add component](#)

Add Component
[Cancel](#)
[Save](#)

Required fields are marked with *.

Component name

URL slug

Project

Weblate

Version control system

Git

Source code repository

Repository push URL

Repository browser

Expected repository URL

Source string file reporting address

Repository branch

Push branch

File mask

Metadata base language file

Use base file

☒

Intermediate language file

Language for new translations

File format

Android String Resources

☐ Locked

☒ Allow translation propagation

☒ Use on all suggestions

☐ Suggestion voting

Automatic suggestions

1

Translation flags

Enhanced checks

Translation browser

MT Locomote

Contributor agreement

Add new translation

Create new language file

Language code style

Default based on the file format

☐ Manage strings

Merge style

Rebase

Control message when translating

Translated using Weblate (% language_name %)

Control message when adding translation

Added translation using Weblate (% language_name %)

Control message when merging translation

Deleted translation using Weblate (% language_name %)

Control message when merging translation

Merge branch (%component_name_branch %) into Weblate

Control message when add-on makes a change

Update translation files

Merge request message

Translations update from (% url_repo %)

☒ Push on commit

Age of changes to commit

24

☒ Push on save

Source language

English

Language filter

-

Variety regular expression

Priority

Medium

☐ Restricted component

Show in projects

Weblate

☐ Use as a glossary

Glossary color

Silver

Remote version

Local version

[Cancel and edit source](#)
[Save and return to list](#)
[Save](#)

Ayrıca bakınız:

Bileşen yapılandırması, İki dilli ve tek dilli biçimler

2.24 Weblate için destek almak

Weblate, topluluk destekli telif hakkı olmayan özgür bir yazılımdır. Aboneler ek ücret ödemeden öncelikli destek alırlar. Ön ödemeli yardım paketleri herkes tarafından kullanılabilir. Geçerli destek teklifleri ile ilgili bilgi almak için: <https://weblate.org/support/>.

2.24.1 Destek bütünleştirmesi

3.8 sürümünde geldi.

Satın alınan destek paketleri, isteğe bağlı olarak, Weblate [abonelik yönetimi](#) arayüzünüzle bütünleştirilebilir ve buradan bir bağlantı bulabilirsiniz. Kopyanızla ilgili temel bulut sunucusu bilgileri de bu şekilde Weblate tarafına bildirilir.

W

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

4.17 — 996b857d85695da8e748a95f7678feb68ff2c3f4

Support status

Community support

Refresh support status

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

2.24.2 Weblate üzerine gönderilen veriler

- Weblate kopyanızın yapılandırıldığı adres
- Sitenizin başlığı
- Kullandığınız Weblate sürümü
- Weblate veri tabanınızdaki bazı nesnelerin günlükleri (projeler, bileşenler, diller, kaynak dizgeler ve kullanıcılar)
- Kopyanızın herkese açık SSH anahtarı

Ayrıca, *Weblate uygulamasını keşfedin* açık olduğunda:

- Herkese açık projelerin listesi (ad, adres ve site)

Başka herhangi bir veri gönderilmez.

2.24.3 Bütünleştirme hizmetleri

- Destek paketinizin hala geçerli olup olmadığını görmek
- *Weblate tarafından sunulan yedekleme depolama alanı*
- *Weblate uygulamasını keşfedin*

İpucu: Satın alınan destek paketleri o sırada etkinleştirilir ve bütünleştirilmeden kullanılabilir.

2.24.4 Weblate uygulamasını keşfedin

4.5.2 sürümünde geldi.

Discover Weblate, kullanıcıların Weblate sunucularını ve topluluklarını bulmasını kolaylaştıran bir abonelik hizmetidir. Kullanıcılar <<https://weblate.org/discover/>> adresinden kaydedilmiş hizmetlere göz atabilir ve katkıda bulunulacak projeleri bulabilir.

Listelenme

İpucu: Discover Weblate hizmetine katılmak, Weblate kopyanızın sunucunuz ile ilgili bazı bilgiler göndermesini sağlar. Bilgi almak için: *Weblate üzerine gönderilen veriler*.

Sunucunuzun etkin bir destek aboneliği ile Discovery Weblate hizmetinde listelenmesi için (bilgi almak için: *Destek bütünleştirilmesi*) yönetim panosundan şu seçeneği açmanız yeterlidir:

W 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

4.17 — 996b857d85695da8e748a95f7678feb68ff2c3f4

Support status

Community support

Refresh support status

Discover Weblate

Your Weblate is not listed on weblate.org

Enable discovery

Browse 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 4.17 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Sunucunuzu destek aboneliği olmadan Discover Weblate hizmetinde listeleme:

1. <https://weblate.org/user/> adresinden kendinizi kaydedin
2. Weblate sunucunuzu <https://weblate.org/subscription/discovery/> adresindeki Discover Weblate veri tabanına kaydedin
3. Weblate kopyanızda hizmetin etkinleştirmesini onaylayın ve *Weblate Discovery kullanılsın* düğmesini kullanarak Weblate yönetimi sayfanızda keşif listesini açın:

424

Bölüm 2. Yönetici belgeleri

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	4.17 — 996b857d85695da8e748a95f7678feb68ff2c3f4
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 4.17 About Weblate Legal Contact Documentation Donate to Weblate

Listelemeyi özelleştirme

`<https://weblate.org/user/>` adresinden bir metin ve görsel (570 x 260 piksel) belirterek listelemeyi özelleştirebilirsiniz.

2.25 Yasal belgeler

Not: Burada, belirli yasal yargı yetkisindeki bölgelerde Weblate işletmek için gerek duyabileceğiniz çeşitli yasal bilgileri bulabilirsiniz. Herhangi bir kesinlik veya doğruluk garantisi olmadan yol göstermek için verilmiştir. Weblate kullanımınızın yürürlükteki tüm yasa ve yönetmeliklere uygun olmasını sağlamak asıl olarak sizin sorumluluğunuzdadır.

2.25.1 Lisans uyumluluğu

Weblate, [REUSE 3.0](#) uyumlu lisans koşulları altında dağıtılır.

2.25.2 ITAR ve diğer ihracat denetimleri

Weblate, kendi veri merkezinde veya özel sanal bulutunuzda çalıştırılabilir. Bu nedenle, ITAR veya diğer ihracat denetimi bilgilerini tutmak için kullanılabilir. Ancak bu uyumluluğu sağlamak son kullanıcıların sorumluluğundadır.

Hosted Weblate hizmeti, ITAR veya diğer ihracat denetimleriyle uyumluluğu açısından denetlenmemiştir ve şu anda ülkeye göre çeviri erişimini kısıtlama olanağı sunmamaktadır.

2.25.3 ABD şifreleme denetimleri

Weblate içinde herhangi bir şifreleme kodu bulunmaz. Ancak kimlik doğrulama, veri bütünlüğü ve gizlilik için şifreleme kullanan üçüncü taraf bileşenleri kullandığından ihracat denetimlerine tabi olabilir.

Büyük olasılıkla Weblate, ECCN 5D002 veya 5D992 olarak sınıflandırılacaktır ve herkese açık özgür yazılım olarak EAR kapsamına girmemelidir (bilgi almak için: [EAR kapsamına girmeyen şifreleme öğeleri](#)).

Weblate tarafından kullanılan yazılım bileşenleri (yalnızca şifreleme işleviyle ilgili bileşenler listelenmiştir):

Python

Bilgi almak için: https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is_Python_subject_to_export_laws.3F

GnuPG

Weblate tarafından isteğe bağlı olarak kullanılır

Git

Weblate tarafından isteğe bağlı olarak kullanılır

curl

Git tarafından kullanılır

OpenSSL

Python ve cURL tarafından kullanılır

Şifreleme anahtarlarının gücü, Weblate yapılandırmasına ve etkileşime girdiği üçüncü taraf bileşenlerine bağlıdır. Ancak herhangi bir iyi kurulumda, tüm dış aktarma kısıtlamalı şifreleme işlevlerini kapsayacaktır:

- Simetrik algoritma için 56 bitten fazla
- Asimetrik bir algoritma için 512 biti aşan tamsayıların çarpanlara ayrılması
- Asimetrik bir algoritma için 512 biti aşan bir sonlu alanın çarpım grubundaki ayrık logaritmaların hesaplanması
- Asimetrik bir algoritma için yukarıdakinden farklı bir grupta 112 biti aşan ayrık logaritma

Weblate üzerinde herhangi bir şifreleme etkinleştirme özelliği yoktur. Ancak hiçbir şifreleme kodu bulunmayacak şekilde yapılandırılabilir. Şifreleme özellikleri şu işlemlerde kullanılır:

- Güvenli iletişim kuralı (HTTPS) kullanarak uzak sunuculara erişme
- Kod işlemleri için imza oluşturma (PGP)

Ayrıca bakınız:

[Açık kaynaklı yazılım üzerindeki ihracat denetimleri \(EAR\)](#)

3.1 Weblate için katkıda bulunmak

Weblate platformunu geliştirmenin pek çok yolu vardır. Kodlama, görsel tasarım, belge hazırlama, destek olma veya fikir verme gibi size uygun olan birini seçebilirsiniz:

- *Weblate sorunlarını bildirmek*
- *Weblate kodlarına katkıda bulunmak*
- *Weblate modüllerine katkıda bulunmak*
- *Weblate arayüzünü çevirme*
- *Weblate belgelerine katkıda bulunun*
- *Weblate tartışmaları*
- *Weblate geliştirilme çalışmalarına maddi destek vermek*

3.1.1 Weblate arayüzünü çevirme

Weblate, gene Weblate kullanılarak sürekli olarak [çevriliyor](#). Weblate kullanımının olabildiğince çok dilde sağlanmasına yardımcı olmaktan çekinmeyin. Böylece Weblate kullanıcıları ile yaklaşır!

Kaynak dizgede olası bir hata bulursanız, bunu Weblate düzenleyicisinde bir yorumla belirtebilirsiniz. Böylece üzerinde tartışılabilir ve düzeltilebilir. Eminseniz, *Kaynak dizge konumu* bölümündeki bağlantıya tıklayabilir ve düzeltmenizle birlikte bir çekme isteği gönderebilirsiniz.

3.1.2 Weblate belgelerine katkıda bulunun

İstediğiniz belgeler sayfasını iyileştirebilirsiniz. Bunun için sayfanın sağ üst köşesindeki *GitHub üzerinde düzenle* düğmesine tıklayarak kolayca yapabilirsiniz.

Lütfen yazarken şu yönergelerle uyun:

1. Geçerliyse, belgenin bir bölümünü kaldırmayın.
2. Açık ve kolay anlaşılır bir dil kullanın. Teknoloji belgeleri yazıyorsunuz, şiir değil. Belgeleri okuyacak herkesin ana dili İngilizce değildir, düşünceli olun.
3. Emin değilseniz sormaktan çekinmeyin. Düzenleme yaparken bazı özellikler ile ilgili soru sormanız gerekiyorsa, yanıt almadan önce belgeleri değiştirmeyin. Özetle: Değiştirin ya da sorun. İkisini aynı anda yapmayın.
4. Belgeleri izlerken açıklanan işlemleri yaparak değişikliklerinizi doğrulayın.
5. Onaylama ve birleştirmenin daha kolay ve hızlı yapılması için çekme isteğinde küçük değişiklik parçaları gönderin.
6. Büyük bir makaleyi yeniden yazmak ve yapısını değiştirmek istiyorsanız, bunu iki adımda yapın:
 1. Yeniden yazın
 2. Yeniden yazma onaylandıktan, iyileştirildikten ve birleştirildikten sonra, başka bir çekme isteğinde paragrafların yapısını değiştirin.

İpucu: Belgeleri çevirebilirsiniz.

3.1.3 İç dil tanımlarını zenginleştirmek

Dil tanımları `weblate-language-data` deposunda bulunur.

Eksik dil tanımlarını `languages.csv` dosyasına ekleyebilirsiniz. Diğer dosyalar bu dosyadan oluşturulur.

3.1.4 Weblate tartışmaları

Bir fikriniz varsa ve bir sorun olarak bildirmeye uygun olup olmadığından emin değilseniz endişelenmeyin. [GitHub tartışmaları](#) ile topluluğa katılıp tartışabilirsiniz.

3.1.5 Weblate geliştirilme çalışmalarına maddi destek vermek

Weblate geliştirme çalışmalarını [bağış sayfasından](#) destekleyebilirsiniz. Buradan toplanan fonlar, özgür yazılım projeleri için ücretsiz barındırma hizmeti sağlanması ve Weblate özelliklerinin geliştirilmesi için kullanılır. Fonlama hedefleri ile gururlu bir fon sağlayıcı olarak alacağınız ödüller gibi konular için [`bağış sayfasına`](#) bakabilirsiniz.

Weblate için bağış yapan destekçiler

Weblate destekçilerinin listesi:

- 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

Bu listede yer almak ister misiniz? [Weblate için bağış](#) seçeneklerine bakabilirsiniz.

3.2 Weblate kodlarına katkıda bulunmak

Weblate kaynak kodlarını anlamak için lütfen önce *Weblate kaynak kodları*, *Weblate ön yüzü* ve *Weblate iç yapısı* bölümlerine bakın.

3.2.1 Kod tabanı ile başlamak

Weblate kod tabanına alışmak için bazı hatalar arıyorsanız, [good first issue](#) olarak etiketlenmiş olanlara bakın.

Bu konular üzerinde sormadan çalışmaya başlayabilirsiniz. Yalnızca çalışmaya başladığınızı duyurun. Böylece herkes birisinin bu konu üzerinde çalıştığını bilir.

3.2.2 Weblate platformunu yerel olarak çalıştırma

Weblate geliştirmeye başlamanın en kolay yolu *Kaynaklardan kurulum* bölümündeki yönergeleri izlemektir. Böylece, düzenlenebilir Weblate kaynaklarını içeren bir virtualenv oluşturabilirsiniz.

1. Weblate kaynak kodlarını kopyalayın:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Bir virtualenv oluşturun:

```
virtualenv .venv
.venv/bin/activate
```

3. Weblate kurulumunu yapın (bunun için bazı sistem bağımlılıklarına gerek duyulur. Bilgi almak için: *Kaynaklardan kurulum*):

```
pip install -e .
```

3. Geliştirme için yararlı olacak tüm bağımlılıkları kurun:

```
pip install -r requirements-dev.txt
```

4. Bir geliştirme sunucusu başlatın:

```
weblate runserver
```

5. Yapılandırmanıza bağlı olarak, Celery işlemlerini de başlatmak isteyebilirsiniz:

```
./weblate/examples/celery start
```

6. Bir deneme yapın (bilgi almak için: *Yerel sınama*):

```
. scripts/test-database.sh
./manage.py test
```

Ayrıca bakınız:

Kaynaklardan kurulum

3.2.3 Weblate platformunu Docker üzerinde yerel olarak çalıştırma

Docker ve docker-compose kuruluysa, yalnızca aşağıdakileri yürüterek geliştirme ortamını çalıştırabilirsiniz:

```
./rundev.sh
```

Bir geliştirme Docker kalıbı oluşturur ve başlatır. Weblate ile <http://127.0.0.1:8080/> adresinden `admin` kullanıcı adı ve `admin` parolası ile oturum açabilirsiniz. Yeni kurulum boştur, bu nedenle *Çeviri projelerini ve bileşenleri eklemek* ile ilerlemek isteyebilirsiniz.

Bunun için `Dockerfile` ve `docker-compose.yml`, `dev-docker` klasöründe bulunur.

Betik ayrıca sınamaları yürütmek için bazı parametreleri kabul eder. `test` parametresiyle çalıştırılır ve ardından herhangi bir `test` parametresi belirtilir. Örneğin sınamaları yalnızca `weblate.machine` modülünde çalıştırır:

```
./rundev.sh test --failfast weblate.machine
```

Not: Sınamaları çalıştırmadan önce Docker kapsayıcılarınızın çalışır durumda olduğundan emin olun. Bunu `docker ps` komutunu çalıştırarak denetleyebilirsiniz.

Günlükleri görüntülemek için:

```
./rundev.sh logs
```

Arka plan kapsayıcılarını durdurmak için şunu çalıştırın:

```
./rundev.sh stop
```

Betiği bağımsız değişkenler olmadan çalıştırmak, Docker kapsayıcısını yeniden oluşturur ve yeniden başlatır.

Not: Bu, üretim için uygun bir kurulum değildir. Güvenli olmayan birkaç ayar yapılmıştır, ancak geliştirmeyi kolaylaştırır.

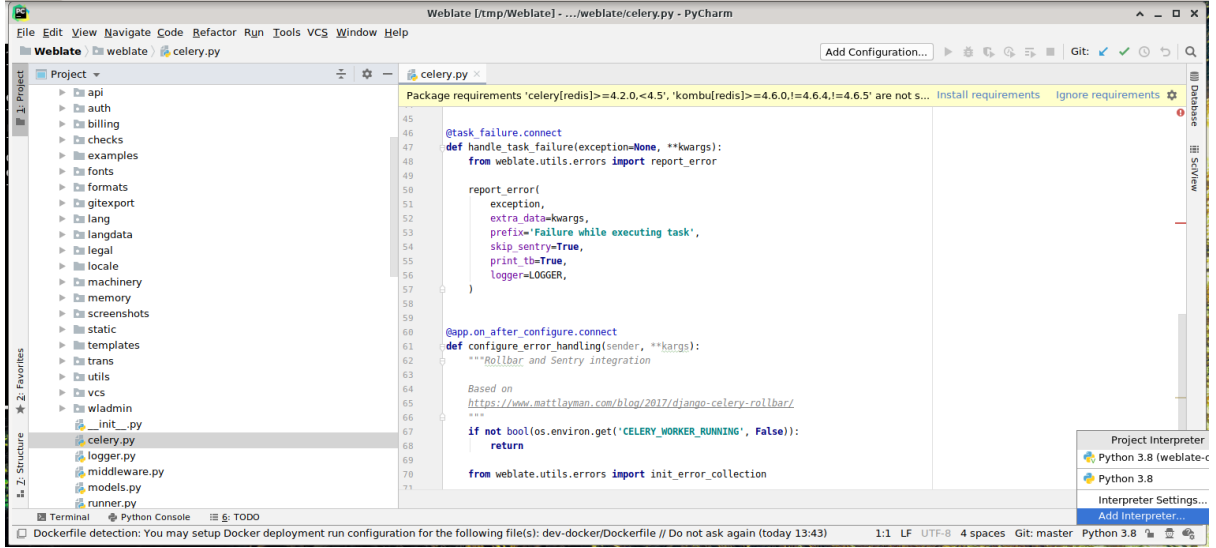
3.2.4 Devel kopyanızı ön yüklemek

Deneme çevirileri oluşturmak için `import_demo` ve yönetici kullanıcı oluşturmak için `createadmin` komutlarını kullanmak isteyebilirsiniz.

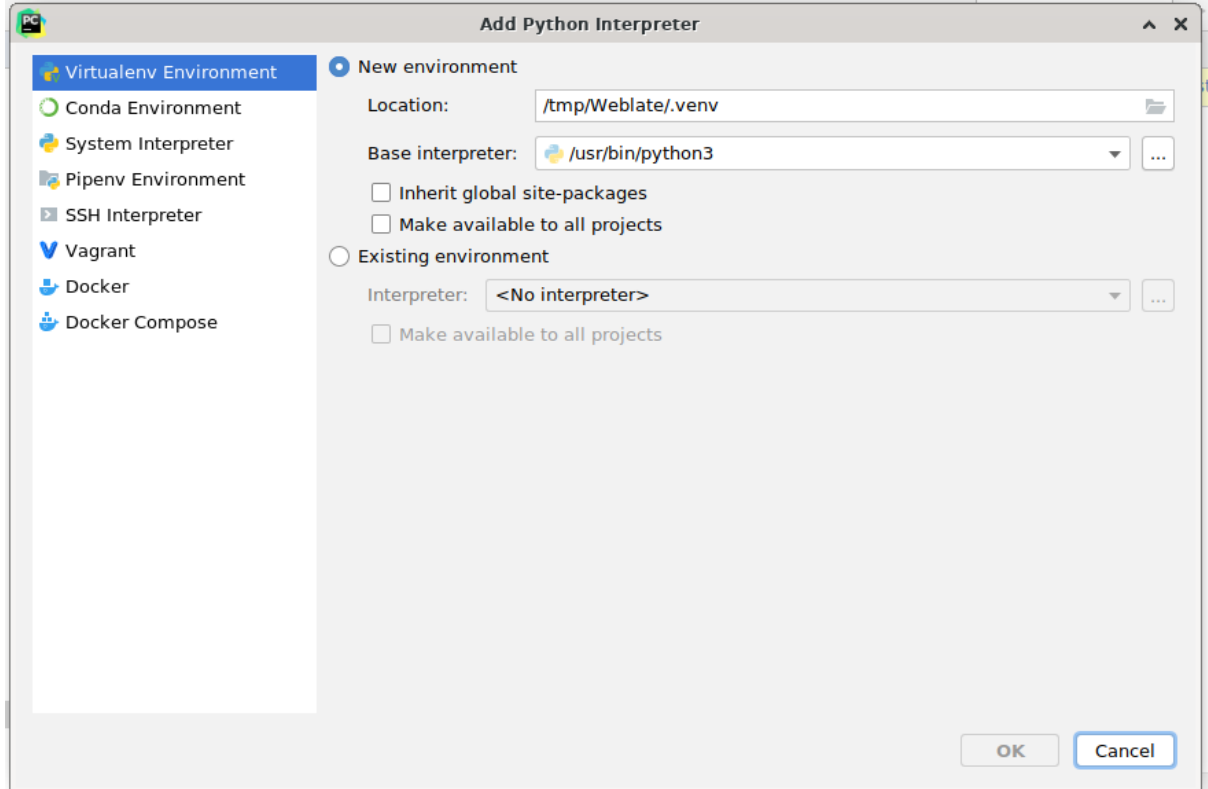
3.2.5 PyCharm ile Weblate kodlamak

PyCharm, Python için bilinen bir IDE uygulamasıdır. Weblate projenizi PyCharm içinde ayarlamanıza yardımcı olacak bazı yönergeleri aşağıda bulabilirsiniz.

GitHub deposunu bir klasöre kopyaladığınızı düşünürsek, PyCharm ile açmanız yeterlidir. IDE açıldıktan sonra, ilk adım kullanmak istediğiniz yorumlayıcıyı belirtmektir:



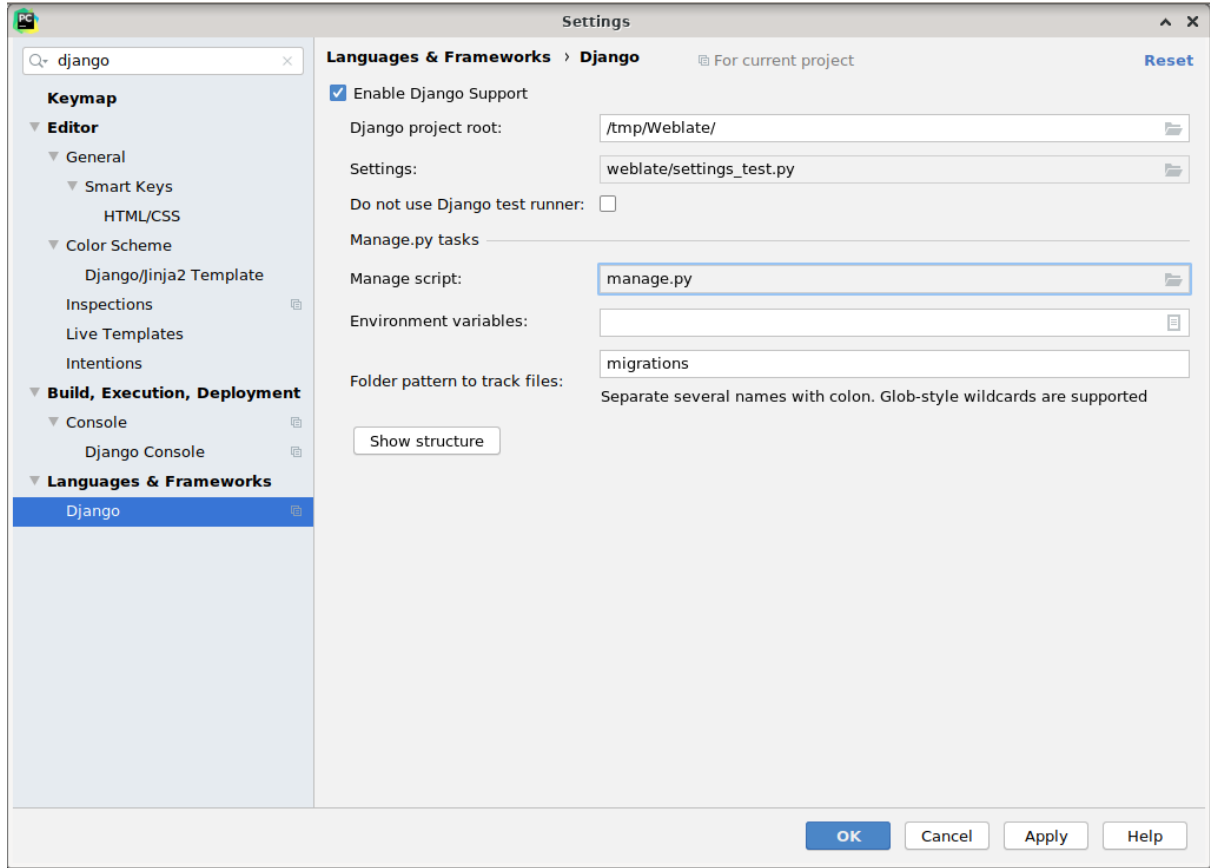
PyCharm sizin için bir virtualenv oluşturabilir ya da zaten var olan bir virtualenv kullanmayı seçebilirsiniz:



Yorumlayıcı ayarlandıktan sonra bağımlılıkları kurmayı unutmayın: Konsol üzerinden (IDE içindeki konsol varsa-ayılan olarak doğrudan belirttiğiniz virtualenv kullanır) ya da eksik bağımlılıklar ile ilgili bir uyarı aldığınızda arayüz üzerinden.

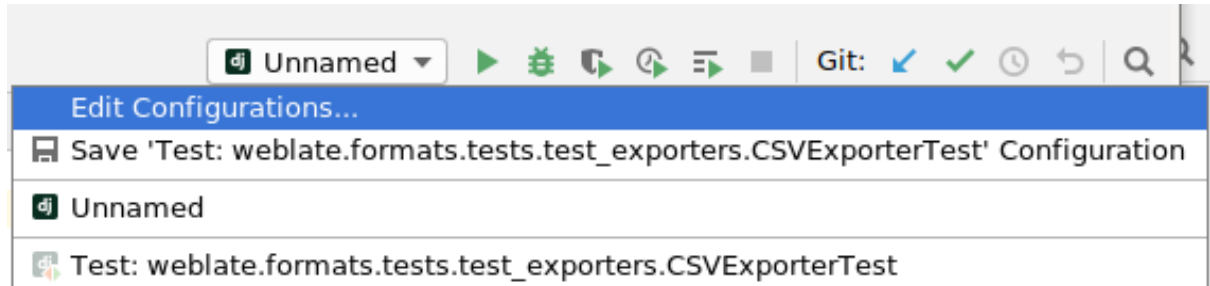
İkinci adım, PyCharm içinde yerel olarak Django kullanmak için doğru bilgiyi ayarlamaktır: Buradaki fikir, IDE üzerindeki birim sınamalarını hemen tetikleyebilmektir. Bunun için Django projesinin kök yolunu ve ayarlarına giden

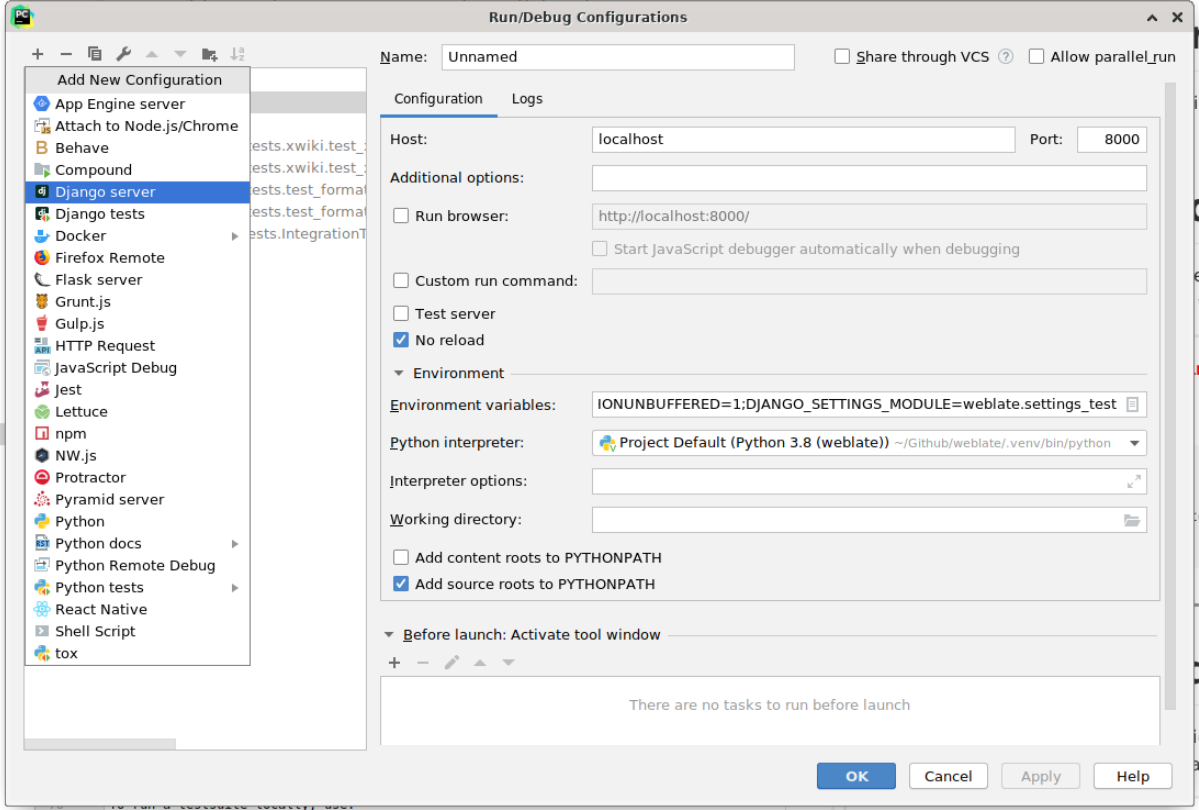
yolu belirtmeniz gerekir:



Dikkatli olun, *Django proje kökü* deponun gerçek kök klasörüdür, Weblate alt klasörü değildir. Ayarlar ile ilgili olarak, depodaki `weblate/settings_test.py` dosyasını kullanabilirsiniz. Bununla birlikte kendi ayarınızı da oluşturarak oradan da ayarlayabilirsiniz.

Son adım, sunucuyu çalıştırmak ve hata ayıklayabilmek için koda kesme noktaları koymaktır. Bunun için, yeni bir *Django Server* yapılandırması oluşturulur:





İpucu: *No reload* özelliğine dikkat edin: Dosyaları değiştirirseniz sunucunun canlı olarak yeniden yüklenmesini engeller. Bu durum, var olan hata ayıklayıcı kesme noktalarının, normalde sunucu yeniden yüklendikten sonra yok sayılacakları zaman kalıcı olmasına yol açar.

3.3 Weblate kaynak kodları

Weblate, [GitHub](#) üzerinde geliştirilmektedir. Kodu çatallayabilir ve çekme istekleri yapabilirsiniz. Başka herhangi bir biçimdeki yamalar da kabul edilir.

Ayrıca bakınız:

Weblate platformunun içeriden nasıl görüldüğünü anlamak için [Weblate iç yapısı](#) bölümüne bakabilirsiniz.

3.3.1 Kodlama rehberi

Weblate için geliştirilen kodlar [Tasarım ilkelerine uygun güvenlik](#) göz önünde bulundurularak yazılmalıdır.

Tüm kodlar, kodun davranışını açıklayan belgelerle birlikte iletilmelidir. Yöntemler, karmaşık kod blokları veya kullanıcı tarafından görülebilen özellikleri belgelemeyi unutmayın.

Herhangi bir yeni kod [PEP 484](#) türünde ipuçları kullanılmalıdır. Var olan kod henüz bunları içermediğinden bunu henüz CI ile denetlemiyoruz.

3.3.2 Kodlama standardı ve kodu temizlemek

Kod, PEP-8 kodlama yönergelerine uymalı ve **black** kod biçimlendiricisi ile biçimlendirilmelidir.

Kod kalitesini denetlemek için **flake8** kullanabilirsiniz. Önerilen eklentiler `.pre-commit-config.yaml` içinde bulunur ve geçerli yapılandırması `setup.cfg` dosyasındadır.

Tüm bunları dayatmak için en kolay yaklaşım `pre-commit` kurmaktır. Depoda, gönderilen dosyaların düzgün olduğunu doğrulayacak bir yapılandırma bulunur. Kurulumdan sonra (`requirements-lint.txt` içinde bulunur) Weblate checkout sırasında `pre-commit install` komutunu çalıştırarak açın. Böylece tüm değişiklikleriniz kendiliğinden denetlenir.

Denetimi el ile de başlatabilirsiniz. Tüm dosyaları denetlemek için şunu yürütün:

```
pre-commit run --all
```

3.4 Weblate hatalarını ayıklamak

Hatalar, uygulama çökmelerine veya çeşitli yanlış davranışlara yol açabilir. Böyle bir sorun ile ilgili bilgi toplayabilir ve [sorun izleyici](#) üzerine gönderebilirsiniz.

3.4.1 Hata ayıklama kipi

Hata ayıklama kipinin açılması, özel durumların tarayıcıda görüntülenmesini sağlar. Bu kip, site arayüzündeki sorunları ayıklamak için yararlıdır. Ancak başarıyı olumsuz etkilediğinden ve kişisel verileri sızdırabileceğinden üretim ortamı için uygun değildir.

Üretim ortamında, hata raporları ile ilgili e-postaları almak için [ADMINS](#) komutunu kullanın ya da bir üçüncü taraf hizmeti kullanarak hata toplamayı yapılandırın.

Ayrıca bakınız:

Hata ayıklama kipini kapatın, Yöneticileri düzgün şekilde yapılandırın, Hata raporlarını derlemek

3.4.2 Weblate günlükleri

Weblate, arka planda neler olup bittiği ile ilgili ayrıntılı günlükler tutabilir. Varsayılan yapılandırmada `syslog` kullanılır ve bu da günlüğün `/var/log/messages` veya `/var/log/syslog` içinde tutulmasını sağlar (`syslog` daemon yapılandırmanıza bağlı olarak).

Celery işlemi (bilgi almak için *Celery ile arka plan görevlerini kullanmak*) genellikle kendi günlüklerini de tutar. Örnek sistem genelindeki kurulumlarda günlükler `/var/log/celery/` altındaki birkaç dosyaya kaydedilir.

Docker kapsayıcıları çıktılarını günlüğe kaydeder (Docker dünyası için her zaman olduğu gibi). Bu günlüklere `docker-compose logs` komutuyla bakabilirsiniz.

Ayrıca bakınız:

Örnek yapılandırma içinde `LOGGING` yapılandırması bulunur.

3.4.3 Arka plan görevleri yapılamıyor

Celery işlemleri tarafından arka planda birçok şey yapılır. E-posta gönderme ya da bileşen kaldırma gibi işlemler yapılmıyorsa, bununla ilgili bir sorun olabilir.

Bu durumda şunlar denetlenmelidir:

- Celery işleminin çalışıp çalışmadığını denetleyin. Bilgi almak için: `ref:celery`
- Celery kuyruğunun durumunu *Yönetim arayüzü* içinden ya da `celery_queues` komutuyla denetleyin
- Hatalar için Celery günlüklerine bakın (bilgi almak için: *Weblate günlükleri*)

3.4.4 Weblate e-postaları alınamıyor

Giden e-postanın düzgün çalışıp çalışmadığını `sendtestemail` yönetim komutuyla (farklı ortamlarda nasıl çağrılacağı ile ilgili bilgi almak için: *Yönetim komutlarını çağırmak*) ya da *Araçlar* sekmesi altındaki *Yönetim arayüzü* komutuyla doğrulayabilirsiniz.

Bunlar e-postaları doğrudan gönderir. Böylece SMTP yapılandırmanızın doğru olduğunu doğrular (bilgi almak için: *Giden e-postayı yapılandırmak*). Bununla birlikte, Weblate üzerinden gelen e-postaların çoğu arka planda gönderilir ve Celery ile ilgili bazı sorunlar da olabilir. Hata ayıklama için *Arka plan görevleri yapılamıyor* bölümüne bakabilirsiniz.

3.4.5 Uygulama çökmelerini incelemek

Uygulamanın çökmesi durumunda, çökme ile ilgili olabildiğince fazla bilgi toplamak yararlıdır. Bunun için, bu tür bilgileri kendiliğinden toplayabilen üçüncü taraf hizmetleri kullanılabilir. Nasıl ayarlanacağı ile ilgili bilgi almak için *Hata raporlarını derlemek*.

3.4.6 Sessiz hatalar

Arka plan işleme için birçok görev Celery üzerine aktarılmıştır. Hatalar kullanıcı arabiriminde görüntülenmez, ancak Celery günlüklerinde görünür. *Hata raporlarını derlemek* komutunu yapılandırmak, bu tür hataları daha kolay fark etmenizi sağlar.

3.4.7 Başarım sorunları

Bazı senaryolarda Weblate başarımı kötü oluyorsa, sorunu gösteren günlük kayıtlarını ve kodun nasıl geliştirilebileceğini anlamamıza yardımcı olabilecek her şeyi toplayın.

Bazı isteklerin herhangi bir belirti olmadan çok uzun sürmesi durumunda, *Hata raporlarını derlemek* ile birlikte `dogslow` kurmak ve hata toplama aracında kesin ve ayrıntılı geri izlemeler almak isteyebilirsiniz.

Yavaş başarımın veri tabanına bağlı olması durumunda, `DEBUG` komutunu etkinleştirdikten sonra aşağıdaki yapılandırmayı kullanarak tüm veri tabanı sorgularının günlüğe kaydedilmesini de sağlayabilirsiniz:

```
LOGGING["loggers"]["django.db.backends"] = {"handlers": ["console"], "level":
↪ "DEBUG" }
```

3.5 Weblate iç yapısı

Not: Bu bölüm size Weblate iç yapısına temel bir genel bakış sunar.

Weblate, kod yapısının çoğu [Django](#) temelinden türetilmiştir.

3.5.1 Klasör yapısı

Weblate ana deposunun klasör yapısına hızlı bir bakış:

docs

Bu belgelerin ‘Sphinks <<https://www.sphinx-doc.org/>>’_ kullanılarak oluşturulabilen kaynak kodu.

dev-docker

Geliştirme sunucusunu çalıştırmak için Docker kodu. Bilgi almak için: *Weblate platformunu Docker üzerinde yerel olarak çalıştırma*.

weblate

Bir [Django](#) uygulaması olarak Weblate kaynak kodu. Bilgi almak için: *Weblate iç yapısı*.

weblate/static

İstemci dosyaları (CSS, JavaScript ve görseller). Bilgi almak için: *Weblate ön yüzü*.

3.5.2 Modüller

Weblate birkaç Django uygulamasından oluşur (bazıları isteğe bağlıdır). Bilgi almak için: *İsteğe bağlı Weblate modülleri*:

accounts

Kullanıcı hesabı, profiller ve bildirimler.

addons

Weblate davranışını değiştiren eklentiler. Bilgi almak için: *Eklentiler*.

api

[Django REST çatısı](#) üzerine geliştirilmiş API.

auth

Kimlik doğrulama ve izinler.

billing

İsteğe bağlı *Faturalama* modülü.

checks

Çeviri dizgesi: ref: *checks* modülü.

fonts

Yazı tipi görüntüleme denetimleri modülü.

formats

Tranlate-toolkit temelli dosya biçimi soyutlama katmanı.

gitexport

İsteğe bağlı *Git dışı aktarıcı* modülü.

lang

	Dil ve çoğul modelleri tanımlayan modül.
<code>legal</code>	İsteğe bağlı <i>Yasal</i> modülü.
<code>machinery</code>	Makine çevirisi hizmetleri bütünleştirmesi.
<code>memory</code>	İç çeviri belleği, bkz: <i>Çeviri belleği</i> .
<code>screenshots</code>	Ekran görüntüsü yönetimi ve optik karakter tanıma (OCR) modülü.
<code>trans</code>	Çevirileri işleyen ana modül.
<code>utils</code>	Çeşitli yardımcı programlar.
<code>vcs</code>	Sürüm denetimi sistemi soyutlaması.
<code>wladmin</code>	Django yönetim arayüzü özelleştirmesi.

3.6 Eklentileri geliştirmek

Weblate üzerinde yerelleştirme iş akışını özelleştirmek için *Eklentiler* kullanılabilir.

```
class weblate.addons.base.BaseAddon (storage=None)
    Weblate eklentileri için temel sınıf.

    classmethod can_install (component, user)
        Eklentinin belirtilen bileşenle uyumlu olup olmadığını denetler.

    configure (settings)
        Yapılandırmayı kaydeder.

    daily (component)
        Kanca günlük olarak tetiklenir.

    classmethod get_add_form (user, component, **kwargs)
        Yeni eklenti eklemek için yapılandırma formunu görüntüler.

    get_settings_form (user, **kwargs)
        Bu eklenti için yapılandırma formunu görüntüler.

    post_add (translation)
        Kanca yeni çeviri eklendikten sonra tetiklenir.

    post_commit (component)
        Kanca değişiklikler depoya işlendikten sonra tetiklenir.

    post_push (component)
        Kanca depo yukarı akışa yollandıktan sonra tetiklenir.
```

post_update (*component, previous_head: str, skip_push: bool*)

Kanca depo yukarı akıştan güncellendikten sonra tetiklenir.

Parametreler

- **previous_head** (*str*) – Güncellemeden önce deponun HEAD bölümü. İlk kopyalamada boş olabilir.
- **skip_push** (*bool*) – Eklenti işleminin değişiklikleri yukarı akışa itmeyi atlayıp atlamayacağı. Genellikle bunu temeldeki yöntemlere `commit_and_push` ya da `commit_pending` olarak iletebilirsiniz.

pre_commit (*translation, author*)

Kanca değişiklikler depoya işlenmeden önce tetiklenir.

pre_push (*component*)

Kanca depo yukarı akışa yollanmadan önce tetiklenir.

pre_update (*component*)

Kanca depo yukarı akıştan güncellenmeden önce tetiklenir.

save_state ()

Eklenti durum bilgilerini kaydeder.

store_post_load (*translation, store*)

Kanca bir dosya işlendikten sonra tetiklenir.

Parametre olarak bir dosya biçimi sınıfının bir kopyasını alır.

Bu seçenek, dosyanın nasıl kaydedileceğini ayarlamak gibi dosya biçimi sınıfı parametrelerini değiştirmek için faydalıdır.

unit_pre_create (*unit*)

Kanca yeni birim oluşturulmadan önce tetiklenir.

İşte örnek bir eklenti:

```
# 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 Weblate ön yüzü

Ön yüz şu anda Bootstrap, jQuery ve birkaç üçüncü taraf kitaplığı kullanılarak oluşturulmuştur.

3.7.1 Desteklenen tarayıcılar

Weblate, tüm bilinen tarayıcıların ve platformların en yeni ve kararlı sürümlerini destekler.

Güncel WebKit, Blink ya da Gecko sürümünü kullanan alternatif tarayıcılar, doğrudan ya da platformun site görünümü API uygulaması üzerinden açıkça desteklenmez. Bununla birlikte, Weblate (çoğu durumda) bu tarayıcılarda da düzgün bir şekilde görüntülenmeli ve çalışmalıdır.

Eski tarayıcılar çalışabilir, ancak bazı özellikler sınırlı olabilir.

3.7.2 Bağımlılık yönetimi

Üçüncü taraf kitaplıklarını güncellemek için Yarn paket yöneticisi kullanılır. Yapılandırma `scripts/yarn` içinde dir ve kitaplıkları yükseltmek, bunları oluşturmak ve üçüncü taraf ön yüz kodunun bulunduğu `weblate/static/vendor` içindeki konumlara kopyalamak için `scripts/yarn-update` sarmalayıcı betiği kullanılır. Weblate kodu doğrudan `weblate/static` içine yerleştirilmeli ya da belirli alt klasörler bulunmalıdır (`weblate/static/editor` gibi).

Yeni üçüncü taraf kitaplığı eklemek genellikle şu şekildedir:

```
# 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 Kodlama biçemi

Weblate, hem JavaScript hem de CSS dosyalarının kod biçimlendirmesi için [Prettier](#) kullanır.

Ek olarak JavaScript kodunu denetlemek için [ESLint](#) kullanıyoruz.

3.7.4 Yerelleştirme

Ön yüz kodunda kullanıcı tarafından görülebilecek herhangi bir metne gerek duyuyorsanız, bu yerelleştirilebilir olmalıdır. Çoğu durumda yalnızca, metninizi `gettext` işlevinin içine kaydırmanız yeterlidir. Ancak kullanılabilecek daha karmaşık özellikler de vardır:

```
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'
```

Ayrıca bakınız:

[Django belgelerinde çeviri başlığı](#)

3.7.5 Simgeler

Weblate şu anda material design simgelerini kullanıyor. Yeni simgeler arıyorsanız, [Material Design Icons](#) ya da [Material Design Resources](#) seçeneğini işaretleyin.

Ek olarak, SVG boyutunu küçültmek için `scripts/optimize-svg` kullanılır. Çünkü simgelerin çoğu yolların biçimine izin vermek için HTML içine gömülüdür.

3.8 Weblate sorunlarını bildirmek

Weblate [hata izleyicisi](#) GitHub üzerinde bulunur.

Karşılaştığınız sorunları bildirmeye ya da Weblate için iyileştirme önerilerinde bulunmaktan çekinmeyen. Sorun bildirimi için size kolaylık sağlayacak çeşitli kalıplar hazırlanmıştır.

Bulduğunuz şey bir Weblate güvenlik sorunuysa, lütfen aşağıdaki [Güvenlik sorunları](#) bölümüne bakın.

Hata bildiriminiz ya da özellik isteğiniz ile ilgili emin değilseniz, [Weblate tartışmaları](#) bölümünü deneyebilirsiniz.

3.8.1 Güvenlik sorunları

Topluluğa yanıt vermesi ve yükseltmesi için zaman tanımak amacıyla, tüm güvenlik sorunlarını gizlice bildirmeniz önemle önerilir. Güvenlik sorunlarını ele almak için HackerOne kullanılır ve doğrudan [HackerOne](#) adresinden bildirilebilir. Oraya gönderdikten sonra, topluluğun olayı çözmek için sınırlı ama yeterli zamanı vardır.

Alternatif olarak, HackerOne üzerine aktarılan security@weblate.org adresine bildirebilirsiniz.

Herhangi bir nedenle HackerOne kullanmak istemiyorsanız, bildirimi michal@weblate.org adresine e-posta olarak gönderebilirsiniz. İletiyi '3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D' PGP anahtarıyla şifrelemeyi seçebilirsiniz. PGP anahtarını [Keybase](#) üzerinden de alabilirsiniz.

Not: Weblate, birçok şey için üçüncü taraf bileşenleri kullanır. Genel olarak bu bileşenlerden birini etkileyen bir güvenlik açığı bulursanız, lütfen bunu doğrudan ilgili projeye bildirin.

Bunların bazıları şunlardır:

- [Django](#)
- [Django REST çatısı](#)
- [Python Social Auth](#)

3.9 Weblate sınaama paketi ve sürekli bütünleştirme

Sınama paketleri geçerli kodun çoğu için bulunabilir. Herhangi bir yeni özellik için sınama çalışmaları ekleyerek kapsamı artırın ve çalıştığını doğrulayın.

3.9.1 Sürekli bütünleştirme

Güncel sınama sonuçları [GitHub işlemleri](#) bölümünde bulunabilir ve kapsama alanı [Codecov](#) bölümünde raporlanır.

Farklı yönleri doğrulayacak birkaç görev vardır:

- Birim sınamaları
- Belge oluşturma ve dış bağlantılar
- Desteklenen tüm sürümlerden aktarım sınaması
- Kod standardı
- Kurulum doğrulaması (oluşturulan dist dosyalarının hiçbir şeyi kaçırmamasını ve sınanabilmesini sağlar)

CI yapılandırması `.github/workflows` klasöründe bulunur ve yoğun olarak `ci` klasöründeki yardımcı betikleri kullanır. Betikler el ile de yürütülebilir, ancak çoğunlukla kullanılacak Django ayarları dosyasını ve veri tabanı bağlantısını tanımlayan birkaç ortam değişkeni gereklidir. Bunun örnek tanımını `scripts/test-database.sh` içinde bulabilirsiniz:

```
# 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
```

Basit yürütme şöyle görünebilir:

```
. scripts/test-database.sh
./ci/run-migrate
./ci/run-test
./ci/run-docs
```

3.9.2 Yerel sınama

Bir sınama paketini yerel olarak çalıştırmak için şunu kullanın:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

İpucu: Sınama kullanılacak bir veri tabanı (PostgreSQL) sunucusuna gerek duyacaksınız. Django varsayılan olarak, `test_` ön eki olan sınamaları çalıştırmak için ayrı bir veri tabanı oluşturur. Bu nedenle ayarlarınızın `weblate` kullanacak şekilde yapılandırılması durumunda, sınamalar `test_weblate` veri tabanını kullanır. Kurulum önerileri için [Weblate için veri tabanı kurulumu](#) bölümüne bakabilirsiniz.

`weblate/settings_test.py` CI ortamında da kullanılır (bilgi almak için: [Sürekli bütünleştirme](#)) ve ortam değişkenleri kullanılarak ayarlanabilir:

```
# 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
```

Sinamaları çalıştırmadan önce, bazı sinamalar için var olmaları gerektiğinden durağan dosyaları toplamalısınız:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

Çalıştırılacak sinamaları tek tek de belirtebilirsiniz:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.gitexport
```

İpucu: Sinamalar geliştirici docker kapsayıcısı içinde de yürütülebilir. Bilgi almak için: [Weblate platformunu Docker üzerinde yerel olarak çalıştırma](#).

Ayrıca bakınız:

Django sinamalarını çalıştırmak ve yazmak ile ilgili bilgi almak için [Testing in Django](#) bölümüne bakabilirsiniz.

3.10 Veri şemaları

Weblate, dış JSON dosyalarının düzenini tanımlamak için [JSON şeması](#) kullanır.

3.10.1 Weblate çeviri belleği şeması

https://weblate.org/schemas/weblate-memory.schema.json	
tür	dizi
öğeler	Çeviri belleği ögesi
tür	nesne
özellikler	
• category	Dizge kategorisi
	1 genel, 2 paylaşılmış, 10000000+ projeye özel, 20000000+ kullanıcıya özel
tür	integer
examples	1
minimum	0
default	1
• origin	Dizge kaynağı
	Dosya adı veya bileşen adı

[sonraki sayfaya devam](#)

Tablo 1 – önceki sayfadan devam

	tür	<i>string</i>
	examples	test.tmx project/component
• source	default	
	<i>Kaynak dizge</i>	
	tür	<i>string</i>
	examples	Merhaba
• source_language	minLength	1
	default	
	<i>Kaynak dil</i>	
	ISO 639-1 / ISO 639-2 / IETF BCP 47	
• target	tür	<i>string</i>
	examples	en
	pattern	^[^]+\$
	default	
• target_language	<i>Hedef dizge</i>	
	tür	<i>string</i>
	examples	Ahoy
	minLength	1
• target_language	default	
	<i>Hedef dil</i>	
	ISO 639-1 / ISO 639-2 / IETF BCP 47	
	tür	<i>string</i>
	examples	tr
	pattern	^[^]+\$
	default	
	additionalProperties	False
definitions		

Ayrıca bakınız:*Çeviri belleği, dump_memory, import_memory***3.10.2 Weblate kullanıcı verilerini dışa aktarma**

https://weblate.org/schemas/weblate-userdata.schema.json		
tür	<i>nesne</i>	
özellikler		
• basic	<i>Temel</i>	
	tür	<i>nesne</i>
	özellikler	
• username	<i>Kullanıcı adı</i>	
	tür	<i>string</i>
	examples	admin
	default	
• full_name	<i>Ad soyad</i>	
	tür	<i>string</i>
	examples	Weblate Yöneticisi
	default	
• email	<i>E-posta</i>	
	tür	<i>string</i>
	examples	noreply@ornek.com
	default	
	format	email

sonraki sayfaya devam

Tablo 2 – önceki sayfadan devam

• profile	• date_joined	Katılma tarihi	
		tür	string
		examples	2019-11-18T18:53:54.862Z
		default	
		format	date-time
	additionalProperties	False	
	Profil		
	tür	nesne	
	özellikler		
	• language	Dil	
		tür	string
		examples	tr
		pattern	^[^]*\$
		default	
	• suggested	Önerilmiş dizge sayısı	
		tür	integer
		examples	1
		default	0
	• translated	Çevrilmiş dizge sayısı	
		tür	integer
		examples	24
		default	0
	• uploaded	Yüklenmiş ekran görüntüsü sayısı	
		tür	integer
		examples	1
		default	0
	• hi-de-completed	Tamamlanmış çevirileri panoda gizler	
		tür	boolean
		examples	False
		default	True
	• secondary_in_zen	Zen kipinde ikincil çevirileri görüntüler	
		tür	boolean
		examples	True
		default	True
	• hi-de_source_sec	İkincil çeviri varsa kaynağı gizler	
		tür	boolean
		examples	False
		default	True
	• editor_link	Düzenleyici bağlantısı	
		tür	string
		examples	
		pattern	^.*\$
		default	
	• translate_mode	Çeviri düzenleyici kipi	
		tür	integer
		examples	0
		default	0
	• zen_mode	Zen düzenleyici kipi	
		tür	integer
		examples	0
		default	0
	• special_chars	Özel karakterler	
		tür	string
		examples	
		pattern	^.*\$

sonraki sayfaya devam

Tablo 2 – önceki sayfadan devam

• auditlog	• dashboard_view	default		
		<i>Varsayılan pano görünümü</i>		
		tür	<i>integer</i>	
		examples	1	
		default	0	
	• dashboard_components	<i>Varsayılan bileşen listesi</i>		
		default	null	
		anyOf	tür	<i>null</i>
			tür	<i>integer</i>
	• languages	<i>Çevrilmiş diller</i>		
		tür	<i>dizi</i>	
		default		
		öğeler	<i>Dil kodu</i>	
			tür	<i>string</i>
		examples	tr	
		pattern	^.*\$	
		default		
	• secondary_languages	<i>İkincil diller</i>		
		tür	<i>dizi</i>	
		default		
		öğeler	<i>Dil kodu</i>	
			tür	<i>string</i>
		examples	sk	
		pattern	^.*\$	
		default		
	• watched	<i>İzlenen projeler</i>		
		tür	<i>dizi</i>	
		default		
		öğeler	<i>Proje kısaltması</i>	
			tür	<i>string</i>
		examples	weblate	
		pattern	^.*\$	
		default		
	• auditlog	additionalProperties	False	
		<i>Denetim günlüğü</i>		
		tür	<i>dizi</i>	
		default		
		öğeler	<i>Öğeler</i>	
			tür	<i>nesne</i>
		özellikler		
		• address	<i>IP adresi</i>	
			tür	<i>string</i>
		examples	127.0.0.1	
		pattern	^.*\$	
		default		
		• user_agent	<i>Kullanıcı uygulaması</i>	
			tür	<i>string</i>
		examples	PC / Linux / Firefox 70.0	
		pattern	^.*\$	
		default		
		• timestamp	<i>Zaman damgası</i>	
			tür	<i>string</i>
		examples	2019-11-18T18:58:30.845Z	

sonraki sayfaya devam

Tablo 2 – önceki sayfadan devam

• activity	default	
	format	date-time
	Etkinlik	
	tür	string
	examples	oturum açma
	pattern	^.*\$
	default	
additionalProperties	False	
definitions		

Ayrıca bakınız:*Kullanıcı profili, dumpuserdata*

3.11 Weblate yayınları

3.11.1 Yayın zamanlaması

Weblate, sürümler, iki aylık aralıklarla yayınlanır (x.y). Bunları genellikle gözden kaçan sorunları düzeltmek için bir dizi hata düzeltme sürümü izler (x.y.z).

Ana sürümün değişmesi, yükseltme işleminin bu sürümü atlayamayacağını gösterir. Daha yüksek x.y sürümlerine yükseltmeden önce her zaman x.0 sürümüne yükseltilmesi gerekir.

Ayrıca bakınız:*Weblate yazılımını güncellemek*

3.11.2 Yayın planlaması

Yakında yayınlanacak sürümlerin özellikleri GitHub kilometre taşları kullanılarak derlenir. Yol haritamızı <<https://github.com/WeblateOrg/weblate/milestones>> adresinde görebilirsiniz.

3.11.3 Yayınlama süreci

Yayınlanma öncesi denetlenmesi gerekenler:

1. Yeni çevrilen dilleri `./scripts/list-translated-languages` ile denetleyin.
2. Son sürümü `:command: ./scripts/prepare-release` ile ayarlayın.
3. Ekran görüntülerinin güncel olduğundan emin olun `:command: make -j 12 -C docs update-screenshots`.
4. Bekliyor olabilecek çevirileri birleştirin `wlc push; git remote update; git merge origin/weblate`

Yayını gerçekleştirin:

5. Yayını oluşturun `./scripts/create-release --tag` (gereksinimler için aşağı bakın).

Yayın sonrası el ile yapılacak adımlar:

6. Docker kalıbını güncelleyin.
7. GitHub kilometre taşını kapatın.
8. Docker kalıbı sılandıktan sonra bir etiket ekleyip itin.
9. Helm çizelgesini yeni sürüme güncelleyin.

10. Birleştirme sınavasında işlenmesi için `.github/workflows/migrations.yml` içine yeni sürümü ekleyin.
11. Site indirme bağlantılarındaki sürümü artırın.
12. Depodaki sürümü `./scripts/set-version` ile artırın.
13. Tüm readthedocs.org çevirilerinin `./scripts/rtd-projects` ile oluşturulup oluşturulmadığını denetleyin.

To create tags using the `./scripts/create-release` script you will need following:

- Yayını imzalamak için kişisel anahtar ile GnuPG
- Weblate Git depolarına itme erişimi (etiketleri iter)
- Yapılandırılmış **hub** aracı ve Weblate deposunda sürüm oluşturma erişimi
- Weblate indirme sunucusuna SSH erişimi (site indirmeleri oraya kopyalanır)

3.12 Güvenlik ve gizlilik

Tüyo: Weblate üzerinde güvenlik, kullanıcılarımızın gizliliğine önem veren bir ortam sağlar.

Weblate geliştirme çalışmaları, [Linux Vakfı Temel Altyapı Girişiminin İyi Örneklerini](#) izlemektedir.

Ayrıca bakınız:

Güvenlik sorunları

3.12.1 Güvenlik güncellemeleri

Yalnızca son sürümün güvenlik güncellemelerini alacağı garanti edilir.

3.12.2 Güvenlik açıkları için bağımlılıkları izlemek

Bağımlılıklarımızdaki güvenlik sorunları [Dependabot](#) ile izlenir. Python ve JavaScript kitaplıkları kapsanır ve son kararlı sürümde, güvenlik açıklarını önlemek için bağımlılıklar güncellenir.

İpucu: Üçüncü taraf kitaplıklarında Weblate üzerinde etkisi olmayan güvenlik açıkları olabilir. Bu nedenle Weblate hata düzeltme sürümleri yayınlanmaz ve bunlar giderilmez.

3.12.3 Docker kapsayıcısı güvenliği

Docker kapsayıcıları [Anchore](#) ve [Trivy](#) güvenlik tarayıcıları ile düzenli olarak taranır.

Böylece, güvenlik açıklarını erken bulmamız ve iyileştirmeleri hızlı bir şekilde yayınlamamız sağlanır.

Bu taramaların sonuçlarını GitHub üzerinden alabilirsiniz. Bunlar CI üzerinde SARIF (Statik Analiz Sonuçları Değişim Biçimi) yapıları biçiminde tutulur.

Ayrıca bakınız:

Sürekli bütünleştirme

3.13 Weblate modüllerine katkıda bulunmak

Ana deponun yanında, Weblate birkaç Python modülünden oluşur. Bütün bunlar aynı yapıyı izler ve bu belgelerde tümü kapsanır.

Örneğin, bu şunları kapsar:

- `wlc`, Python istemci kitaplığı, *Weblate istemcisi* bölümüne bakın
- `translation-finder`, depodaki çevrilebilir dosyaları keşfetmek için kullanılır
- `language-data`, Weblate dil tanımları, *Dil tanımları* bölümüne bakın

3.13.1 Kodlama rehberi

Weblate için geliştirilen kodlar *Tasarım ilkelerine uygun güvenlik* göz önünde bulundurularak yazılmalıdır.

Tüm kodlar, kodun davranışını açıklayan belgelerle birlikte iletilmelidir. Yöntemler, karmaşık kod blokları veya kullanıcı tarafından görülebilen özellikler belgelenmeyi unutmayın.

Herhangi bir yeni kod **PEP 484** türünde ipuçları kullanılmalıdır. Var olan kod henüz bunları içermediğinden bunu henüz CI ile denetlemiyoruz.

3.13.2 Sınamaları yürütmek

Sınamalar `py.test` kullanılarak yürütülür. Öncelikle sınama gereksinimlerini kurmanız gerekir:

```
pip install -r requirements-test.txt
```

Daha sonra depo checkout sayfasında sınama paketini yürütebilirsiniz:

```
py.test
```

Ayrıca bakınız:

CI bütünleştirmesi *Weblate sınama paketi ve sürekli bütünleştirme* ile çok benzer.

3.13.3 Kodlama standardı ve kodu temizlemek

Kod, PEP-8 kodlama yönergelerine uymalı ve **black** kod biçimlendiricisi ile biçimlendirilmelidir.

Kod kalitesini denetlemek için **flake8** kullanabilirsiniz. Önerilen eklentiler `.pre-commit-config.yaml` içinde bulunur ve geçerli yapılandırması `setup.cfg` dosyasındadır.

Tüm bunları dayatmak için en kolay yaklaşım `pre-commit` kurmaktır. Depoda, gönderilen dosyaların düzgün olduğunu doğrulayacak bir yapılandırma bulunur. Kurulumdan sonra (`requirements-lint.txt` içinde bulunur) Weblate checkout sırasında `pre-commit install` komutunu çalıştırarak açın. Böylece tüm değişiklikleriniz kendiliğinden denetlenir.

Denetimi el ile de başlatabilirsiniz. Tüm dosyaları denetlemek için şunu yürütün:

```
pre-commit run --all
```

Ayrıca bakınız:

Weblate kaynak kodları

3.14 Weblate hakkında

3.14.1 Proje hedefleri

Çevirmenlerin katkıda bulunmasını kolaylaştıran, geniş bir *Desteklenen dosya biçimleri* yelpazesini destekleyen sıkı *Sürüm denetimi bütünleştirilmesi* bütünleşmesi ile internet temelli sürekli yerelleştirme aracı.

3.14.2 Proje adı

“Weblate”, “web” ve “translate” sözcüklerinin birleşimidir.

3.14.3 Proje sitesi

Açılış sayfası <https://weblate.org> ve bulut barındırma hizmeti <https://hosted.weblate.org> adresinde bulunur. Belgeler <https://docs.weblate.org> adresinde görülebilir.

3.14.4 Proje logoları

Proje logoları ve diğer çizelgeler <https://github.com/WeblateOrg/graphics> adresinde bulunabilir.

3.14.5 Liderlik

Bu proje, michal@cihar.com adresinden ulaşılabilen Michal Čihař tarafından geliştiriliyor.

3.14.6 Katkıda bulunanlar

Weblate, Michal Čihař tarafından kuruldu. 2012 yılında kurulduğundan bu yana binlerce kişi katkıda bulundu.

3.15 Lisans

Daha ayrıntılı lisans bilgileri Weblate kaynak kodunda bulunabilir ve [REUSE 3.0 teknik özellikleri](#) yönergelerini izler.

Telif Hakkı © Michal Čihař michal@cihar.com

Bu yazılım özgürdür: Yeniden dağıtılabilir ve/veya Özgür Yazılım Vakfı tarafından yayınlanan GNU Genel Kamu Lisansı koşulları altında, lisansın 3. sürümü ya da (seçiminize göre) daha sonraki bir sürümü koşullarına uygun olarak değiştirilebilirsiniz.

Bu yazılım, yararlı olacağı umuduyla dağıtılır. Ancak HERHANGİ BİR GARANTİ VERİLMEZ; TİCARETE UYGUNLUK VEYA BELİRLİ BİR AMACA UYGUNLUK GARANTİSİ VERİLMEZ. Ayrıntılı bilgi almak için GNU Genel Kamu Lisansı koşullarına bakabilirsiniz.

Bu yazılımla birlikte GNU Genel Kamu Lisansı metninin bir kopyasını almış olmalısınız. Bulamıyorsanız <https://www.gnu.org/licenses/> adresine bakabilirsiniz.

4.1 Weblate 4.17

Released on April 17th 2023.

- Raporların dile göre süzülmesi sağlandı.
- Kullanımdan kaldırılan `cleanup_celery` komutu kaldırıldı.
- Bazı ekiplerin yaşadığı özel proje görünürlüğü sorunları düzeltildi.
- Kendiliğinden çevirinin diğer bileşenlerden çeviri yaparken hedef durumunu dikkate alması sağlandı.
- Herkese açık kullanıcı profillerinin başarımı iyileştirildi.
- Sentry bütünleşmesi başarımı iyileştirildi.
- Ed25519 SSH anahtarları desteği eklendi.
- Ölçümlerin depolanması yeniden yazıldı.
- Dizgeleri konuma göre arama özelliği eklendi.
- Belgelendirmede iyileştirmeler yapıldı.
- Çeviri değiştirilmemiş denetiminin sözlükte çevrilemez olarak imlenmiş terimlere uygun davranması sağlandı.
- Kendiliğinden Devanagari danda düzeltme özelliği eklendi.
- Her dilin proje çeviri belleğini indirme özelliği eklendi.
- Kullanılmayan bileşenler için yeni uyarı eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.2 Weblate 4.16.4

16 Mart 2023 tarihinde yayınlandı.

- Bağımlılıklar güncellendi.
- Arka plan görevlerinin zamanlaması geliştirildi.

Ayrıntılı olarak tüm değişiklikler.

4.3 Weblate 4.16.3

15 Mart 2023 tarihinde yayınlandı.

- Proje yedeklemelerinde oturum işlenmesi iyileştirildi.
- Bağımlılıklar güncellendi.
- Yerelleştirme güncelleştirmeleri yapıldı.
- Belgelendirmede iyileştirmeler yapıldı.

Ayrıntılı olarak tüm değişiklikler.

4.4 Weblate 4.16.2

8 Mart 2023 tarihinde yayınlandı.

- Çeviri belleğinde arama sorunu düzeltildi.
- Kendiliğinden çeviri daha fazla hizmet ile düzeltildi.
- Örtüşen sözlük terim eşleşmelerinin sunumu geliştirildi.
- Bazı biçimlerde İngilizce olmayan kaynak dil için çoğul ayrıştırması düzeltildi.
- go-i18n v2 JSON dosyaları desteği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.5 Weblate 4.16.1

1 Mart 2023 tarihinde yayınlandı.

- Sınama paketi sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.6 Weblate 4.16

1 Mart 2023 tarihinde yayınlandı.

- Biçim dizesi denetimlerinin yinelenen biçimleri de algılaması sağlandı.
- Bazı özel biçimlendirilmiş dizgelerin arama başarımı iyileştirildi.
- Celery görev zamanlamasının kayıt yeri veri tabanı olarak değiştirildi.
- IBM Watson Language Translator desteği eklendi.
- 4.14 sürümünde kullanımdan kaldırılan sürüm denetimi sistemi bütünleştirmesi desteği kaldırıldı.

- Bitbucket sunucusu çekme istekleri desteği eklendi.
- Gettext PO dosyalarında çakışmaların işlenmesi iyileştirildi.
- API ile ekleme sırasında dizge durumunu tanımlama desteği eklendi.
- CORS tarafından izin verilen kaynakları yapılandırma desteği eklendi.
- Çoğul biçimler için kendiliğinden öneri desteği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.7 Weblate 4.15.2

25 Ocak 2023 tarihinde yayınlandı.

- Varsayılan yapılandırmada gotext JSON ve i18next v4 biçimleri etkinleştirildi.
- Bozuk dosyalar yüklendiğinde yaşanan çökme sorunu düzeltildi.
- Git depo durumunda eski klasörlerin görüntülenmesi sağlandı.

Ayrıntılı olarak tüm değişiklikler.

4.8 Weblate 4.15.1

19 Ocak 2023 tarihinde yayınlandı.

- Kendiliğinden çeviri önerileri düzeltildi.
- Bazı uç durumlarda yaşanan eklenti sayfasının çökmesi sorunu düzeltildi.
- Bazı durumlarda yeni çevirilerde yaşanan kalıbın çevrilmemesi sorunu düzeltildi.
- **REUSE 3.0** kullanılarak lisanslama belgelendirildi.
- Takım yönetiminde kullanıcıların sayfalandırması düzeltildi.
- Proje oluşturma ve kaydetme başarımı iyileştirildi.
- gotext JSON dosyaları desteği eklendi.
- i18next v4 dosyaları desteği eklendi.
- API üzerinde sayfalandırma artık özelleştirilebilir.

Ayrıntılı olarak tüm değişiklikler.

4.9 Weblate 4.15

16 Aralık 2022 tarihinde yayınlandı.

- Tek bir dizge için değişikliklere göz atma özelliği eklendi.
- Diğer bileşenlerden kendiliğinden çeviri yapılırken çoğulların işlenmesi düzeltildi.
- Dizgeyi öneri olarak göndermek için Alt+Enter klavye kısayolu eklendi.
- Fluent biçiminde yerleştirilebilir ögeler desteği eklendi.
- Çeviri belleğinin başarımı iyileştirildi.
- İyi bilinen kod barındırma hizmetleri için repoweb tarama bağlantılarının kendiliğinden oluşturulması sağlandı.
- Birkaç görünümün başarımı iyileştirildi.

- Çoğul biçimleri olan dizgelerin listelenmesi iyileştirildi.
- HTML head bölümüne markup kodu eklenmesi sağlandı.
- Eklentide, MO dosyaları oluşturulurken yalnızca çevrilen dosyaların katılması sağlandı.
- Kurallı ifade imlerinin işlenmesi düzeltildi.
- Çoğul biçimlerde yer belirtici denetimi davranışı iyileştirildi.
- Çeviri dosyalarının Google Play ile uyumlu biçimde adlandırılması özelliği eklendi.
- API üzerine etiket desteği eklendi.
- İşlemeler için bildirimlerden farklı e-posta seçme özelliği eklendi.
- Docker kalıbı artık hata ayıklama kipini varsayılan olarak etkinleştirmiyor.
- Sözlük terimleri sözlük bileşeni önceliğine göre sıralandı.
- Takım üyeleri ekleyebilen ya da kaldırabilen takım yöneticileri eklendi.
- Kullanıcıları silmeden önce bir açılır pencere onayı eklendi.
- XML çıktısını özelleştirmek için eklenti eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.10 Weblate 4.14.2

5 Kasım 2022 tarihinde yayınlandı.

- Çeviri belleğinden kayıtları kaldırma özelliği eklendi.
- Çift dil uyarısında çözümleme iyileştirildi.
- Ard arda yinelenen sözcük denetiminin doğruluğu iyileştirildi.
- Çok sayıda bildirim gönderme ölçeklendirilmesi iyileştirildi.
- Alt yazı çevirisi için dizge durumunun işlenmesi iyileştirildi.
- VCS hizmeti API anahtarlarının _CREDENTIALS listesi yerine _TOKEN/_USERNAME yapılandırması ile güvenli olmayan yapılandırması kullanımdan kaldırıldı.
- Yüklenen bazı CSV dosyalarının işlenmesi düzeltildi.
- Fark görünümünde boşluk değişimlerinin işlenmesi iyileştirildi.
- Yönetim sayfalarına kendiliğinden öneri yönetimi bağlantısı eklendi.
- Geçmiş yorumların kaldırılması/çözümlemesi için izleme eklendi.
- İlişkili bileşenlerle proje yedeklerinin geri yüklenmesi düzeltildi.
- Hesap açılmadığında captcha girişi düzeltildi.
- DeepL dil desteği iyileştirildi.
- Kimliği doğrulanan depolarla internet kancaları uyumluluğu iyileştirildi.
- Python 3.11 desteği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.11 Weblate 4.14.1

15 Eylül 2022 tarihinde yayınlandı.

- Bazı durumlarda yaşanan proje yedeklerinin oluşturulması sorunu düzeltildi.
- Dosya yüklemede hata bildirimi iyileştirildi.
- Kimlik doğrulama yapılırken GitHub üzerinden e-postaları doğrulanmış tüm kullanıcıların alınması sağlandı.
- Sözlük terimlerinin bağlam ya da anahtarlarla eşleştirmesi engellendi.
- Dizge kaldırma bildirimleri eklendi.
- Sözlükte çevrilemeyen terimlerin yönetimi iyileştirildi.
- Takım yönetimi sayfasında takım üyelerinin sayısı listelendi.
- Grup yönetimi arayüzü eklendi.
- Onaylama kullanıldığında onaylama istatistiklerinin her zaman görüntülenmesi sağlandı.
- Birim API uygulamasına arama özelliği eklendi.
- Onaylama iş akışında, salt okunur dizgeler için ilerleme çubuğunun görüntülenme sorunu düzeltildi.
- Birmanca için noktalama denetimleri iyileştirildi.
- Ölçüm verilerinin çöp biriktirmesi düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.12 Weblate 4.14

22 Ağustos 2022 tarihinde yayınlandı.

- Geçmişteki eklenti değişikliklerinin izlenmesi sağlandı.
- Windows RC, HTML ve metin dosyalarından yaşanan çeviri işleme sorunu düzeltildi.
- Dil kodu biçemi yapılandırma seçenekleri genişletildi.
- Son CLDR sürümlerinde güncellenen çoğul biçimler desteği eklendi.
- Çok sayıda çeviri içeren bileşenler güncellenirken bellek kullanımı azaltıldı.
- SAP Translation Hub çeviri etki alanı desteği eklendi.
- Kaynak dizge konumlarında mutlak bağlantıların kullanılabilmesi sağlandı.
- Bazı ters vekil sunucuların arkasında çalışma iyileştirildi.
- API, çeviri belleğini kapsayacak şekilde genişletildi.
- Belge çevirisi iş akışı iyileştirildi.
- HTML ve metin dosyaları çevirisinin güvenilirliği artırıldı.
- Proje düzeyinde yedekleme özelliği eklendi.
- Çeviri belleği aramalarının başarımı ve bellek kullanımı iyileştirildi.

Ayrıntılı olarak tüm değişiklikler.

4.13 Weblate 4.13.1

1 Temmuz 2022 tarihinde yayınlandı.

- Geçmiş kayıtlarında yaşanan önerilerin izlenmesi sorunu düzeltildi.
- Cloudflare üzerinden yaşanan ters vekil sunucu bilgisini işleme sorunu düzeltildi.
- Ayırıştırma hatasının bir bileşenin çevrilmesini kilitlemesi sağlandı.
- Keşif eklentisinde yaşanan ara dosyayı yapılandırma sorunu düzeltildi.
- Yer belirticilerde DeepL çeviri davranışı düzeltildi.
- API üzerinden dizgelerin çevrilmemesi sorunu düzeltildi.
- API üzerine bir gruptan kullanıcı kaldırma özelliği eklendi.
- Kullanıcı davet e-postalarında yaşanan denetim günlüğü sorunu düzeltildi.
- Java biçimlendirme dizgelerinde yaşanan im adları sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.14 Weblate 4.13

15 Haziran 2022 tarihinde yayınlandı.

- Dil adlarını güncelleme davranışı değiştirildi.
- Proje listesine sayfalandırma eklendi.
- Yeni birimler oluşturma API uygulaması artık yeni oluşturulan birim ile ilgili bilgiler veriyor.
- Bileşen keşfinde bir ara dilin yapılandırılması eklendi.
- CSV biçimlerine sabit kodlama çeşitleri eklendi.
- Temeldeki uygulamaya daha iyi uyması amacıyla bazı biçimler için bağlam ve konumun işlenmesi değiştirildi.
- ResourceDictionary biçimi desteği eklendi.
- Renk körleri için ilerleme çubuğu renkleri iyileştirildi.
- Dizge kaldırılırken yaşanan çeşitlerin temizlenmesi sorunu düzeltildi.
- Django 4.1 uyumluluğu sağlandı.
- XLIFF içinde kaçırılan XML öğelerini depolama özelliği eklendi.
- Yer belirtici denetim hatalarının biçimlendirilmesi iyileştirildi.
- /.well-known/change-password /accounts/password/ adresine yönlendirildi.
- Makine çevirisi hizmetleri artık her proje için ayrı olarak yapılandırılabilir.
- Yorumların çözümlenmesi için ayrı bir izin eklendi ve *Dizgeleri onaylama* rolüne verildi.
- Alternatif çevirileri CSV dosyasında depolama özelliği eklendi.
- Yer belirtici denetimi artık büyük/küçük harfe duyarsız olarak da yapılabilir.

Ayrıntılı olarak tüm değişiklikler.

4.15 Weblate 4.12.2

11 Mayıs 2022 tarihinde yayınlandı.

- Bazı bileşenlerde yaşanan proje çeviri belleğinin yeniden oluşturulması sorunu düzeltildi.
- Bileşenlerin çevrilmemiş dizgelere göre sıralanması sorunu düzeltildi.
- Yeni dil eklenirken yaşanan olası çeviri kaybı sorunu düzeltildi.
- Aktarım işlemleri yapılırken Weblate SSH anahtarının oluşturulduğundan emin olunması sağlandı.

Ayrıntılı olarak tüm değişiklikler.

4.16 Weblate 4.12.1

29 Nisan 2022 tarihinde yayınlandı.

- Çekme isteği ileti başlığı düzeltildi.
- Fluent biçiminde söz dizimi hatalarının işlenmesi iyileştirildi.
- Bildirim e-postalarında yaşanan avatar görüntülenme sorunu düzeltildi.
- İnternet üzerinden para kazanma özelliği eklendi.
- Çeviriler kaldırılırken yaşanan eski kaynak dizgelerinin kaldırılması sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.17 Weblate 4.12

20 Nisan 2022 tarihinde yayınlandı.

- *Nokta işareti farklı* için Amharca desteği eklendi.
- *Soru işareti farklı* için Birmanca desteği eklendi.
- *Sözde yerel oluşturma* eklentisinin seçenekleri genişletildi.
- Bir dizgedeki tüm kalite denetimlerini yok saymak için `ignore-all-checks` imi eklendi.
- *Sözde yerel oluşturma* eklentisinin denetim hatalarını tetiklemesi engellendi.
- *Gitea çekme isteği* özelliği eklendi.
- *Dil kodu biçimi* için Linux biçimi dil kodu eklendi.
- Proje çeviri belleğini yeniden oluşturma özelliği eklendi.
- API bir dosyadan bileşenler oluşturma için iyileştirildi.
- Diğer çevrilere kopyala ve çoğalt düğmeleri eklendi.
- Birleştirme isteği iletisi bileşen düzeyinde yapılandırılabilir kılındı.
- XML etiketlerinde en fazla uzunluk kısıtlama davranışı iyileştirildi.
- Fluent dosyalarının ek yorumlarla yüklenmesi sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.18 Weblate 4.11.2

4 Mart 2022 tarihinde yayınlandı.

- Binary sürümdeki bozuk MO dosyaları sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.19 Weblate 4.11.1

4 Mart 2022 tarihinde yayınlandı.

- Git ve Mercurial ile yaşanan argümanların eksik temizlenmesi sorunu düzeltildi - CVE-2022-23915. Ayrıntılı bilgi almak için [GHSA-3872-f48p-pxqj](#) bölümüne bakabilirsiniz.
- CSV dosyalarından belirsiz dizgelerin yüklenmesi sorunu düzeltildi.
- API üzerinden takım oluşturma özelliği eklendi.
- Kullanıcı anma önerilerinin görüntülenmesi sorunu düzeltildi.
- Proje belirteçlerine erişim artık özelleştirilebilir.

Ayrıntılı olarak tüm değişiklikler.

4.20 Weblate 4.11

25 Şubat 2022 tarihinde yayınlandı.

- XSS - CVE-2022-24710 sorunları düzeltildi. Ayrıntılı bilgi almak için [GHSA-6jp6-9rf9-gc66](#) adresine bakabilirsiniz.
- API kullanarak eklenti kurulumu sorunu düzeltildi.
- *İşlem gereken dizgeler* adı *Tamamlanmamış dizgeler* olarak değiştirildi.
- *ICU MessageFormat söz dizimi* kaynaklı yanlış pozitif sonuç sorunu düzeltildi.
- Diğer kullanımlar listesinde kilit ve katılımcı anlaşmasının görüntülenmesi sağlandı.
- Eski dizgeler ya da eksik çoğullar bulunduran PO dosyalarının güncellenme sorunu düzeltildi.
- Gerrit ile bir araya toplama eklentisi uyumluluğu iyileştirildi.
- Kullanıcı dillerinin *Accept-Language* üst bilgisine göre kendiliğinden hazırlanması sağlandı.
- Dizge kaldırılırken sorunların işlenmesi iyileştirildi.
- Weblate için artık Django 3.2 gerekiyor.
- Proje kodu kimlik doğrulaması ile bazı yazma işlemleri düzeltildi.
- Depodaki dizgeler değiştiğinde dizge durumunun izlenmesi sorunu düzeltildi.
- Depodaki dizge değişikliklerinin izlenmesi sağlandı.
- Gezinmeyi iyileştirmek için çeviriler listesine sabitlenebilen başlık eklendi.
- *Java özellikleri* ile yaşanan çevrilmeyen dizgeler sorunu düzeltildi.
- ASCII olmayan dal adlarıyla Git işlemi sorunu düzeltildi.
- *Çeviriyi kaynak dizgeyle önceden doldurma* eklentisi eklendi.
- *Hızlı ileri almadan birleştir* *Birleştirme biçemi* özelliği eklendi.
- Yeni eklenen dizgelerde *Kendiliğinden çeviri* eklentisinin tetiklenmesi sorunu düzeltildi.

- Birmanca için noktalama denetimleri iyileştirildi.
- Kullanıcılara erişim izni vermek için proje düzeyinde özel takımların tanımlanması sağlandı. Bilgi almak için: [Her proje için erişim denetimi yönetimi](#).
- Uyarılara belge bağlantıları eklendi.
- Docker kapsayıcının, gerektiğinde giden e-posta için TLS/SSL seçeneğini kendiliğinden etkinleştirmesi sağlandı.
- Çözümlemiş yorumları arama özelliği eklendi.
- borgbackup 1.2 desteği eklendi.
- *Kendiliğinden çevrilmiş* etiketinin uygulanması sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.21 Weblate 4.10.1

22 Aralık 2021 tarihinde yayınlandı.

- Django 4.0 yükseltmesi ile gelen değişiklikler belgelendirildi.
- *Kendiliğinden çevrilmiş* etiketinin görüntülenmesi sorunu düzeltildi.
- Paylaşılan bir deposu olan bileşenlerde yaşanan dalın API ile görüntülenmesi sorunu düzeltildi.
- Başarısız olan itme uyarısının çözümlemesi iyileştirildi.
- Değişikliklere göz atarken sayfayı el ile düzenleme sorunu düzeltildi.
- *Kashida harfi kullanılmış* doğruluğu iyileştirildi.
- Weblate Docker kapsayıcısı artık Python 3.10 kullanıyor.

Ayrıntılı olarak tüm değişiklikler.

4.22 Weblate 4.10

16 Aralık 2021 tarihinde yayınlandı.

- DeepL ile düzenlilik ve yer belirtici özelliği eklendi.
- Artık proje ve dil düzeyinde toplu düzenleme ve arama/değiştirme kullanılabilir.
- Arama ve değiştirme için süzme eklendi.
- Düzeltme: “Kendiliğinden çevir” yetkisi artık *Diller* grubunun bir parçası değil.
- “Kendiliğinden çevir”, *Yönetim* ve yeni *Kendiliğinden çeviri* grubuna eklendi.
- Özel karakterlerle XLSX dosyalarının oluşturulması sorunu düzeltildi.
- GitHub kimlik doğrulama yönetim bölümüne, kullanıcının belirli bir GitHub kuruluşunda veya takımında olup olmadığını denetleme özelliği eklendi.
- API üzerine iletilen geçersiz parametrelerle ilgili geri bildirim iyileştirildi.
- API için proje kapsamında erişim kodları özelliği eklendi.
- Bazı durumlarda yaşanan dizge silme sorunu düzeltildi.
- Yeni eklenen dizgelerin çevrilmesi sorunu düzeltildi.
- Süzülmesini kolaylaştırmak için kendiliğinden çevrilmiş dizgeler etiketlendi.

Ayrıntılı olarak tüm değişiklikler.

4.23 Weblate 4.9.1

19 Kasım 2021 tarihinde yayınlandı.

- Kalıp değiştirildikten sonra yaşanan tek dilli dosyaların yüklenmesi sorunu düzeltildi.
- İmlerdeki boşlukların işlenmesi iyileştirildi.
- İndirme API uygulamasına süzme özelliği eklendi.
- Yeni çeviriler eklerken yaşanan istatistik görünümü sorunu düzeltildi.
- GitHub SSH anahtar değişikliği ile ilgili sorunlar çözüldü.

Ayrıntılı olarak tüm değişiklikler.

4.24 Weblate 4.9

10 Kasım 2021 tarihinde yayınlandı.

- Geçmiş etkinliklerin ayrıntı düzeyi artırıldı.
- Geçmişin görüntülenmesi iyileştirildi.
- Çeviri sayfalarının başarımı iyileştirildi.
- Çeviri dosyası indirmelerini kısıtlama özelliği eklendi.
- `safe-html` artık `md-text` ile kullanıldığında Markdown kodlarını anlayabiliyor.
- `max-length` kod imi artık `xml-text` ile kullanıldığında XML kodlamasını yok sayıyor.
- *Çevirinin en fazla uzunluğu* içinde oluşturulan metinlerin boyutları sorunu düzeltildi.
- Yaklaşan Google ilke değişikliklerine yardımcı olmak için uygulama mağazası başlığının uzunluğu 30 karaktere düşürüldü.
- SSH çağrısını `SSH_EXTRA_ARGS` aracılığıyla özelleştirme özelliği eklendi.
- ICU MessageFormat için denetimler eklendi.
- Makine çevirisi arka uçlarında hata durumunun ele alınması iyileştirildi.
- Dizgelerdeki olağan dışı boşluk karakterleri vurgulandı.
- Düzenleme yaparken çevrilen dizgede kalma seçeneği eklendi.
- Borg çağrısını `BORG_EXTRA_ARGS` aracılığıyla özelleştirme özelliği eklendi.
- Tek dilli çevirilerde yaşanan MO dosyalarının oluşturulması sorunu düzeltildi.
- Tüm bileşen çevirilerini ZIP dosyası olarak indirmek için API uç noktası eklendi.
- Python 3.10 desteği eklendi.
- Yönetim arayüzüne e-posta davetini yeniden gönderme özelliği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.25 Weblate 4.8.1

10 Eylül 2021 tarihinde yayınlandı.

- Django yönetici arayüzünde yaşanan kullanıcı silme sorunu düzeltildi.
- Eklenti parametreleri daha ayrıntılı olarak belgelendirildi.
- Sözlükte yaşanan JavaScript sorunu düzeltildi.
- Tutarlılık denetimlerinde eşleşme sayısına sınır eklendi.
- Makine çevirilerinde yer belirtecilerin işlenmesi iyileştirildi.
- API kullanarak eklenti oluşturma sorunu düzeltildi.
- Alt bilgiye gizlilik ilkesi bağlantısı eklemek için `PRIVACY_URL` seçeneği eklendi.
- Üye e-posta adresleri proje yöneticilerinden gizlendi.
- Çakışma durumunda Gettext PO birleştirmesi iyileştirildi.
- Sözlük vurgulaması iyileştirildi.
- XML denetimleriyle `safe-html` iminin davranışı iyileştirildi.
- Bağlantılı bileşenler için işleme iletileri düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.26 Weblate 4.8

21 Ağustos 2021 tarihinde yayınlandı.

- Apple stringsdict biçimine destek eklendi.
- PostgreSQL ile tam arama işlemine büyük/küçük harfe duyarlılık özelliği eklendi.
- Bazı durumlarda yaşanan sözlük açıklamalarının kaydedilmesi sorunu düzeltildi.
- Belgelendirmede iyileştirmeler yapıldı.
- Başarım iyileştirmeleri yapıldı.
- Gerrit ile bir araya toplama eklentisi uyumluluğu iyileştirildi.
- Tek dilli sözlük bileşenlerine dizge ekleme sorunu düzeltildi.
- Çeşitler işlenirken başarım iyileştirildi.
- Bir araya toplama eklentisinde yaşanan yukarı akış değişikliklerinin işlenmesini atlama sorunu düzeltildi.
- İndirmelerin dosya uzantısını korunsun.
- Fluent biçimi desteği eklendi.
- JSON biçimlerini girintilemek için sekmeleri kullanma özelliği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.27 Weblate 4.7.2

15 Temmuz 2021 tarihinde yayınlandı.

- Bir projede daha fazla dil kısaltmasının yapılandırılması özelliği eklendi.
- API üzerinde yaşanan arama dizgesini doğrulama sorunu düzeltildi.
- Bir etki alanı değişikliğinden sonra Git dışı aktarıcı adresleri düzeltildi.
- Windows RC dosyaları için temizleme eklentisi düzeltildi.
- XLIFF güncellemesinde yaşanabilecek çökme sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.28 Weblate 4.7.1

30 Haziran 2021 tarihinde yayınlandı.

- Sözlüğe terim ekleme açılır penceresi iyileştirildi.
- LibreTranslate makine çevirisi hizmeti desteği eklendi.
- Yeni projeler oluşturulurken hız sınırlaması eklendi.
- Dosya güncelleme başarımı iyileştirildi.

Ayrıntılı olarak tüm değişiklikler.

4.29 Weblate 4.7

17 Haziran 2021 tarihinde yayınlandı.

- Yapılandırma sağlığı denetimi iyileştirildi.
- Gettext PO dosyasında kullanılan `object-pascal-format` desteği eklendi. Ayrıntılı bilgi: *Object Pascal biçimi*.
- Amacı daha iyi açıklamak için *Yakındaki dizgeler*, *Benzer dizgeler* olarak yeniden adlandırıldı.
- *mi18n lang dosyaları* desteği eklendi.
- SAML kimlik doğrulama bütünleştirmesi iyileştirildi.
- Uç durumları daha iyi işlemek için *Gerrit* bütünleştirmesi düzeltildi.
- Weblate gereksinimi Django 3.2 olarak güncellendi.
- E-posta kimlik doğrulaması devre dışı bırakıldığında yaşanan kullanıcıları davet etme sorunu düzeltildi.
- Dil tanımları iyileştirildi.
- Kullanıcıların bir projeye katkıda bulunmalarını engelleme özelliği eklendi.
- Sözlük dillerinin kendiliğinden oluşturulması sorunu düzeltildi.
- Eklentilerle ilgili belgeler zenginleştirildi.
- Bağlantılı depoları olan bileşenlerin başarımları iyileştirildi.
- Ücretsiz DeepL API desteği eklendi.
- Kullanıcı yönetimi için artık Django yönetici arayüzüne gerek duyulmuyor.

Ayrıntılı olarak tüm değişiklikler.

4.30 Weblate 4.6.2

8 Mayıs 2021 tarihinde yayınlandı.

- Paylaşılan bileşenin projeler arasında taşınmasından sonra yaşanan çökme sorunu düzeltildi.
- Boş özellikler dosyalarına yeni dizgelerin eklenmesi sorunu düzeltildi.
- Sağdan sola yazılan dillerde kopya simgesinin hizalaması düzeltildi.
- Bilgi sekmesindeki dizge istatistikleri zenginleştirildi.
- Git üzerinde yok sayılan çeviri dosyalarının işlenmesi sorunu düzeltildi.
- Ölçüm başarımı iyileştirildi.
- Sözlükleri kaydederken ortaya çıkabilen sorun düzeltildi.
- Farklı çoğul kuralları olan dillerde tutarlılık denetiminin davranışı düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.31 Weblate 4.6.1

2 Mayıs 2021 tarihinde yayınlandı.

- Eski istenmeyen ileti koruma kodu kaldırıldı.
- Kaynak çoğul denetiminin doğruluğu iyileştirildi.
- Docker kullanıcı arayüzü dilleri listesi güncellendi.
- Çekme istekleri oluşturulurken alınan hata iletileri iyileştirildi.
- Pagure üzerinde çekme istekleri oluşturma sorunu düzeltildi.
- Kurulmuş eklentilerin kendiliğinden tetiklenmesi sorunu düzeltildi.
- Yükseltme yapılırken olası ön belleğe alma sorunları düzeltildi.
- Yükleme yapılırken tek dilli çevirilere yeni birimler eklenmesi sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.32 Weblate 4.6

19 Nisan 2021 tarihinde yayınlandı.

- `auto_translate` yönetim komutuna çeviri kipini belirten bir parametre eklendi.
- *Metin dosyaları* desteği eklendi.
- Tüm nesneler için eğilimler ve ölçümler eklendi.
- İkincil dillerdeki metinleri doğrudan kopyalama özelliği eklendi.
- Değişikliklere göz atarken tarihe göre süzme özelliği eklendi.
- Etkinlik çizelgeleri iyileştirildi.
- İletişim formu e-postalarının göndericisini yapılandırma özelliği eklendi.
- Bileşen oluşturma API uygulamasında parametre doğrulaması iyileştirildi.
- Hız sınırlaması artık süper kullanıcılara uygulanmıyor.
- Kendiliğinden çeviri eklentisinin başarımı ve güvenilirliği iyileştirildi.

- Docker kapsayıcısına hız sınırlamasını özelleştirme özelliği eklendi.
- Bileşen oluşturma API uygulamasına kendiliğinden *İç Weblate adresleri* kullanma özelliği eklendi.
- Dizge listesinde durumun gösterilmesi sadeleştirildi.
- Parola karması varsayılan olarak Argon2 kullanacak biçimde güncellendi.
- Çeviri durumunu gösteren ilerleme çubukları sadeleştirildi.
- Amacı netleştirmek için *Eksik dilleri ekleme* adı değiştirildi.
- Dizge durumunu XLIFF olarak kaydetme sorunu düzeltildi.
- Dil genelinde arama özelliği eklendi.
- Docker dağıtımında ilk kez *Yatay ölçeklendirme* desteği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.33 Weblate 4.5.3

1 Nisan 2021 tarihinde yayınlandı.

- Ölçümlerin toplanması sorunu düzeltildi.
- Dizge eklenirken yaşanabilecek çökme sorunu düzeltildi.
- Arama sorgusu örnekleri iyileştirildi.
- Değiştirme yüklemesinde yaşanması olası yeni eklenen dizgelerin kaybolması sorunu düzeltildi.

4.34 Weblate 4.5.2

26 Mart 2021 tarihinde yayınlandı.

- Kendiliğinden çeviri için zamanlama yapılandırması eklendi.
- Lua biçim denetimi eklendi.
- *Art arda yinelenen sözcükler var* denetiminde biçim dizgeleri yok sayıldı.
- Çeviri sayfasından ekran görüntüsü yüklenmesi sağlandı.
- Depo bakımına zorunlu dosya eşitlemesi eklendi.
- Daha uzun kodlu diller için kendiliğinden öneriler düzeltildi.
- Yeni dizge ekleme başarımı iyileştirildi.
- Kalite denetimlerinde çeşitli hata düzeltmeleri yapıldı.
- Çeşitli başarımlar iyileştirmeleri yapıldı.
- *Weblate uygulamasını keşfedin* ile bütünleştirme eklendi.
- Salt okunur dizgelerin denetlenmesi davranışı düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.35 Weblate 4.5.1

5 Mart 2021 tarihinde yayınlandı.

- Bazı uç durumlarda yaşanan sözlük imlerinin düzenlenmesi sorunu düzeltildi.
- Birkaç sayfanın başarımını artırmak için ölçümlerin kullanımı zenginleştirildi.
- TMX dosyalarında doğru kaynak dilin kaydedilmesi sağlandı.
- API kullanarak tek dilli PO yüklemelerinin işlenmesi iyileştirildi.
- Sözlük bileşenlerinde uyarı davranışı iyileştirildi.
- Markdown bağlantı denetimleri iyileştirildi.
- Öge yollarında sözlük ve kaynak dil belirtildi.
- Büyük projelerde bileşen listeleri sayfalandı.
- Çeviri, bileşen ya da proje silme başarımı iyileştirildi.
- Toplu düzenleme başarımı iyileştirildi.
- ODF dosyaları için “Düzenlenmesi gerekli” ve “Onaylandı” durumlarının korunması sorunu düzeltildi.
- Çeviri dosyası indirmelerini özelleştirme arayüzü iyileştirildi

Ayrıntılı olarak tüm değişiklikler.

4.36 Weblate 4.5

19 Şubat 2021 tarihinde yayınlandı.

- Gettext PO dosyasında kullanılan `lua-format` desteği eklendi.
- Bir bileşeni projeler arasında paylaşma özelliği eklendi.
- Birden çok biçim imiyle birden çok adsız değişken denetimi davranışı düzeltildi.
- Posta listesi alanı projelerden kaldırıldı ve yerine çevirmenler için genel yönergeler eklendi.
- Sahte yerel oluşturma eklentisi eklendi.
- TermBase eXchange dosyaları desteği eklendi.
- Bir im kullanarak dizge çeşitlerini el ile tanımlama desteği eklendi.
- Tutarlılık denetimlerinin başarımı iyileştirildi.
- Uzun dizgeler için çeviri belleği başarımı iyileştirildi.
- Açıklama metinlerinde arama desteği eklendi.
- İki dilli biçimlerde dizgelerin eklenmesi ve silinmesi özelliği eklendi.
- Amazon Translate makine çevirisinde desteklenen dillerin listesi zenginleştirildi.
- Java özellikleri için Java MessageFormat denetimleri kendiliğinden etkinleştirildi.
- Bir çeviriye yeni dizgeler eklemek için yeni bir yükleme yöntemi eklendi.
- Çeviriye göz atmak için basit bir arayüz eklendi.
- Sözlükler artık normal bileşenler olarak tutuluyor.
- Artık bileşen API uygulaması kullanıldığından sözlüklere özel API kaldırıldı.
- Bazı imleri değiştirmek için basitleştirilmiş arayüz eklendi.
- Sözlüğe çevrilemeyen veya yasak terimler desteği eklendi.

- Sözlüklere terminoloji tanımlama özelliği eklendi.
- Görsel klavyeye daha fazla alan sağlamak için metin yönünü değiştirme seçeneği taşındı.
- Kullanıcının katkıda bulunduğu projeleri kendiliğinden izleme seçeneği eklendi.
- Çevirinin sözlük ile eşleşip eşleşmediği denetimi eklendi.
- Gezinme metninin rengini belirleme özelliği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.37 Weblate 4.4.2

14 Ocak 2021 tarihinde yayınlandı.

- Dağıtılan bir MO dosyasının bozulması sorunu düzeltildi.

4.38 Weblate 4.4.1

13 Ocak 2021 tarihinde yayınlandı.

- Çoğul değişikliklerini geri alma sorunu düzeltildi.
- Proje ayarları için yardım görüntüleme sorunu düzeltildi.
- Kullanıcı yönetimi iyileştirildi.
- Tek dilli PO dosyalarında bağlamın işlenmesi iyileştirildi.
- Temizleme eklentisinin HTML, ODF, IDML ve Windows RC biçimlerine davranışı düzeltildi.
- CSV dosyalarında yaşanan konumun işlenmesi sorunu düzeltildi.
- Dosya indirmeleri için içerik sıkıştırmasının kullanılması sağlandı.
- ZIP dosyasından içe aktarmada kullanıcı deneyimi iyileştirildi.
- Yükleme dosya biçimlerinin algılanması iyileştirildi.
- Pagure için yinelenen çekme istekleri engellendi.
- Hayalet çevirilerin görüntülenmesi başarımlı iyileştirildi.
- Çeviri düzenleyicisi yerel tarayıcı metin alanını kullanacak biçimde yeniden uygulandı.
- Temizleme eklentisinin eklenen yeni dizgeleri bozması sorunu düzeltildi.
- Eklentiler için API eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.39 Weblate 4.4

15 Aralık 2020 tarihinde yayınlandı.

- Bileşen oluştururken doğrulama iyileştirildi.
- Weblate gereksinimi Django 3.1 olarak güncellendi.
- Yönetim arayüzüne görünümü özelleştirme özelliği eklendi.
- Toplu düzenlemede yaşanan salt okunur durum işleme sorunu düzeltildi.
- CodeMirror bütünleştirilmesi iyileştirildi.

- Çeviri dosyalarından boş dizgeleri kaldırma eklentisi eklendi.
- Çeviriler için CodeMirror düzenleyicisinin kullanılması sağlandı.
- XML, HTML, Markdown ve reStructuredText için çeviri düzenleyiciye söz dizimi vurgulaması eklendi.
- Çeviri düzenleyicide yerleştirilebilir ögelerin vurgulanması sağlandı.
- Standart olmayan dil kodlarının desteği iyileştirildi.
- Belirsiz dil kodları kullanıldığında uyarı eklendi.
- Yeni bir çeviri eklenirken kullanıcıya süzölmüş bir dil listesinin sunulması sağlandı.
- Geçmişteki değişiklikler için arama yetenekleri zenginleştirildi.
- Fatura ayrıntı sayfaları ve Libre barındırma iş akışı iyileştirildi.
- Çeviri istatistikleri API uygulaması zenginleştirildi.
- Çeviri yaparken “diğer çeviriler” sekmesi iyileştirildi.
- Görevler API uygulaması eklendi.
- Dosya yükleme başarımı iyileştirildi.
- Kullanıcı tarafından tanımlanan özel karakterlerin görüntülenmesi iyileştirildi.
- Kendiliğinden çeviri başarımı iyileştirildi.
- Kullanıcı arayüzünde birkaç küçük iyileştirme yapıldı.
- ZIP indirmelerinin adlandırılması iyileştirildi.
- İzlenmeyen projelerin bildirimlerini alma seçeneği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.40 Weblate 4.3.2

4 Kasım 2020 tarihinde yayınlandı.

- Belirli bileşen dosya maskeleriyle yaşanan çökme sorunu düzeltildi.
- Ard arda yinelenen sözcük denetiminin doğruluğu iyileştirildi.
- Pagure çekme istekleri desteği eklendi.
- Başarısız olan hesap açma işlemlerinin hata iletileri iyileştirildi.
- Geliştirici yorumlarının Markdown olarak görüntülenmesi geri alındı.
- “master” dalından farklı bir varsayılan dal kullanan Git depolarının kurulumu basitleştirildi.
- Yeni oluşturulan iç depoların varsayılan dal olarak main kullanması sağlandı.
- reStructuredText çevrilirken değiştirilmemiş çevirinin yanlış pozitif sonuç oranı azaltıldı.
- Bazı durumlarda karşılaşılan CodeMirror görüntüleme sorunları düzeltildi.
- Anlamını netleştirmek için Kalıp grubunun adı “Kaynaklar” olarak değiştirildi.
- Yolları uzun olan depolar için GitLab çekme istekleri düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.41 Weblate 4.3.1

21 Ekim 2020 tarihinde yayınlandı.

- Kendiliğinden çeviri başarımı iyileştirildi.
- Kimliği doğrulanmış kullanıcıların oturumlarının sona ermesi sorunu düzeltildi.
- Sürüm bilgilerini gizleme özelliği eklendi.
- Bitbucket sunucusuyla kanca uyumluluğu iyileştirildi.
- Çeviri belleği güncellemelerinin başarımı iyileştirildi.
- Bellek kullanımı azaltıldı.
- Matris görünümünün başarımı iyileştirildi.
- Bir kullanıcının bir projeden kaldırılması için onay adımı eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.42 Weblate 4.3

15 Ekim 2020 tarihinde yayınlandı.

- API uygulamasına kullanıcı istatistikleri eklendi.
- Sayfalanmış sayfalardaki bileşen sıralaması sorunu düzeltildi.
- Bir sözlük için kaynak dil tanımlaması eklendi.
- GitHub ve GitLab çekme istekleri desteği yeniden yazıldı.
- Öneri kaldırıldıktan sonra istatistik sayılarının doğru olması sorunu düzeltildi.
- Herkese açık kullanıcı profili zenginleştirildi.
- Zorunlu denetimlerin yapılandırılması sorunu düzeltildi.
- İç yedekleme belgeleri iyileştirildi.
- Kaynak dil özniteliği projeden bir bileşene taşındı.
- Vue I18n biçimlendirme denetimi eklendi.
- Genel yer belirtici denetimine kurallı ifade desteği eklendi.
- Matris kipinin görünümü iyileştirildi.
- Makine çevirisinin adı kendiliğinden öneriler olarak değiştirildi.
- Birden çok GitLab ya da GitHub kopyası ile etkileşim desteği eklendi.
- API, proje güncellemelerini, birim güncellemeleri ve silmeleri ile sözlükleri kapsayacak zenginleştirildi.
- Birim API uygulamasının çoğul dizgeleri düzgün şekilde işlemesi sağlandı.
- Bileşen oluşturma ZIP dosyası ya da belge yüklemesini işleyebilmesi sağlandı.
- API yanıt durum kodları birleştirildi.
- Katılımcı anlaşmasında Markdown kullanılabilmesi sağlandı.
- Kaynak dizge izlemesi iyileştirildi.
- JSON, YAML ve CSV biçimleriyle uyumluluk iyileştirildi.
- Dizgeleri silme özelliği eklendi.
- Dosya indirme başarımı iyileştirildi.

- Depo yönetimi görünümü iyileştirildi.
- Android için Java biçiminin kendiliğinden etkinleştirilmesi sağlandı.
- Yerelleştirilmiş ekran görüntüleri özelliği eklendi.
- Python 3.9 desteği eklendi.
- Belirli koşullarda yaşanan HTML dosyalarının çevrilmesi sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.43 Weblate 4.2.2

2 Eylül 2020 tarihinde yayınlandı.

- JSON biçimlerinde yaşanan kaynak dizgelerin eşleşmesi sorunu düzeltildi.
- Bazı kimlik doğrulama yapılandırmaları için oturum açma yönlendirmesi düzeltildi.
- Grup eşitleme ile LDAP kimlik doğrulaması sorunu düzeltildi.
- Kendiliğinden çeviri ilerlemesini bildirirken yaşanan çökme sorunu düzeltildi.
- Art bilgiler etkinken Git işlemlerinin bir araya toplanması sorunu düzeltildi.
- API kullanarak yerel sürüm denetimi sistemi bileşenleri oluşturma sorunu düzeltildi.

4.44 Weblate 4.2.1

21 Ağustos 2020 tarihinde yayınlandı.

- Android kaynaklarında yaşanan bazı yerel ayarlar için çoğul biçimlerin kaydedilmesi sorunu düzeltildi.
- Bazı XLIFF dosyaları için temizleme eklentisinde yaşanan çökme sorunu düzeltildi.
- Docker kalıbında yerelleştirme CDN sunucusu kurulması sağlandı.

4.45 Weblate 4.2

18 Ağustos 2020 tarihinde yayınlandı.

- Kullanıcı sayfaları iyileştirildi ve kullanıcı listesi eklendi.
- 3.x sürümlerinden aktarım desteği kaldırıldı. 4.1 ya da 4.0 sürümünden aktarım yapın.
- Birkaç tek dilli biçime dışa aktarım özelliği eklendi.
- Etkinlik çizelgeleri iyileştirildi.
- Görüntülenecek Yakındaki dizgeler sayısının yapılandırılması sağlandı.
- Depo hataları ile karşılaşılan bileşenler için kilitleme desteği eklendi.
- Ana gezinme basitleştirildi (düğmeler simgelerle değiştirildi).
- Google Çeviri bütünleştirmesinde dil kodunun işlenmesi iyileştirildi.
- Git bir araya toplama eklentisi `Co-authored-by`: art bilgileri oluşturabilir.
- Arama sorgusu işleyici iyileştirildi.
- Biçim dizgesi denetimlerinin kullanıcıya bildirilmesi iyileştirildi.
- Toplu durum değişikliklerinin başarımı iyileştirildi.

- Proje ya da bileşen yeniden adlandırıldıktan sonra uyumluluk yönlendirmeleri eklendi.
- Dizge onayı, bileşen kitleme ve lisans değişikliği bildirimleri eklendi.
- ModernMT desteği eklendi.
- Dosya yüklenirken onaylanan çevirilerin üzerine yazılmasından kaçınılması sağlandı.
- Bazı uyumluluk adres yönlendirmelerinin desteği kaldırıldı.
- ECMAScript kalıp sabitleri denetimi eklendi.
- Bir bileşeni izleme seçeneği eklendi.
- JSON birim anahtarlarının başındaki nokta kaldırıldı.
- Çeviri belleği için ayrı Celery kuyruğu kaldırıldı.
- Tüm bileşenlerin bir dili bir kerede çevirmesi sağlandı.
- Content-Security-Policy HTTP üst bilgilerinin yapılandırılması sağlandı.
- Dillere proje düzeyinde takma ad verme özelliği eklendi.
- HTML ya da JavaScript yerelleştirmesine yardımcı olmak için yeni eklenti eklendi. Bilgi almak için: [JavaScript yerelleştirme CDN](#).
- Weblate etki alanının ayarlar bölümünden yapılandırılması sağlandı. Bilgi almak için: [SITE_DOMAIN](#).
- Bileşene ve projeye göre arama desteği eklendi.

4.46 Weblate 4.1.1

19 Haziran 2020 tarihinde yayınlandı.

- Docker üzerinde kendiliğinden düzeltme ya da eklenti yapılandırmasının değiştirilmesi sorunu düzeltildi.
- “Hakkında” sayfasında yaşanabilecek çökme sorunu düzeltildi.
- Bayt koduna derlenmiş yerel ayar dosyalarının kurulumu iyileştirildi.
- Sözlüğe sözcük ekleme sorunu düzeltildi.
- Makine çevirisi için klavye kısayolları sorunu düzeltildi.
- Bazı kurulumlarda günlük olaylarının atlanmasına neden olan hata ayıklama çıktısı kaldırıldı.
- Proje listesindeki kilit göstergesi sorunu düzeltildi.
- Bazı kurulumlarda GPG anahtarlarının listelenmesi sorunu düzeltildi.
- Kullanılacak DeepL API sürümünün seçme özelliği eklendi.
- SAML hizmet sağlayıcısı olarak davranma özelliği eklendi. Bilgi almak için: [SAML kimlik doğrulaması](#).

4.47 Weblate 4.1

15 Haziran 2020 tarihinde yayınlandı.

- Belirtilen ülke koduyla yeni çeviriler oluşturma özelliği eklendi.
- Ekran görüntüsü ile kaynak dizgeleri arama özelliği eklendi.
- İstatistik içgörülerindeki bilgiler zenginleştirildi.
- “Çeviri” sayfalarında arama düzenlemesi iyileştirildi.
- Eşzamanlı depo güncellemelerinin işlenmesi iyileştirildi.
- Proje oluşturma formuna kaynak dil eklendi.

- Çevirmen hesaplarına değişiklik sayısı eklendi.
- Bazı durumlarda kullanıcı arayüzü dil seçimi düzeltildi.
- Hesap açma kapalıyken kayıt yöntemlerinin beyaz listeye eklenmesi sağlandı.
- Sözlükte ilgili terimlerin aranması iyileştirildi.
- Çeviri belleği eşleşmeleri iyileştirildi.
- Aynı makine sonuçları gruplandırıldı.
- Ekran görüntüsünü çeviri sayfasında düzenlemek için doğrudan bağlantı eklendi.
- Silme onayı penceresi iyileştirildi.
- ZIP indirmeye kalıplar katıldı.
- Duyurulara Markdown ve bildirim yapılandırması özellikleri eklendi.
- Denetim listelerindeki ayrıntılar zenginleştirildi.
- Yeni dosya biçimleri desteği eklendi: *Laravel PHP dizgeleri*, *HTML dosyaları*, *OpenDocument biçimi*, *IDML biçimi*, *Windows RC dosyaları*, *INI çevirileri*, *Inno Setup INI çevirileri*, *GWT özellikleri*, *go-i18n JSON dosyaları*, *ARB dosyası*.
- Yok sayılan denetimlerin durumu için tutarlı olarak yok sayıldı kullanıldı.
- Etkinleştirilecek varsayılan eklentileri yapılandırma özelliği eklendi.
- Yok sayılan denetimler için düzenleyici klavye kısayolu düzeltildi.
- Yer belirtici bulunan dizgelerin makine çevirisi iyileştirildi.
- Kullanıcı dillerini başlatmayı kolaylaştırmak için hayalet çevirinin görüntülenmesi sağlandı.
- Dil kodunun işlenmesi iyileştirildi.
- Kullanıcı dilindeki çevirilerin listenin ilk sırasında görüntülenmesi sağlandı.
- Şekillendiriciler adı daha genel olan Çeşitler olarak değiştirildi.
- Yeni kalite denetimleri eklendi: *Birden çok değişken adlandırılmamış*, *Uzun süredir çevrilmemiş*, *Art arda yinelenen sözcükler var*.
- Çeviri belleğini boşaltma özelliği yeniden eklendi.
- Kaynak denetimlerini yok sayma seçeneği düzeltildi.
- Değişiklikleri itmek için farklı dallar yapılandırma özelliği eklendi.
- API uygulamasına HTTP üst bilgilerinde hız sınırlama durumunu bildirme özelliği eklendi.
- Google Translate V3 API (gelişmiş) desteği eklendi.
- Bileşen düzeyinde erişimi kısıtlama özelliği eklendi.
- Çeviri imlerinde boşluk ve diğer özel karakterleri kullanma desteği eklendi. Bilgi almak için: *Parametreleri kullanarak davranışı özelleştirmek*.
- Etkinse oluşturulan metin denetiminin her zaman görüntülenmesi sağlandı.
- API uygulamasına değişiklikleri süzme özelliği eklendi.
- Sözlüklerin projeler arasında paylaşılması özelliği eklendi.

4.48 Weblate 4.0.4

7 Mayıs 2020 tarihinde yayınlandı.

- Bazı Python 3.8 ortamlarında yaşanan sına paketini çalıştırma sorunu düzeltildi.
- Belgelerdeki yazım hataları düzeltildi.
- Bazı durumlarda yaşanan API kullanılarak bileşenlerin oluşturulması sorunu düzeltildi.
- Mobil gezinmeyi bozan JavaScript hataları düzeltildi.
- Bazı denetimlerin görüntülenmesinde yaşanan çökme sorunu düzeltildi.
- Ekran görüntüleri listesi düzeltildi.
- Aylık özet bildirimleri düzeltildi.
- Çeviride bulunmayan birimlerdeki ara çeviri davranışı düzeltildi.

4.49 Weblate 4.0.3

2 Mayıs 2020 tarihinde yayınlandı.

- Raporlarda yaşanabilecek kilitlenme sorunu düzeltildi.
- Yorumlardaki kullanıcı anmalarının büyük/küçük harfe duyarsız olması sağlandı.
- Süper kullanıcı olmayan kullanıcılar için PostgreSQL aktarımı düzeltildi.
- Bileşen oluşturulurken depo adresinin değiştirilmesi sorunu düzeltildi.
- Yukarı akış deposu kaybolduğunda yaşanan kilitlenme sorunu düzeltildi.

4.50 Weblate 4.0.2

27 Nisan 2020 tarihinde yayınlandı.

- Çeviri istatistiklerinin başarıımı iyileştirildi.
- Etiketleri değiştirme başarıımı iyileştirildi.
- Toplu düzenleme başarıımı iyileştirildi.
- Çeviri belleği başarıımı iyileştirildi.
- Bileşen silinirken yaşanabilecek çökme sorunu düzeltildi.
- Bazı uç durumlarda yaşanan çeviri değişikliklerinin görüntülenme sorunu düzeltildi.
- Çok uzun Celery kuyruğu uyarısı iyileştirildi.
- Tutarlılık denetiminde yaşanması olası yanlış pozitif sonuç sorunu düzeltildi.
- Bağlı bileşen deposu değiştirilirken yaşanan kilitlenme sorunu düzeltildi.
- Değişiklik listesine, CSV dosyalarına ve raporlara düzenleme uzaklığı eklendi.
- Kanada Fransızcası için noktalama işareti aralığı denetimlerinde yaşanan yanlış pozitif sonuç sorunu düzeltildi.
- XLIFF dışı aktarmasında yaşanan yer belirteciler sorunu düzeltildi.
- Sıfır genişlik denetiminde yaşanan yanlış pozitif sonuç sorunu düzeltildi.
- Yapılandırma hatalarının raporlanması iyileştirildi.
- İki dilli kaynak yükleme sorunu düzeltildi.

- DeepL makine çevirisi için desteklenen dillerin kendiliğinden algılanması sağlandı.
- Bazı uç durumlarda yaşanan sabit ilerleme çubuğu görüntüsü sorunu düzeltildi.
- Çevrilmemiş dizgelerde yaşanan bazı denetimlerin tetiklenmesi sorunu düzeltildi.

4.51 Weblate 4.0.1

16 Nisan 2020 tarihinde yayınlandı.

- PyPI ile yaşanan paket kurulumu sorunu düzeltildi.

4.52 Weblate 4.0

16 Nisan 2020 tarihinde yayınlandı.

- Weblate için artık Python 3.6 ya da üzeri gerekiyor.
- Bileşen uyarılarına yönetim özeti eklendi.
- Bozuk depo tarayıcı adreslerine bileşen uyarısı eklendi.
- Oturum ve hesap açma sayfaları iyileştirildi.
- Proje erişim denetimi ve iş akışı yapılandırması proje ayarlarıyla bütünleştirildi.
- i18next için ara değerlendirme ve iç içe işleme denetimi ve vurgulayıcı eklendi.
- Yüzde yer belirtecileri için denetim ve vurgulayıcı eklendi.
- Denetim hatalarında önerilerin görüntülenmesi sağlandı.
- Geçmişe kaynak dizge değişikliklerinin kaydedilmesi sağlandı.
- Microsoft Translator, API 3 sürümüne yükseltildi.
- Çeviri belleği arka ucu yeniden yazıldı.
- *Arama* içine birkaç `is` : araması desteği eklendi.
- İç kara listeler için *Çeviri değiştirilmemiş* kullanılması sağlandı.
- Tek dilli po dosyalarındaki yorumların ayıklanması iyileştirildi.
- Beyaz tahta iletileri duyurular olarak yeniden adlandırıldı.
- Hesap açma e-postalarında bazen yaşanan sorunlar düzeltildi.
- Daha fazla söz dizimi çeşidini işlemek için LINGUAS güncelleme eklentisi iyileştirildi.
- Tek dilli XLIFF kaynak dosyasını düzenlenme sorunu düzeltildi.
- *Arama* içine tam eşleşme desteği eklendi.
- API, ekran görüntüleri, kullanıcılar, gruplar, bileşen listeleri ve gelişmiş proje oluşturma işlemlerini kapsayacak biçimde zenginleştirildi.
- İki dilli çevirilere kaynak yükleme desteği eklendi.
- Geliştiricilerden ara dil desteği eklendi.
- Kaynak dizgeleri onaylama desteği eklendi.
- Platform genelinde çeviri belleği için indirme seçenekleri zenginleştirildi.

4.53 Weblate 3.x serisi

4.53.1 Weblate 3.11.3

11 Mart 2020 tarihinde yayınlandı.

- Belirli önceliği olan alanların aranması sorunu düzeltildi.
- Son eklenen dizgeler için önceden tanımlanmış sorgu düzeltildi.
- Aramada çift eşleşmelerin görüntülenmesi sorunu düzeltildi.
- Gmail üzerinde bildirimlerin görüntülenmesi sorunu düzeltildi.
- Geçmişteki değişikliklerin geri alınması sorunu düzeltildi.
- Özet bildirimlerindeki etkinliklere bağlantılar eklendi.
- Hesap silme onayı e-postası düzeltildi.
- Docker kapsayıcısında Slack kimlik doğrulaması desteği eklendi.
- Abone olunmayan diller için bildirim gönderilmesi engellendi.
- Başarım özetine Celery kuyrukları katıldı.
- Eklentiler için belge bağlantıları sorunu düzeltildi.
- Değiştirilmemiş çeviri denetiminde hatalı negatifler azaltıldı.
- CVE-2020-6802 sorununu çözmek için bleach bağımlılığı artırıldı.
- Geçmişte projesi düzeyindeki değişikliklerin listelenmesi sorunu düzeltildi.
- Bazı uç durumlarda yaşanan istatistiklerin geçersiz kılınması sorunu düzeltildi.
- Belirli dizge durumları için arama düzeltildi.
- Biçim dizgesi denetimi, eksik yüzde simgesini algılayacak şekilde iyileştirildi.
- Bazı üçüncü taraf hizmet sağlayıcılarında yaşanan kimlik doğrulama sorunu düzeltildi.

4.53.2 Weblate 3.11.2

22 Şubat 2020 tarihinde yayınlandı.

- Önerilerin görüntülenmesi düzeltildi.
- Bazı dizgelerin yanlış bir biçimde sözcük içermediğinin bildirilmesine yol açan sorun düzeltildi.

4.53.3 Weblate 3.11.1

20 Şubat 2020 tarihinde yayınlandı.

- Celery kurulum değişiklikleri belgesi yazıldı.
- Bileşen oluşturulurken dosya adı doğrulaması iyileştirildi.
- Bazı bağımlılıkların en düşük sürümleri düzeltildi.
- Belirli Django sürümlerini kullanan grupların eklenmesi sorunu düzeltildi.
- Yukarı akış deposuna el ile itme sorunu düzeltildi.
- Sözlük eşleştirmesi iyileştirildi.

4.53.4 Weblate 3.11

17 Şubat 2020 tarihinde yayınlandı.

- API ile bileşen oluşturulurken sürüm denetimi sistemi itme adresinin kullanılması sağlandı.
- Görüntülenen genişlik denetimi artık görüntüye görseli de ekliyor.
- Bildirim e-postalarındaki bağlantılar düzeltildi.
- Düz metin içerikli e-postaların görünümü iyileştirildi.
- Yok sayılan denetimlerin görüntülenmesi ve yeniden etkinleştirilmesi sağlandı.
- Tek dilli çevirilerde yakın anahtarların görüntülenmesi sağlandı.
- Dizge şekillendirmelerini gruplama özelliği eklendi.
- Sistem denetimlerine yeni Weblate sürümlerine yükseltme önerisi eklendi.
- Yinelenen dil uyarısı için daha ayrıntılı inceleme sağlandı.
- Proje sayfalarına ayrıntılı lisans bilgileri eklendi.
- Gerektiğinde yerel kopyaların sıklık düzeyinin kendiliğinden kaldırılması sağlandı.
- İşlem gerektiren dizgelerin indirilmesinde yaşanan sorun düzeltildi.
- Aynı dosya modelinin iki kez kullanılmasına karşı uyarı eklendi.
- Yerleştirilebilir XML öğelerinin ayıklanması iyileştirildi.
- `SINGLE_PROJECT` ayarına seçilmiş projeye yönlendirme dayatması eklendi.
- Yorumları çözümleme özelliği eklendi.
- İmleri toplu olarak düzenleme özelliği eklendi.
- labels desteği eklendi.
- Toplu düzenleme eklentisi eklendi.
- *Denetimleri dayatmak* seçeneği eklendi.
- Onay bağlantılarının varsayılan geçerlilik süresi artırıldı.
- Matomo bütünleştirmesi iyileştirildi.
- Kaynak dizge değişikliklerinin düzgün işlenmesi için *Çevrilmiş* düzeltildi.
- Kendiliğinden güncelleme yapılandırması `AUTO_UPDATE` seçeneği ile zenginleştirildi.
- LINGUAS eklentilerinin Weblate üzerindeki çevirileri tam olarak eşitlemesi sağlandı.

4.53.5 Weblate 3.10.3

18 Ocak 2020 tarihinde yayınlandı.

- translate-toolkit 2.5.0 desteği eklendi.

4.53.6 Weblate 3.10.2

18 Ocak 2020 tarihinde yayınlandı.

- Projelere kilit göstergesi eklendi.
- Bazı tarayıcılarda titremeye neden olan CSS hatası düzeltildi.
- Yerel ayarları İngilizce olmayan sistemlerde yaşanan arama sorunu düzeltildi.
- GitHub ve Bitbucket kancaları için depo eşleştirmesi iyileştirildi.
- Bazı Python 2.7 kurulumlarında yaşanan veri aktarımı sorunları düzeltildi.
- Git için sıklık düzeyi ile kopyalama yapılandırılması özelliği eklendi.
- Arka planda bildirimlerin işlenmesi iyileştirildi.
- Web tarayıcısına geri dönerken yaşanan formun bozuk gönderilmesi sorunu düzeltildi.
- YAML biçimlendirmesini yapılandırmak için yeni eklenti eklendi.
- Tek çoğul biçimli dillerde tetiklenmemesi için çoğul biçimler aynı denetimi düzeltildi.
- Bazı alanlarda kurallı ifade aramasında yaşanan sorun düzeltildi.

4.53.7 Weblate 3.10.1

9 Ocak 2020 tarihinde yayınlandı.

- API çeviri oluşturma özelliği ile zenginleştirildi.
- Veri aktarımlarında yaşanan bazı uç durumlar düzeltildi.
- Django 3.0 uyumluluğu sağlandı.
- Veri temizleme başarımı iyileştirildi.
- Özelleştirilebilir security.txt desteği eklendi.
- Değişiklik günlüğünde sayfa yolları iyileştirildi.
- Panoda çeviri listesinin görünümü iyileştirildi.
- Web kancaları için HTTP yanıtları iyileştirildi.
- Docker kapsayıcısına GitLab birleştirme istekleri desteği eklendi.

4.53.8 Weblate 3.10

20 Aralık 2019 tarihinde yayınlandı.

- Uygulama kullanıcı arayüzü iyileştirildi.
- Çift boşluk denetimi eklendi.
- Yeni diller oluşturulurken yaşanan sorun düzeltildi.
- Silinmiş e-posta adreslerine denetim günlüğü bildirimlerinin gönderilmesi engellendi.
- Salt okunur dizge desteği eklendi.
- Yorumlara Markdown desteği eklendi.
- Proje bilgilerine çeviri yönerge metninin yerleştirilmesi sağlandı.
- İkincil diller için panoya kopyalama özelliği eklendi.
- Mercurial desteği iyileştirildi.
- Git deposundan alma başarımı iyileştirildi.

- Dizge yaşını arama özelliği eklendi.
- Tüm çeviriler için kaynak dilin görüntülenmesi sağlandı.
- Yakın dizgelerin bağlamının görüntülenmesi sağlandı.
- Depo işlemleriyle ilgili bildirimler desteği eklendi.
- Çevirilerin listelenmesi iyileştirildi.
- Arama yetenekleri zenginleştirildi.
- Düzenlenecek olarak imlenmiş kendiliğinden çeviri dizgeleri desteği eklendi.
- Bağlı bileşen uyarıları için yinelenen bildirimlerin gönderilmesi engellendi.
- Varsayılan birleştirme isteği iletisi iyileştirildi.
- Zen kipinde dizge durumunun daha iyi belirtilmesi sağlandı.
- Yandex Translate için daha fazla dil desteği eklendi.
- Bildirim e-postalarının görünümü iyileştirildi.
- Çeviri lisansı için seçenek sunuldu.

4.53.9 Weblate 3.9.1

28 Ekim 2019 tarihinde yayınlandı.

- Yedeklerden bazı gereksiz dosyalar kaldırıldı.
- Raporlardaki yaşanması olası kilitlenme sorunları düzeltildi.
- Veri tabanları arasında aktarma sorunu düzeltildi.
- Git depolarına dayatarak itme desteği eklendi.
- Kayıt belirtecinin geçersiz kılınması riski azaltıldı.
- Hesap silme işleminin hız sınırlayıcıya takılma sorunu düzeltildi.
- Önceliğe göre arama özelliği eklendi.
- JSON dosyasına dizge eklenirken yaşanabilecek çökme sorunu düzeltildi.
- Güvenli HTML denetimi ve düzeltmesinin kaynak dizgenin markup koduna uyması sağlandı.
- Davet edilmiş ve silinmiş kullanıcılara bildirim gönderilmesi engellendi.
- Docker kapsayıcısında Celery içindeki Redis SSL bağlantısı düzeltildi.

4.53.10 Weblate 3.9

15 Ekim 2019 tarihinde yayınlandı.

- İndirilen dosyalara Weblate üst verileri eklendi.
- Başarısız olan denetimler için kullanıcı arayüzü iyileştirildi.
- Biçim denetimlerinde eksik dizgeler belirtildi.
- Fransızca noktalama aralığı için özel denetim eklendi.
- Bazı kalite denetimi hatalarını düzeltme desteği eklendi.
- Yeni projeler oluşturmak için ayrı izinler eklendi.
- İstatistikler karakter sayıları ile zenginleştirildi.
- Java biçimi dil kodları desteği iyileştirildi.

- Yer belirticiler için yeni genel denetim eklendi.
- WebExtension JSON yer belirticileri desteği eklendi.
- Düz XML biçimi desteği eklendi.
- API, proje, bileşen ve çeviri silme ve oluşturma özellikleri ile zenginleştirildi.
- Gitea ve Gitee internet kancaları desteği eklendi.
- Yeni özel kurallı ifade temelli denetim eklendi.
- Paylaşılan çeviri belleğine katkıda bulunma yapılandırması sağlandı.
- Birden fazla çeviri dosyasını indirmek için ZIP desteği eklendi.
- XLIFF standardı, en fazla genişlik ve yazı tipi işlenmesi ile uyumlu hale getirildi.
- Web uygulamalarını çevirmek üzere güvenli HTML biçimlendirmesi için yeni denetim ve düzeltici eklendi.
- Desteklenmeyen yapılandırma için bileşen uyarısı eklendi.
- İlk çevirileri hazırlamak için kendiliğinden çeviri eklentisi eklendi.
- Kendiliğinden çeviri, önerileri eklemek üzere zenginleştirildi.
- Özet görünümüne eklenti parametreleri eklendi.
- Sentry için Raven yerine modern Sentry SDK desteği eklendi.
- Örnek ayarları, üretim ortamına daha uygun olacak biçimde değiştirildi.
- BorgBackup kullanılarak kendiliğinden yedekleme özelliği eklendi.
- İstenmeyen dosya güncellemelerini önlemek için RESX için temizleme eklentisi ayrıldı.
- Gelişmiş arama özellikleri eklendi.
- Kullanıcıların kendi raporlarını indirmeleri sağlandı.
- Bileşenlerin yapılandırılmasına yardımcı olacak yerelleştirme rehberi eklendi.
- GitLab birleştirme istekleri desteği eklendi.
- Depo durumunun görüntülenmesi iyileştirildi.
- Kendiliğinden çeviri işlemlerinin arka planda yapılması sağlandı.

4.53.11 Weblate 3.8

15 Ağustos 2019 tarihinde yayınlandı.

- Benzer bileşenlerin basitleştirilmiş oluşturulması desteği eklendi.
- XML temelli dosya biçimleri için çeviri imlerini işleme desteği eklendi.
- İstisnaların Celery günlüğüne kaydedilmesi sağlandı.
- Depo kapsamındaki eklentilerin başarımı iyileştirildi.
- Bildirim e-postalarının görünümü iyileştirildi.
- Parola sıfırlama davranışı düzeltildi.
- Çeviri sayfalarının çoğunda başarımlar iyileştirildi.
- Weblate tarafından bilinmeyen dillerin listesi düzeltildi.
- Keşfedilen bileşenlere eklentileri kopyalama özelliği eklendi.
- Dosya içeriğinin yüklenenlerle değiştirilmesi desteği eklendi.
- Sürüm denetimi sistemi temelli olmayan içeriğin çevrilmesi desteği eklendi.
- Sosyal ağlarda kullanılmak üzere OpenGraph pano bileşeni görseli eklendi.

- Canlandırılmalı ekran görüntüleri desteği eklendi.
- Tek dilli XLIFF dosyalarının işlenmesi iyileştirildi.
- Tek bir etkinlik için birden fazla bildirim gönderilmesi engellendi.
- Değişiklikleri süzme desteği eklendi.
- Raporlama için önceden tanımlanmış süreler zenginleştirildi.
- Azure depoları için internet kancası desteği eklendi.
- Bekleyen öneriler veya çevrilmemiş dizgeler ile ilgili yeni abone olma bildirimleri eklendi.
- Bildirim e-postalarına tek tıkla abonelikten ayrılma bağlantısı eklendi.
- Çevrilmiş denetiminde yaşanan yanlış pozitif sonuç sorunu düzeltildi.
- Yöneticiler için yeni yönetim arayüzü eklendi.
- Dizge önceliğinin imler kullanılarak belirlenmesi sağlandı.
- Dil yönetimi görünümleri eklendi.
- Qt kitaplığı ve Ruby biçim dizgeleri için denetimler eklendi.
- Tek proje kurulumlarına daha uygun yapılandırma eklendi.
- Tek dilli çevirilerdeki kaynak dizge değişikliğinde yeni dizge için bildirim eklendi.
- Arama özelliği olan çeviri belleği için ayrı bir görünüm eklendi.

4.53.12 Weblate 3.7.1

28 Haziran 2019 tarihinde yayınlandı.

- Belgeler güncellendi.
- Bazı gereksinim kısıtlamaları düzeltildi.
- Dil veri tabanı güncellendi.
- Yerelleştirme güncelleştirmeleri yapıldı.
- Çeşitli kullanıcı arayüzü iyileştirmeleri yapıldı.
- Desteklenmeyen ancak keşfedilen çeviri dosyalarının işlenmesi iyileştirildi.
- Eksik dosya biçimi gereksinimlerinin daha ayrıntılı olarak bildirilmesi sağlandı.

4.53.13 Weblate 3.7

21 Haziran 2019 tarihinde yayınlandı.

- Bildirimler için ayrı Celery kuyruğu eklendi.
- API ile göz atarken uygulamayla tutarlı bir görünüm kullanıldı.
- Raporlara onaylanmış istatistikleri eklendi.
- Çeviri bileşeni güncellenirken ilerleme durumunun bildirilmesi sağlandı.
- Çalışan arka plan bileşeni güncellemesini iptal etme özelliği eklendi.
- Dosya adı değiştirme işlemleri için kalıp dili zenginleştirildi.
- Düzenleyici bağlantısı ve depo tarayıcı adresi için kalıplar kullanıldı.
- Çeviri düzenirken en fazla uzunluk ve geçerli karakter sayısının belirtilmesi sağlandı.
- Değiştirilmemiş çeviri denetiminde kısaltmaların işlenmesi iyileştirildi.

- Yeni katkıda bulunanlar için giriş sayfası yenilendi.
- msgmerge eklentisini yapılandırma desteği eklendi.
- Bildirim gönderirken SMTP bağlantısının açılması geciktirildi.
- Hata günlüğü iyileştirildi.
- MO oluşturma eklentisinde özel konum kullanılması sağlandı.
- Eski önerileri veya yorumları temizleme eklentileri eklendi.
- Zen düzenleyicisinde yatay kipi etkinleştirme seçeneği eklendi.
- Birçok bağlı bileşenle içe aktarma başarımı iyileştirildi.
- Bazı durumlardaki kurulum örnekleri düzeltildi.
- Değişiklik uyarılarının görüntülenmesi iyileştirildi.
- Yeni yatay istatistikler pano bileşeni eklendi.
- Çoğul biçimler için biçim dizgeleri denetimi iyileştirildi.
- Yazı tipi yönetim aracı eklendi.
- Görüntülenen metin boyutları için yeni denetim eklendi.
- Alt yazı biçimleri desteği eklendi.
- Diller için genel tamamlanma istatistikleri eklendi.
- Proje kapsamı ve genel kapsamda raporlama özelliği eklendi.
- Çeviri durumunu görüntüleyen kullanıcı arayüzü iyileştirildi.
- Weblate logosu ve renk şeması yenilendi.
- Bit eşlem simgelerinin görünümü yenilendi.

4.53.14 Weblate 3.6.1

26 Nisan 2019 tarihinde yayınlandı.

- Tek dilli XLIFF dosyalarının işlenmesi iyileştirildi.
- Bazı uç durumlarda yaşanan özet bildirimleri sorunu düzeltildi.
- Eklenti betiği hata uyarısı sorunu düzeltildi.
- Tek dilli PO dosyaları için MO dosyası oluşturma sorunu düzeltildi.
- Kaldırılan denetimlerin görüntülenmesi sorunu düzeltildi.
- Proje listesinde yönetilen projelerin belirtilmesi sağlandı.
- Eksik sürüm denetimi sistemi deposundan kurtarma için güncellemeye izin verildi.

4.53.15 Weblate 3.6

20 Nisan 2019 tarihinde yayınlandı.

- Kullanıcı verilerini indirme desteği eklendi.
- Eklentilerin kurulum sırasında kendiliğinden tetiklenmesi sağlandı.
- Aktarım çakışmalarını çözümlmek için yönergeler iyileştirildi.
- Temizleme eklentisinin uygulama mağazası üst veri çevirileriyle uyumlu olması sağlandı.
- Yeni çeviriler eklenirken dil kodu söz diziminin yapılandırılabilmesi sağlandı.

- 2020 Nisan ayında desteğinin bitmesi planlanan Python 2 sürümünün kullanıldığı ile ilgili uyarı eklendi.
- Görsel klavye için kaynak dizgeden özel karakterlerin ayıklanması sağlandı.
- Katılımcı istatistikleri, kaynak ve hedef sayılarını yansıtacak şekilde zenginleştirildi.
- Çeviriler kullanıcılar için devre dışı bırakılmış olsa bile, yöneticilerin ve tutarlılık eklentilerinin çeviri yapabilmesi sağlandı.
- Language-Team üst bilgi değişikliğini devre dışı bırakma düğmesinin açıklaması düzeltildi.
- Yorumlarda anılan kullanıcıların bilgilendirilmesi sağlandı.
- Bileşen kurulumunda dosya biçiminin kendiliğinden algılanması özelliği kaldırıldı.
- Tek dilli PO dosyaları için MO dosyası oluşturma sorunu düzeltildi.
- Özet bildirimleri eklendi.
- Bileşen bildirimlerini sessize alma desteği eklendi.
- Yeni uyarılar, beyaz tahta iletileri ve bileşenler için bildirimler eklendi.
- Yönetilen projelerin bildirimlerinin yapılandırılabilmesi sağlandı.
- Üç harfli dil kodlarının işlenmesi iyileştirildi.

4.53.16 Weblate 3.5.1

10 Mart 2019 tarihinde yayınlandı.

- Celery systemd birimi örneği düzeltildi.
- Oturum açma ile HTTP depolarından gelen bildirimler düzeltildi.
- Tek dilli çeviriler için kaynak dizgeyi düzenlemedeki yarış durumu düzeltildi.
- Günlüklere başarısız olan eklenti çalıştırma çıktısı eklendi.
- Yeni dil ekleme seçeneklerinin doğrulanması iyileştirildi.
- Bileşen ayarlarında dosya biçimlerinin düzenlenmesi sağlandı.
- Kurulum yönergeleri Python 3 kullanacak biçimde güncellendi.
- Çevirilerin yüklenmesinde başarımlar ve tutarlılık iyileştirmeleri yapıldı.
- Microsoft Terminology hizmetinin geçerli Zeep sürümleriyle uyumlu olması sağlandı.
- Yerelleştirme güncelleştirmeleri yapıldı.

4.53.17 Weblate 3.5

3 Mart 2019 tarihinde yayınlandı.

- İç çeviri belleğinin başarımı iyileştirildi.
- Genel çeviri belleği yönetimi arayüzü eklendi.
- Hatalı bileşen durumu uyarısı iyileştirildi.
- Beyaz tahta iletilerini yönetme kullanıcı arayüzü eklendi.
- Eklenti işleme iletilerinin yapılandırılabilmesi sağlandı.
- Yukarı akış deposu güncellenirken işleme sayısı azaltıldı.
- Bileşenler projeler arasında taşınırken yaşanabilecek üst veri kaybı sorunu düzeltildi.
- Zen kipinde gezinme iyileştirildi.
- Birkaç yeni kalite denetimi eklendi (Markdown ve adresler ile ilgili).

- Uygulama mağazası üst veri dosyaları desteği eklendi.
- GitHub ya da Gerrit bütünleştirmesini değiştirme desteği eklendi.
- Kashida harfleri için denetim eklendi.
- İşlemeleri sorumluya göre bir araya toplama seçeneği eklendi.
- XLSX dosya biçimi desteği iyileştirildi.
- Tesseract 4.0 ile uyumluluk sağlandı.
- Faturalama eklentisine, ödenmemiş faturalar için 45 gün sonra projeleri silme özelliği eklendi.

4.53.18 Weblate 3.4

22 Ocak 2019 tarihinde yayınlandı.

- XLIFF yer belirtecileri desteği eklendi.
- Birden fazla Celery görev kuyruğu kullanma özelliği eklendi.
- Projeleri ve bileşenleri yeniden adlandırma ve taşıma desteği eklendi.
- Raporlara karakter sayıları eklendi.
- Çeviri dosyalarının kendiliğinden algılanmasıyla çeviri bileşenlerinin yönlendirme ile oluşturulması özelliği eklendi.
- Git işlemleri birleştirme iletilerinin özelleştirilebilmesi sağlandı.
- Gezinmede bileşen uyarılarının görsel olarak görüntülenmesini sağlandı.
- Çeviri dosyalarını yükleme başarımı iyileştirildi.
- Yollamadan önce Git işlemlerini bir araya toplama eklentisi eklendi.
- Çeviri değişikliklerinin görüntülenmesi iyileştirildi.
- Varsayılan birleştirme biçimi değiştirilerek yeniden temellendirildi ve yapılandırılabilmesi sağlandı.
- Dil kodundaki alt etiketlerin özel kullanımının işlenmesi iyileştirildi.
- Tam metin dizini güncelleme başarımı iyileştirildi.
- Dosya yükleme API işlevi daha fazla parametreyi destekleyecek biçimde zenginleştirildi.

4.53.19 Weblate 3.3

30 Kasım 2018 tarihinde yayınlandı.

- Bileşen ve proje silme desteği eklendi.
- Bazı tek dilli çeviriler için başarımlar iyileştirildi.
- Bir çeviriyle ilgili sorunları vurgulamak için çeviri bileşeni uyarıları eklendi.
- Yapılabildiğinde, XLIFF dizgesi resname değerinin bağlam olarak sunulması sağlandı.
- XLIFF durumları desteği eklendi.
- DATA_DIR içindeki yazılamayan dosyalar denetimi eklendi.
- Değişiklikler için CSV dışı aktarma işlemi iyileştirildi.

4.53.20 Weblate 3.2.2

20 Ekim 2018 tarihinde yayınlandı.

- Artık gerek duyulmayan Babel bağımlılığı kaldırıldı.
- Dil tanımları güncellendi.
- LDAP ve Celery eklentilerinin belgeleri iyileştirildi.
- Yeni dos-eol ve auto-java-messageformat imlerinin etkinleştirilmesinde yaşanan sorun düzeltildi.
- PyPI paketindeki setup.py sınamasının çalıştırılmasında yaşanan sorun düzeltildi.
- Çoğul biçimlerin işlenmesi iyileştirildi.
- Bazı uç durumlarda yaşanan çeviri yükleme API sorunu düzeltildi.
- El ile değiştirilirken Git yapılandırmasının güncellenmesinde yaşanan sorun düzeltildi.

4.53.21 Weblate 3.2.1

10 Ekim 2018 tarihinde yayınlandı.

- Python 2.7 sürümünde backports.csv belge bağımlılığı eklendi.
- Sınamaların Root olarak çalıştırılması sorunu düzeltildi.
- Gitexport modülünde hataların işlenmesi iyileştirildi.
- Yeni eklenen dillerin ilerleme raporlaması düzeltildi.
- Celery işlemi hatalarının Sentry üzerine doğru şekilde bildirilmesi sağlandı.
- Qt Linguist ile yeni çeviriler oluştururken yaşanan sorun düzeltildi.
- Ara sıra yaşanan tam metin dizini güncelleme sorunları düzeltildi.
- Yeni bileşenler oluşturulurken doğrulama iyileştirildi.
- Eski önerilerin temizlenmesi desteği eklendi.

4.53.22 Weblate 3.2

6 Ekim 2018 tarihinde yayınlandı.

- Kendiliğinden eklenti kurulumu için install_addon yönetim komutu eklendi.
- Hız sınırı ayarlarının daha ayrıntılı olması sağlandı.
- Excel dosyalarının dışa ve içe aktarılması desteği eklendi.
- Birden çok bileşen bulma eklentisi olması durumunda bileşen temizleme iyileştirildi.
- Microsoft Terminology makine çevirisi arka ucu yeniden yazıldı.
- Weblate üzerindeki bazı işlemlerin yükü Celery üzerine aktarıldı.
- Arama yetenekleri geliştirildi ve kurallı ifade araması eklendi.
- Youdao Zhiyun API makine çevirisi desteği eklendi.
- Baidu API makine çevirisi desteği eklendi.
- Celery kullanan bütünsel bakım ve temizleme görevleri eklendi.
- Çeviri yükleme başarımı neredeyse %25 oranında iyileştirildi.
- Yükleme sırasında üst bilgilerin birleştirilmesi özelliği kaldırıldı.
- Özel işleme iletileri özelliği kaldırıldı.

- Düzenleme kipinin (zen/tam) yapılandırılabilmesi sağlandı.
- Hataların Sentry üzerine bildirilmesi desteği eklendi.
- Depolara kendiliğinden güncelleme desteği eklendi.
- Kullanıcılar tarafından proje ve bileşen oluşturma özelliği eklendi.
- İç çeviri belleğinin yapılan çevirileri kendiliğinden kaydetmesi sağlandı.
- Kullanıcı ve projeler için var olan çeviri belleklerini içe aktarma özelliği eklendi.
- Ekran görüntüleri ile ilgili dizgelerin daha iyi yönetilmesi sağlandı.
- Java MessageFormat denetimi desteği eklendi.

Çözülen sorunların ayrıntılı listesine bakmak için [GitHub üzerindeki 3.2 kilometre taşı](#) bölümüne bakabilirsiniz.

4.53.23 Weblate 3.1.1

27 Temmuz 2018 tarihinde yayınlandı.

- Bazı kurulumlarda yaşanan sınama paketi sorunu düzeltildi.

4.53.24 Weblate 3.1

27 Temmuz 2018 tarihinde yayınlandı.

- 3.0.1 öncesi sürümlerden yükseltmeler desteklenmiyor.
- Ayarlar bölümüne varsayılan işleme iletilerini değiştirme özelliği eklendi.
- Web kancalarının kendi kendine barındırma ortamları ile uyumluluğunu iyileştirildi.
- Amazon Translate desteği eklendi.
- Django 2.1 uyumluluğu sağlandı.
- Django sistem denetimlerine kurulumla ilgili sorunları tanılama özelliği eklendi.
- Yakında kapatılacak libavatar hizmetinin desteği kaldırıldı.
- Değiştirilmemiş çevirileri “Düzenlenmesi gerekli” olarak imleyen eklenti eklendi.
- Çeviri yaparken belirli bir konuma atlama desteği eklendi.
- Çeviri indirme işleminin özelleştirilebilmesi sağlandı.
- Çeviri belleği eşleşmelerinde dizge benzerliğinin hesaplanması iyileştirildi.
- Git işlemlerini GnuPG tarafından imzalama özelliği eklendi.

4.53.25 Weblate 3.0.1

10 Haziran 2018 tarihinde yayınlandı.

- 2.20 sürümünden geçişte yaşanabilecek sorun düzeltildi.
- Yerelleştirme güncelleştirmeleri yapıldı.
- Kullanımdan kaldırılmış kanca örnekleri silindi.
- Ön bellek belgeleri iyileştirildi.
- Yönetici belgelerinin görüntülenmesinde yaşanan sorun düzeltildi.
- Uzun dil adlarının işlenmesi iyileştirildi.

4.53.26 Weblate 3.0

1 Haziran 2018 tarihinde yayınlandı.

- Erişim denetimi yeniden yazıldı.
- Modüllerin taşınmasına ve yeniden adlandırılmasına yol açan çeşitli kod temizlemeleri yapıldı.
- Yeni kendiliğinden bileşen bulma eklentisi eklendi.
- `import_project` yönetim komutunun parametreleri zenginleştirildi.
- Windows RC dosyaları için temel destek eklendi.
- Katkıda bulunan adlarını PO dosya üst bilgilerinde kaydeden yeni eklenti eklendi.
- Her bileşene özgü kanca betikleri özelliği kaldırıldı. Bunun yerine eklentiler kullanıldı.
- Katılımcı anlaşmalarını toplama desteği eklendi.
- Erişim denetimi değişikliklerinin geçmiş kayıtlarında görülmesi sağlandı.
- Bir projedeki tüm bileşenlerin çevirilerinin aynı olmasını sağlayan eklenti eklendi.
- İşleme iletisi kalıplarının daha fazla değişkeni desteklemesi sağlandı.
- Ek metin bağlamı yazma desteği eklendi.

4.54 Weblate 2.x serisi

4.54.1 Weblate 2.20

4 Nisan 2018 tarihinde yayınlandı.

- Subversion depolarını kopyalama hızı iyileştirildi.
- Depo kilitleme özelliği, üçüncü taraf kitaplığı kullanacak biçimde değiştirildi.
- Yalnızca işlem gerektiren dizgeleri indirme desteği eklendi.
- Aynı anda birkaç dilde arama yapma özelliği eklendi.
- Gettext çıktı kaydırmasını yapılandıran yeni eklenti eklendi.
- JSON biçimlendirmesini yapılandıran yeni eklenti eklendi.
- RFC 6750 uyumlu Bearer kimlik doğrulaması kullanılarak, API üzerine kimlik doğrulama desteği eklendi.
- Makine çevirisi hizmetlerini kullanan kendiliğinden çeviri özelliği eklendi.
- Beyaz tahta iletilerine HTML kod imi desteği eklendi.
- Dizge durumlarını toplu değiştirme özelliği eklendi.
- En düşük `translate-toolkit` sürümü 2.3.0 olarak güncellendi. Eski sürümler artık desteklenmiyor.
- İç çeviri belleği eklendi.
- Panoya bileşen listeleri özeti ve her bileşen listesinin özeti sayfaları eklendi.
- DeepL makine çevirisi hizmeti desteği eklendi.
- Makine çevirisi sonuçlarının Weblate ön belleğine alınması sağlandı.
- İşlenmiş değişiklikleri yeniden sıralama özelliği eklendi.

4.54.2 Weblate 2.19.1

20 Şubat 2018 tarihinde yayınlandı.

- 2.18 sürümünden yükseltme sırasında yaşanan aktarım sorunu düzeltildi.
- Dosya yükleme API doğrulaması iyileştirildi.

4.54.3 Weblate 2.19

15 Şubat 2018 tarihinde yayınlandı.

- Bazı dosya biçimlerinde yaşanan içe aktarma sorunları düzeltildi.
- İnsan dostu tarayıcı bilgilerinin denetim günlüğünde görüntülenmesi sağlandı.
- Dosyalar için TMX dışı aktarıcısı eklendi.
- Çeviri dosyalarının yüklenmesinde çeşitli başarımlar iyileştirmeleri yapıldı.
- Django kullanımını öne alacak biçimde Weblate erişim yönetimini devre dışı bırakma seçeneği eklendi.
- Büyük dizgeler için sözlük arama hızı iyileştirildi.
- django_auth_ldap 1.3.0 uyumluluğu sağlandı.
- Yapılandırma hatalarının kaydedilmesi ve kalıcı olarak bildirilmesi sağlandı.
- Beyaz boşluk düzelticisinde imlerin yok sayılmasına uyuldu.
- Bazı Subversion kurulumlarıyla uyumluluk geliştirildi.
- İç makine çevirisi hizmeti iyileştirildi.
- SAP Translation Hub hizmeti desteği eklendi.
- Microsoft Terminology hizmeti desteği eklendi.
- Bildirim e-postalarındaki reklam özelliği kaldırıldı.
- Dil düzeyinde çeviri ilerleme raporları iyileştirildi.
- Farklı çoğul biçim formülleri desteği iyileştirildi.
- stdlayout kullanmayan Subversion depoları desteği eklendi.
- Çeviri iş akışlarını özelleştirme eklentileri eklendi.

4.54.4 Weblate 2.18

15 Aralık 2017 tarihinde yayınlandı.

- Katılımcı istatistikleri zenginleştirildi.
- Özel karakterler sanal klavye yapılandırması iyileştirildi.
- DTD dosya biçimi desteği eklendi.
- Klavye kısayolları, tarayıcı ve sistem kısayolları ile çakışma olasılığını düşürmek için değiştirildi.
- XLIFF dosyaları için onaylandı imi desteği iyileştirildi.
- Gettext PO dosyalarında uzun dizgelerin kaydırılmaması özelliği eklendi.
- Geçerli çevirinin kalıcı bağlantısını kopyalama düğmesi eklendi.
- Django 1.10 desteği kaldırıldı ve Django 2.0 desteği eklendi.
- Çeviri sırasında çevirilerin kilitlenmesi özelliği kaldırıldı.
- Tek dilli çevirilere yeni dizgeler ekleme özelliği eklendi.

- Seçilmiş onaylayıcılar ile çeviri iş akışı özelliği eklendi.

4.54.5 Weblate 2.17.1

13 Ekim 2017 tarihinde yayınlandı.

- Bazı özel durumlarda sınama paketinin çalıştırılmasında yaşanan sorun düzeltildi.
- Yereller güncellendi.

4.54.6 Weblate 2.17

13 Ekim 2017 tarihinde yayınlandı.

- Git kopyalarının varsayılan olarak sık olması sağlandı.
- Büyük çeviri dosyalarını güncelleme başarımı artırıldı.
- Belirli e-posta adresleri ile hesap açılması engelleme özelliği eklendi.
- Kullanıcıların kendi yorumlarını silebilmesi özelliği eklendi.
- Arama ve değiştirme özelliği için ön izleme adımı eklendi.
- Arama ve yükleme formlarındaki ayarların istemci tarafında tutulması sağlandı.
- Arama yetenekleri zenginleştirildi.
- Her projeye özel erişim denetimi yapılandırmasının ayrıntıları artırıldı.
- BASE_DIR varsayılan değeri değiştirildi.
- Yanlışlıkla silmeyi önlemek için hesap silme işleminin iki adımda yapılması sağlandı.
- Proje erişim denetimi ayarlarını düzenleme özelliği eklendi.
- Öneriler için Akismet ile istenmeyen içerikleri engelleme özelliği eklendi.

4.54.7 Weblate 2.16

11 Ağustos 2017 tarihinde yayınlandı.

- Çeşitli başarımlar iyileştirmeleri yapıldı.
- İç içe geçmiş JSON biçimi desteği eklendi.
- WebExtension JSON biçimi desteği eklendi.
- Git dışı aktarıcıda yaşanan kimlik doğrulaması sorunu düzeltildi.
- Belirli durumlar için CSV içe aktarma işlemi iyileştirildi.
- Diğer çeviriler pano bileşeni görünümü iyileştirildi.
- En fazla uzunluk denetimlerinin formda metnin uzunluğunu dayatması sağlandı.
- commit_pending yaşının her bileşene özel yapılandırılabilmesi sağlandı.
- Kullanıcı arayüzünde bazı sadeleştirmeler yapıldı.
- Çevirilerin bileşen/proje/site genelinde aramasında yaşanan sorun düzeltildi.

4.54.8 Weblate 2.15

30 Haziran 2017 tarihinde yayınlandı.

- Diğer çevirilerden daha fazla ilgili çevirinin görüntülenmesi sağlandı.
- Geçerli dizgenin diğer dillerdeki çevirilerini görüntüleme özelliği eklendi.
- Litvanyaca için varsayılan olarak 4 çoğul biçimin kullanılması sağlandı.
- Farklı biçimdeki tek dilli dosyalarda yaşanan yükleme sorunu düzeltildi.
- Başarısız olan kimlik doğrulaması hata iletileri iyileştirildi.
- Sözcüğü sözlükten silerken sayfa durumunun korunması sağlandı.
- İkincil dil çevirisini düzenlemek için doğrudan bağlantı eklendi.
- Perl biçimi kalite denetimi eklendi.
- Yeniden kullanılan parolaları reddetme özelliği eklendi.
- Araç çubuğu sağdan sola yazılan dilleri düzenlemek için zenginleştirildi.

4.54.9 Weblate 2.14.1

24 Mayıs 2017 tarihinde yayınlandı.

- Arama sonuçları sayfalandırılırken yaşanabilecek sorun düzeltildi.
- Bazı uç durumlarda yaşanan eski sürümlerden aktarım sorunu düzeltildi.
- Proje izleme ve izlemeyi bırakma işlemlerinde yaşanabilecek CSRF sorunu düzeltildi.
- Parola sıfırlama işleminde kullanıcı kimliğinin doğrulanması özelliği kaldırıldı.
- Parola unutulduğunda yaşanabilecek CAPTCHA atlama sorunu düzeltildi.

4.54.10 Weblate 2.14

17 Mayıs 2017 tarihinde yayınlandı.

- Sözlük kayıtlarının eklenmesinde AJAX kullanıldı.
- Oturumu kapatma işleminde CSRF engellemesi için POST kullanımına geçildi.
- API anahtarı belirtecini sıfırlama işleminde CSRF engellemesi için POST kullanımına geçildi.
- Weblate tarafından, Content-Security-Policy ayarının varsayılan olarak yapılması sağlandı.
- Yerel düzenleyici adresinin, kendi kendine XSS tehlikesini önlemek için doğrulanması sağlandı.
- Parolanın varsayılan olarak bilinen sorunlara karşı doğrulanması sağlandı.
- Kullanıcıların, hesaplarında yapılan parola değişikliği gibi önemli işlemlerle ilgili bilgilendirilmesi sağlandı.
- CSV dışı aktarımlarında olabilecek formüllerden kaçınılması sağlandı.
- Güvenlikte bazı küçük iyileştirmeler yapıldı.
- Kimlik doğrulama girişimlerine hız sınırlama özelliği eklendi.
- Geçmişe öneri içeriklerinin kaydedilmesi sağlandı.
- Önemli hesap işlemlerinin denetim günlüğüne kaydedilmesi sağlandı.
- Hesap silme ya da ilişkilendirme işlemlerinde parola onayı istenmesi özelliği eklendi.
- Önerinin yapıldığı zamanın görüntülenmesi sağlandı.
- Sondaki noktalı virgül karakteri kalite denetimi eklendi.

- Arama bağlantılarının paylaşılabildiğinden emin olunması sağlandı.
- API üzerine kaynak dizge bilgileri ve ekran görüntüleri eklendi.
- API yüklemesi ile çevirilerin üzerine yazma özelliği eklendi.

4.54.11 Weblate 2.13.1

12 Nisan 2017 tarihinde yayınlandı.

- Yönetilen projelerin kullanıcı profilinde listelenmesinde yaşanan sorun düzeltildi.
- Bazı izinlerin eksik olmasına neden olan aktarım sorunu düzeltildi.
- Çevirileri indirirken geçerli dosya biçiminin listelenmesinde yaşanan sorun düzeltildi.
- Kullanıcı izinlerinin yetersiz olduğu bir projeye erişmeye çalışıldığında HTTP 404 sayfasının görüntülenmesi sağlandı.

4.54.12 Weblate 2.13

12 Nisan 2017 tarihinde yayınlandı.

- Çeviri kalıplarında yaşanan kalite denetimleri sorunu düzeltildi.
- Çeviri kaybında tetiklenen kalite denetimi eklendi.
- Bekleyen kullanıcı önerilerini görüntüleme seçeneği eklendi.
- Bileşen listelerini kendiliğinden oluşturma seçeneği eklendi.
- Kimliğini doğrulamamış kullanıcılara görüntülenecek varsayılan panonun yapılandırılabilmesi sağlandı.
- Onaylama işlemi için 25 rastgele dizgeye göz atma seçeneği eklendi.
- Geçmişe dizge değişikliği işlemlerinin kaydedilmesi sağlandı.
- Yeni çeviri ekleme sırasındaki hata bildirimleri iyileştirildi.
- Proje içindeki her dile özel arama özelliği eklendi.
- Grup erişim denetimi listeleri belirli izinlerle sınırlandırıldı.
- Her projeye özel ayrı erişim denetimi listesi için grup erişim denetimi listesinin kullanılması sağlandı.
- İzin denetiminin daha ayrıntılı yapılması sağlandı.
- Kullanıcı arayüzünde bazı küçük iyileştirmeler yapıldı.

4.54.13 Weblate 2.12

3 Mart 2017 tarihinde yayınlandı.

- Grup yönetimi arayüzü iyileştirildi.
- Yandex Translate API desteği eklendi.
- Site genelinde arama hızı iyileştirildi.
- Proje ve bileşen kapsamında arama özelliği eklendi.
- Proje ve bileşen kapsamında arama ve değiştirme özelliği eklendi.
- Tutsuz çevirilerin görüntülenmesi iyileştirildi.
- Kaynak dosyaları yerel düzenleyicide açma desteği eklendi.
- Görsel klavyeyi özel karakterlerle yapılandırma desteği eklendi.

- Kaynak dizgeleri eşleştirmek için görsel karakter tanıma (OCR) desteği ile ekran görüntüsü yönetimi iyileştirildi.
- Varsayılan işleme iletisine çeviri bilgileri ve adresi eklendi.
- Joomla! çeviri biçimi desteği eklendi.
- Farklı dosya biçimleri için içe aktarma güvenilirliği iyileştirildi.

4.54.14 Weblate 2.11

31 Ocak 2017 tarihinde yayınlandı.

- Dil sayfasına ayrıntılı dil bilgileri eklendi.
- Mercurial arka ucunda iyileştirmeler yapıldı.
- Çeviri bileşeni önceliğini belirtme özelliği eklendi.
- Daha az kullanılan izinlerle bile Grup erişim denetiminin daha tutarlı kullanılması sağlandı.
- Kanca betiklerine WL_BRANCH değişkeni eklendi.
- Geliştirici belgeleri iyileştirildi.
- Git dışı aktarma eklentisinde çeşitli Git sürümleriyle daha iyi uyumluluk sağlandı.
- Her proje ve her bileşen için ayrı istatistikler özelliği eklendi.
- Microsoft Translate API desteğine iyileştirmek için dil kodu eşleme özelliği eklendi.
- Çeviri silme işleminin daha hızlı yapılabilmesi için tam metin temizleme işlemi arka plan görevine taşındı.
- Tek çoğul biçimi olan dillerde çoğul kaynağın görüntülenme sorunu düzeltildi.
- import_project komutunda hata işleme iyileştirildi.
- Çeşitli başarım iyileştirmeleri yapıldı.

4.54.15 Weblate 2.10.1

20 Ocak 2017 tarihinde yayınlandı.

- Parola sıfırlama formundan hesabın varlığı bilgisinin sızdırılması engellendi (CVE-2017-5537).

4.54.16 Weblate 2.10

15 Aralık 2016 tarihinde yayınlandı.

- Çoğul biçimlerin aynı olup olmadığına bakan kalite denetimi eklendi.
- Kimlik doğrulaması kullanan depolarda yaşanan GitHub kancaları sorunu düzeltildi.
- İsteğe bağlı Git dışı aktarma modülü eklendi.
- Microsoft Cognitive Services Translator API desteği eklendi.
- Proje ve bileşen kullanıcı arayüzü basitleştirildi.
- Kontrol karakterlerini kaldıran kendiliğinden düzeltme özelliği eklendi.
- Projelere her dilin özeti görünümü eklendi.
- CSV dışı aktarma özelliği eklendi.
- İstatistiklerin CSV olarak indirilmesi özelliği eklendi.
- Tüm çevrilere hızlı özet için matris görünümü eklendi.

- Değişiklikler ve dizgeler için temel API eklendi.
- Makine çevirileri için Apertium APy sunucusu desteği eklendi.

4.54.17 Weblate 2.9

4 Kasım 2016 tarihinde yayınlandı.

- createadmin yönetim komutu parametreleri zenginleştirildi.
- import_json komutu var olan bileşenleri işlemek için zenginleştirildi.
- YAML dosyaları desteği eklendi.
- Proje sahiplerinin çeviri bileşenini ve proje ayrıntılarını yapılandırması özelliği eklendi.
- Projeler için “Abone olma” terimi “İzleme” terimi ile değiştirildi.
- Projelerin doğrudan proje sayfasından izlenebilmesi sağlandı.
- Çok dilli durum pano bileşeni eklendi.
- Kaynak gösterilmiyorsa ikincil dilin vurgulanması sağlandı.
- Geçmişe öneri silme işlemlerinin kaydedilmesi sağlandı.
- Kullanıcı profilinde dil seçimi arayüzü iyileştirildi.
- Bileşen için beyaz tahta iletilerinin görüntülenmesinde yaşanan sorun düzeltildi.
- Kaydettikten sonra ayarlar sekmesinin seçili tutulması sağlandı.
- Kaynak dizge yorumunun daha belirgin görüntülenmesi sağlandı.
- Git depoları için Gettext PO birleştirme sürücüsünün kendiliğinden kurulması sağlandı.
- Arama ve değiştirme özelliği eklendi.
- Çeviriler için görsel bağlam (ekran görüntüleri) yükleme özelliği eklendi.

4.54.18 Weblate 2.8

31 Ağustos 2016 tarihinde yayınlandı.

- Belgelendirmede iyileştirmeler yapıldı.
- Çeviriler.
- Paketlenmiş JavaScript kitaplıkları güncellendi.
- list_translators yönetim komutu eklendi.
- Django 1.8 desteği eklendi.
- Django 1.10 uyumluluğunda yaşanan sorun düzeltildi.
- Subversion desteği eklendi.
- XML geçerlilik denetimi uyumsuz XML kod imlerinden ayrıldı.
- API, HIDE_REPO_CREDENTIALS ayarlarına uyacak biçimde düzeltildi.
- Zen kipinde kaynak değişikliğinin görüntülenmesi sağlandı.
- Alt + PageUp/PageDown/Home/End kısayol tuşlarının Zen kipinde de çalışması sağlandı.
- Değişikliklerin tam zamanını görüntüleyen ipucu eklendi.
- Çeviri sayfasına süzgeç seçme ve arama yapma seçeneği eklendi.
- Çeviri silme kullanıcı arayüzü eklendi.

- Yerleştirilebilir öğeleri ekleme davranışı iyileştirildi.
- Zen kipinde yaşanan kendiliğinden kilitleme sorunları düzeltildi.

4.54.19 Weblate 2.7

10 Temmuz 2016 tarihinde yayınlandı.

- Google internet çeviri makine çeviri hizmeti kaldırıldı.
- Çeviri eklenirken işleme iletisi iyileştirildi.
- İbranice için Google Çeviri API düzeltildi.
- Mercurial 3.8 ile uyumluluk sağlandı.
- import_json yönetim komutu eklendi.
- Listelenen çevirilerin doğru sıralanması sağlandı.
- Yalnızca bir farkın değil, tam öneri metninin görüntülenmesi sağlandı.
- API zenginleştirildi (ayrıntılı depo durumu, istatistikler, ...).
- Deneme paketindeki deneme depoları için ağ erişimi gereksinimi kaldırıldı.

4.54.20 Weblate 2.6

28 Nisan 2016 tarihinde yayınlandı.

- Dil süzgeci ile bileşenlerin doğrulanmasında yaşanan sorun düzeltildi.
- XLIFF dosyaları desteği iyileştirildi.
- İngilizce olmayan kaynakların makine çevirisi sorunu düzeltildi.
- REST API eklendi.
- Django 1.10 uyumluluğu sağlandı.
- Beyaz tahta iletilerine kategori özelliği eklendi.

4.54.21 Weblate 2.5

10 Mart 2016 tarihinde yayınlandı.

- Proje sahipleri için kendiliğinden çeviri düzeltildi.
- İşleme ve itme işlemlerinin başarımı iyileştirildi.
- Komut satırından öneriler eklemek için yeni yönetim komutu eklendi.
- Dosya yüklerken yorumları birleştirme özelliği eklendi.
- C printf biçimindeki bazı GNU eklentileri desteği eklendi.
- Belgelendirmede iyileştirmeler yapıldı.
- Çevirmen hesaplarını tutma özelliği eklendi.
- Katılımcı istatistikleri oluşturma özelliği eklendi.
- Site genelinde aramanın yalnızca bir dilde arama yapması sağlandı.
- Ermenice için kalite denetimleri iyileştirildi.
- Çeviriler var olmadan çeviri bileşenlerini başlatma özelliği eklendi.
- Qt TS ile yeni çeviriler ekleme özelliği eklendi.

- PHP dosyalarını çevirme desteği iyileştirildi.
- Kalite denetimlerinin başarımı iyileştirildi.
- Denetim hatalarının site genelinde aranması sorunu düzeltildi.
- Kaynak dili belirtme seçeneği eklendi.
- XLIFF dosyaları desteği iyileştirildi.
- import_project için seçenekler listesi zenginleştirildi.
- Beyaz tahta iletileri için hedefleme iyileştirildi.
- Projeler arasında kendiliğinden çeviri özelliği eklendi.
- Tam metin arama dizini iyileştirildi.
- Kendiliğinden çeviri için yönetim komutu eklendi.
- Yerleştirilebilir öğeleri vurgulama özelliği eklendi.
- Yerleştirilebilir öğeler, denetimler ve makine çevirileri için klavye kısayolları eklendi.
- Çeviri kitleme özelliği iyileştirildi.
- AngularJS ara değerlemesi için kalite denetimi eklendi.
- Kapsamlı grup tabanlı erişim denetimi listeleri eklendi.
- Düzenlenmesi gereken dizgeler (eski adıyla belirsiz) terimi açıklığa kavuşturuldu.
- İşlem yapılması gereken ve çevrilmemiş dizgelerle ilgili terimler açıklığa kavuşturuldu.
- Python 3 desteği eklendi.
- Django 1.7 desteği kaldırıldı.
- Yeni Gettext PO dosyaları oluşturmak için msginit bağımlılığı kaldırıldı.
- Yapılandırılabilir pano görünümleri eklendi.
- İşleme hatalarıyla ilgili bildirimler iyileştirildi.
- Aynı adlı bileşenleri import_project komutu ile içe aktarma seçeneği eklendi.
- PHP dosyalarını çevirme desteği iyileştirildi.
- Sözlüğü XLIFF olarak dışa aktarma özelliği eklendi.
- Tüm çevirileri XLIFF ve Gettext PO olarak dışa aktarma özelliği eklendi.
- Belgelendirmede iyileştirmeler yapıldı.
- Kendiliğinden grup atamalarını yapılandırma özelliği eklendi.
- Yeni çevirilerin eklenmesi iyileştirildi.

4.54.22 Weblate 2.4

20 Eylül 2015 tarihinde yayınlandı.

- PHP dosyaları desteği iyileştirildi.
- Anonim kullanıcıya erişim denetimi listesi ekleme özelliği eklendi.
- import_project komutunun yapılandırılabilirliği iyileştirildi.
- Geçmişin CSV dökümünü alma özelliği eklendi.
- Kopyalama/yapıştırma işlemlerinde boşluk karakteri hataları engellendi.
- Bitbucket internet kancaları desteği eklendi.
- Çeviri yüklerken belirsiz dizgeler üzerine uygulanan denetim sıkılaştırıldı.

- Birkaç adres değişti. Yer imlerinizi güncellemeniz gerekebilir.
- Kanca betiklerinin, geçerli klasör olarak sürüm denetimi sistemi kökü ile yürütülmesi sağlandı.
- Kanca betiklerinin, geçerli bileşeni tanımlayan ortam değişkenleriyle yürütülmesi sağlandı.
- Tam metin dizinini en iyi duruma getiren yönetim komutu eklendi.
- Rollbar üzerine hata bildirme desteği eklendi.
- Projelere birden fazla sahip olabilmesi özelliği eklendi.
- Proje sahiplerinin kendilerini yönetebilmesi özelliği eklendi.
- Gettext PO ile kullanılan `javascript-format` desteği eklendi.
- XLIFF olarak yeni çeviri ekleme özelliği eklendi.
- Dosya biçiminin kendiliğinden algılanması iyileştirildi.
- Klavye kısayolları zenginleştirildi.
- Birkaç dil için sözlük eşleştirmesi iyileştirildi.
- Sayfaların çoğunun düzeni iyileştirildi.
- Çeviri yaparken sözlüğe sözcük ekleme özelliği eklendi.
- Weblate tarafından yönetilen dilleri süzme özelliği eklendi.
- CSV dosyalarını çevirme ve içe aktarma özelliği eklendi.
- Durağan dosyaların işlenmesi yeniden yazıldı.
- Tek seçenek olduğunda, üçüncü taraf hizmetine doğrudan oturum / hesap açma bağlantılarının sunulması sağlandı.
- Hesap silinirken bekleyen değişikliklerin işlenmesi sağlandı.
- Site adını değiştiren yönetim komutu eklendi.
- Varsayılan işleyeni yapılandırma seçeneği eklendi.
- Yeni çeviri eklenmesinden sonra kancası eklendi.
- İşlemeye birden çok dosya eklemek için ekle seçeneği eklendi.

4.54.23 Weblate 2.3

22 Mayıs 2015 tarihinde yayınlandı.

- Django 1.6 ve South aktarımlarının desteği kaldırıldı.
- Java özellik dosyaları kullanılırken yeni çeviriler ekleme özelliği eklendi.
- Düzenleme yapmadan öneriyi onaylama özelliği eklendi.
- Google OAuth 2.0 desteği iyileştirildi.
- Microsoft .resx dosyaları desteği eklendi.
- Varsayılan robots.txt dosyası çevirilerin kapsamlı olarak taranmasını engelleyecek biçimde ayarlandı.
- Önerileri kabul etme iş akışı basitleştirildi.
- Önemli bildirimleri her zaman alacak proje sahipleri eklendi.
- Tek dilli kalıbın düzenlenmesini devre dışı bırakma özelliği eklendi..
- Depo durumu görünümünde ayrıntılar artırıldı.
- Çeviriyi değiştirirken kalıbı düzenlemek için doğrudan bağlantı eklendi.
- Proje sahiplerine daha fazla izin eklenmesi sağlandı.

- Zen kipinde ikincil dilin görüntülenmesi sağlandı.
- Kaynak dizgenin ikincil dil için gizlenmesi özelliği eklendi.

4.54.24 Weblate 2.2

19 Şubat 2015 tarihinde yayınlandı.

- Başarım iyileştirmeleri yapıldı.
- Konum ve yorum alanlarında tam metin arama özelliği eklendi.
- Yeni SVG/JavaScript temelli etkinlik çizelgeleri eklendi.
- Django 1.8 desteği eklendi.
- Yorumları silme özelliği eklendi.
- Sahip SVG simgesi eklendi.
- Google Analytics desteği eklendi.
- Çeviri dosyası adlarının işlenmesi iyileştirildi.
- Tek dilli JSON çevirileri desteği eklendi.
- Geçmişe bileşen kilitleme işlemlerinin kaydedilmesi sağlandı.
- Tek dilli çeviriler için kaynak (kalıp) dilini düzenleme özelliği eklendi.
- Temel Gerrit desteği eklendi.

4.54.25 Weblate 2.1

5 Aralık 2014 tarihinde yayınlandı.

- Mercurial depoları desteği eklendi.
- Glyphicon yazı tipi Awesome ile değiştirildi.
- Sosyal ağ kimlik doğrulama hizmetlerinin simgeleri eklendi.
- Düğme renklerinin ve simgelerinin tutarlılığı artırıldı.
- Belgelendirmede iyileştirmeler yapıldı.
- Çeşitli hata düzeltmeleri yapıldı.
- Küçük ekranlar için çeviri listesindeki sütunların kendiliğinden gizlenmesi sağlandı.
- Dosya sistemi yollarının yapılandırması değiştirildi.
- SSH anahtarlarının işlenmesi ve depolanması iyileştirildi.
- Depo kilitleme özelliği iyileştirildi.
- Her kaynak dizge için özelleştirilebilir kalite denetimleri eklendi.
- Tamamlanan çevirilerin panoda gizlenmesi sağlanması.

4.54.26 Weblate 2.0

6 Kasım 2014 tarihinde yayınlandı.

- Bootstrap kullanan yeni uyumlu kullanıcı arayüzü eklendi.
- Sürüm denetimi sistemi arka ucu yeniden yazıldı.
- Belgelendirmede iyileştirmeler yapıldı.
- Site genelindeki iletiler için beyaz tahta eklendi.
- Dizgelerin önceğinin yapılandırılabilmesi sağlandı.
- JSON dosya biçimi desteği eklendi.
- Bazı durumlarda mo dosyaları oluşturulurken yaşanan sorun düzeltildi.
- GitLab bildirimleri özelliği eklendi.
- Çeviri önerilerini devre dışı bırakma özelliği eklendi.
- Django 1.7 desteği eklendi.
- Erişim denetimi listesi projelerine kullanıcı yönetimi eklendi.
- Arama olanakları genişletildi.
- Çevirmenlere çoğul biçimler ile ilgili daha fazla ipucu verilmesi sağlandı.
- Git deposunun kilitlenmesi sorunu düzeltildi.
- Eski Git sürümleriyle uyumluluk sağlandı.
- Erişim denetimi listesi desteği geliştirildi.
- Her dil için tırnak ve diğer özel karakter düğmeleri eklendi.
- İstatistikleri JSONP olarak dışa aktarma özelliği eklendi.

4.55 Weblate 1.x serisi

4.55.1 Weblate 1.9

6 Mayıs 2014 tarihinde yayınlandı.

- Django 1.6 uyumluluğu sağlandı.
- Artık Django 1.4 uyumluluğu korunmuyor.
- Çeviriler için kilitleme ve kilit açma yönetim komutları eklendi.
- Qt TS dosyaları desteği iyileştirildi.
- Kullanıcıların hesaplarını silebilmesi sağlandı.
- Avatarları devre dışı bırakma özelliği eklendi.
- Ad ve soyad öznitelikleri birleştirildi.
- Avatarların sunucu tarafında tutulması ve ön belleğe alınması sağlandı.
- shields.io simgeleri desteği eklendi.

4.55.2 Weblate 1.8

7 Kasım 2013 tarihinde yayınlandı.

- Lütfen yükseltme yönergeleri için rehbera bakın.
- Proje özetinin görünümü güzelleştirildi.
- Paylaşım seçeneklerinin daha iyi görünmesi sağlandı.
- Anonim kullanıcı izinleri üzerinde daha fazla denetim sağlandı.
- Üçüncü taraf hizmetlerini kullanarak oturum açma desteği eklendi. Bilgi almak için rehbera bakabilirsiniz.
- Kullanıcı adı yerine e-posta ile oturum açılabilmesi sağlandı.
- Belgelendirmede iyileştirmeler yapıldı.
- Kaynak dizge onaylama işlemi geliştirildi.
- Tüm dizgelerde arama özelliği eklendi.
- Kaynak dizgelerin izlenmesi iyileştirildi.
- Hesap açılışına Captcha koruması eklendi.

4.55.3 Weblate 1.7

7 Ekim 2013 tarihinde yayınlandı.

- Lütfen yükseltme yönergeleri için rehbera bakın.
- Python parantezi biçim dizgesini denetleme desteği eklendi.
- Kalite denetimlerinin her bileşen için ayrı özelleştirilmesi sağlandı.
- Çeviri istatistiklerine göre ayrıntılar eklendi.
- Öneri, denetim ve yorumların dizgelerle ilişkilendirme yöntemi değiştirildi.
- Kullanıcıların işleme iletisine metin ekleyebilmesi sağlandı.
- Yeni dil isteklerine abone olma özelliği eklendi.
- Yeni çeviriler ekleme özelliği eklendi.
- Pano bileşenleri ve çizelgelerde Pango + Cairo yerine Pillow kullanılması sağlandı.
- Durum simgesi pano bileşeni eklendi.
- Geçersiz metin yönü denetimi kaldırıldı.
- Sözlükte yapılan değişiklikler geçmiş kayıtlarına eklendi.
- Çeviri görünümü başarıımı iyileştirildi.

4.55.4 Weblate 1.6

25 Temmuz 2013 tarihinde yayınlandı.

- Hesap açarken çıkabilecek sorunların işlenmesi iyileştirildi.
- Değişikliklere göz atma özelliği eklendi.
- Makine çevirisi önerilerinin sıralanması düzeltildi.
- MyMemory makine çevirisi desteği iyileştirildi.
- Amagama makine çevirisi desteği eklendi.
- Sık kullanılan sayfalarda çeşitli iyileştirmeler yapıldı.

- Arama sonuçlarında aranan ifadenin vurgulanması sağlandı.
- İleti kaydedilirken kendiliğinden düzeltme özelliği eklendi.
- Çeviri geçmişini izleme ve geri döndürme özellikleri eklendi.
- Google Çeviri API desteği eklendi.
- SSH sunucu anahtarları yönetimi desteği eklendi.
- Çeşitli form doğrulama iyileştirmeleri yapıldı.
- Çeşitli kalite denetimi iyileştirmeleri yapıldı.
- İçerik aktarma başarımı iyileştirildi.
- Önerilere oy verme desteği eklendi.
- Yönetim arayüzü sadeleştirildi.

4.55.5 Weblate 1.5

16 Nisan 2013 tarihinde yayınlandı.

- Lütfen yükseltme yönergeleri için rehbera bakın.
- Herkese açık kullanıcı sayfaları eklendi.
- Çoğul biçimlerin adlandırılması iyileştirildi.
- Sözlüğün TBX olarak dışa aktarılması desteği eklendi.
- Bitbucket bildirimleri desteği eklendi.
- Etkinlik çizelgelerinin her çeviri, dil ya da kullanıcı için görülebilmesi sağlandı.
- `import_project` yönetici komutunun seçenekleri zenginleştirildi.
- Django 1.5 uyumluluğu sağlandı.
- Avatarların libavatar kullanılarak görüntülenmesi sağlandı.
- JSON dışa aktarımının güzel biçimde yazdırılabilmesi sağlandı.
- Çeşitli başarımlar iyileştirmeleri yapıldı.
- Başarısız olan denetimlerin veya belirsiz dizgelerin projelerin veya dillerin ilerleme çubuklarında görüntülenmesi sağlandı.
- Özel ön işleme kancaları ve ek dosyaların işlenmesi özellikleri eklendi.
- Arama özelliği daha iyi başarımlar ve kullanıcı deneyimi sağlamak için yeniden yazıldı.
- Makine çevirileri arayüzü yenilendi.
- Tek dilli po dosyaları desteği eklendi.
- Çeşitli aramaların hızını artırmak için ön belleğe alınmış üst veri miktarı artırıldı.
- Sözcük sayılarının görüntülenmesi sağlandı.

4.55.6 Weblate 1.4

23 Ocak 2013 tarihinde yayınlandı.

- Dizge silme işlemiyle ilgili denetimlerin ve yorumların silinmesinde yaşanan sorun düzeltildi.
- Çevirilerin kendiliğinden yayılmasını engelleme seçeneği eklendi.
- Birleştirme hatalarına abone olma seçeneği eklendi.
- Özel tkit yükleyiciye gerek duyan projelerin doğru biçimde içe aktarılması sağlandı.
- Arama motoru robotlarının erişimini kolaylaştırmak için site haritaları eklendi.
- Bildirim e-postalarına veya özet akışlarına doğrudan dizge bağlantısı eklendi.
- Yönetim arayüzünde çeşitli iyileştirmeler yapıldı.
- Yönetim arayüzünde üretim ortamı kurulumu için ipuçları sunuldu.
- Her dil için ayrı pano bileşeni ve etkileşim sayfası eklendi.
- Çeviri kilitleme işlemi iyileştirildi.
- Pano bileşenleri için daha fazla kod parçası çeşidinin görüntülenmesi sağlandı.
- Başarısız olan denetimlerin veya belirsiz dizgelerin ilerleme çubuklarında belirtilmesi sağlandı.
- İşleme iletisine daha fazla biçimlendirme seçeneği eklendi.
- Makine çevirisi hizmetleriyle ilgili hataların işlenmesinde yaşanan sorun düzeltildi.
- Kendiliğinden çeviri kilitleme davranışı iyileştirildi.
- Önceki kaynak dizgeye göre yapılan değişikliklerin görüntülenmesi sağlandı.
- Alt dizge arama özelliği eklendi.
- Çeşitli kalite denetimi iyileştirmeleri yapıldı.
- Her proje için erişim denetimi listesi özelliği eklendi.
- Birim sınamalarına göre temel kod kapsamı.

4.55.7 Weblate 1.3

16 Kasım 2012 tarihinde yayınlandı.

- PostgreSQL veri tabanı arka ucuyla uyumluluk sağlandı.
- Yukarı akış Git deposunda silinen dillerin kaldırılması sağlandı.
- Kalite denetimlerinin işlenmesi iyileştirildi.
- Yeni denetimler eklendi (BBCode, XML biçimlendirmesi ve yeni satır karakterleri).
- Birleştirme yerine isteğe bağlı yeniden temellendirme desteği sağlandı.
- Weblate kurulumunun yerinin değiştirilmesi sağlandı (örneğin /weblate yolu altında çalıştırmak için).
- Kendiliğinden algılamının başarısız olması durumunda dosya türünü el ile seçme özelliği eklendi.
- Android kaynakları için daha iyi destek sağlandı.
- Web arayüzünden SSH anahtarı oluşturma özelliği eklendi.
- Veri dışa aktarımlarının daha görünür olması sağlandı.
- Bazı özel karakterleri yazmak için yeni düğmeler eklendi.
- Sözlüğü dışa aktarma özelliği eklendi.
- Tüm Weblate kurulumunu kilitleme özelliği eklendi.

- Kaynak dizge denetimleri ve kaynak dizge onaylama özellikleri eklendi.
- Hem çeviriler hem de kaynak dizgeler için kullanıcı yorumları özelliği eklendi.
- Günlük izleme iyileştirildi.
- Değişikliklerin RSS ile izlenebilmesi sağlandı.
- Sağdan sola yazılan dillerin desteği iyileştirildi.

4.55.8 Weblate 1.2

14 Ağustos 2012 tarihinde yayınlandı.

- Weblate veri tabanı aktarımı için South kullanımına geçildi. Yükseltme yapıyorsanız lütfen yükseltme yönergelerine bakın.
- Bağlı Git depolarıyla ilgili yaşanan küçük sorunlar düzeltildi.
- Giriş sayfası Weblate kullanarak çeviri yapan kişilerin ilgisini çekecek biçimde yenilendi.
- Çeviri projelerini tanıtmak için kullanılabilecek pano bileşenleri eklendi.
- Depoyu özgün durumuna sıfırlama seçeneği eklendi (izin verilmiş kullanıcılar için).
- Proje ve bileşenlere çevirileri kilitleme özelliği eklendi.
- Bazı çevirilerin devre dışı bırakılabilmesi sağlandı.
- Yeni çevirilerin eklenmesine yapılandırılabilir seçenekler eklendi.
- Her proje için Git işlemlerini yapılandırma özelliği eklendi.
- Basit antispam koruması eklendi.
- Ana sayfanın düzeni iyileştirildi.
- Her işlemede değişikliklerin kendiliğinden itilmesi sağlandı.
- Çevirmenlere e-posta bildirimleri gönderme özelliği eklendi.
- Ayarlarda yalnızca kullanılan dillerin listelenmesi sağlandı.
- Proje içe aktarılırken bilinmeyen dillerin işlenmesi iyileştirildi.
- Çevirmen tarafından çeviriyi kilitleme özelliği eklendi.
- İsteğe bağlı olarak po dosyasında "Language-Team" üst bilgisinin korunması sağlandı.
- Hakkında sayfasına bazı istatistikler eklendi.
- Django-registration 0.8 sürümü desteği eklendi (ve zorunlu kılındı).
- Denetim hatası olan dizge sayısının ön belleğe alınması sağlandı.
- Kurulum sırasında gereksinimlerin denetlenmesi sağlandı.
- Belgelendirmede iyileştirmeler yapıldı.

4.55.9 Weblate 1.1

4 Temmuz 2012 tarihinde yayınlandı.

- Birkaç çeviri iyileştirildi.
- Bileşen oluştururken doğrulama iyileştirildi.
- Bileşenler arasında paylaşılan Git depoları desteği eklendi.
- Uzak depoyu çekmek için her girişimin işlenmesinin gerekmemesi sağlandı.
- Dizin oluşturma yükünü aktarma desteği eklendi.

4.55.10 Weblate 1.0

10 Mayıs 2012 tarihinde yayınlandı.

- Bileşen eklenirken ve kaydedilirken doğrulama iyileştirildi.
- Android bileşen dosyaları için deneysel destek eklendi (yamanmış ttkit gerekir).
- Kancalardan gelen güncellemelerin arka planda yapılması sağlandı.
- Yükleme yönergeleri iyileştirildi.
- Sözlükte gezinme iyileştirildi.

4.56 Weblate 0.x serisi

4.56.1 Weblate 0.9

18 Nisan 2012 tarihinde yayınlandı.

- Bilinmeyen diller içe aktarılırken yaşanan sorun düzeltildi.
- Yakındaki iletilerin listelenmesi iyileştirildi.
- Birkaç denetim iyileştirildi.
- Belgeler güncellendi.
- Birkaç dilin tanımları eklendi.
- Çeşitli kod temizlemeleri yapıldı.
- Belgelendirmede iyileştirmeler yapıldı.
- Dosya düzeni değiştirildi.
- Yardım betikleri Django 1.4 sürümüne güncellendi.
- Çeviri sırasında gezinme iyileştirildi.
- PO dosyası ad değişikliklerinin işlenmesi iyileştirildi.
- Bileşen oluştururken doğrulama iyileştirildi.
- syncdb ile bütünleşik tam kurulum sağlandı.
- Tüm çeviri sayfalarına yapılan son değişikliklerin listesi eklendi.
- Çevrilmemiş dizge denetiminin, yalnızca biçim dizgesi iletilerini yok sayması sağlandı.

4.56.2 Weblate 0.8

3 Nisan 2012 tarihinde yayınlandı.

- Tam metin araması Whoosh ile değiştirildi.
- Denetimlerde çeşitli düzeltme ve iyileştirmeler yapıldı.
- Yeni updatechecks komutu eklendi.
- Çok sayıda çeviri güncellemesi yapıldı.
- Sık kullanılan terimlerin kaydedilebilmesi için sözlük eklendi.
- Depoların durumu özeti için /admin/report/ eklendi.
- Makine çevirisi hizmetlerinin sayfa yüklemesini engellemesi önlendi.
- Yönetim arayüzüne yararlı verileri güncelleme işlemleri eklendi.
- Kullanıcılar tarafından yapılan değişikliklerin günlük kayıtlarının tutulması sağlandı.
- Tek bir kullanıcıdan daha az işleme oluşturmak için Git işlemlerini erteleme özelliği eklendi.
- Başarısız olan denetimlere göz atma özelliği eklendi.
- Önceden çevrilmiş dizgeleri kullanan kendiliğinden çeviri özelliği eklendi.
- Kullanılan sürümleri görüntüleyen, yeni Hakkında sayfası eklendi.
- Django 1.4 uyumluluğu sağlandı.
- Değişiklikleri site arayüzünden uzak depoya itme özelliği eklendi.
- Başkaları tarafından yapılmış çevirileri gözden geçirme özelliği eklendi.

4.56.3 Weblate 0.7

16 Şubat 2012 tarihinde yayınlandı.

- GitHub bildirimlerinin doğrudan desteklenmesi sağlandı.
- Sahipsiz denetim ve çevirileri temizleme desteği eklendi.
- Çeviri sırasında yakındaki dizgeleri görüntüleme özelliği eklendi.
- Çeviri sırasında benzer dizgeleri görüntüleme özelliği eklendi.
- Dizge araması iyileştirildi.

4.56.4 Weblate 0.6

14 Şubat 2012 tarihinde yayınlandı.

- Çevrilmiş iletiler için çeşitli denetimler eklendi.
- Ayarlanabilir erişim denetimi özelliği eklendi.
- Yeni satır karakterleri bulunan çevirilerin işlenmesi iyileştirildi.
- Tabloların istemci tarafında sıralanması özelliği eklendi.
- Yükseltme yapıyorsanız lütfen yükseltme yönergelerini inceleyin.

4.56.5 Weblate 0.5

12 Şubat 2012 tarihinde yayınlandı.

- **Şu çevrimiçi hizmetleri kullanan makine çevirisi desteği eklendi:**
 - Apertium
 - Microsoft Translator
 - MyMemory
- Birkaç yeni çeviri.
- Yukarı akış değişikliklerinin birleştirilmesi iyileştirildi.
- Eşzamanlı Git çekme ve çevirilerinin işlenmesi iyileştirildi.
- Belirsiz değişikliklerin de yayılması sağlandı.
- Yüklenen dosyalardaki çevirilerin de yayılması sağlandı.
- FastCGI (ve büyük olasılıkla diğerleri) kullanılırken dosya indirmelerinde yaşanan sorun düzeltildi.

4.56.6 Weblate 0.4

8 Şubat 2012 tarihinde yayınlandı.

- Belgelere kullanım rehberi eklendi.
- API kancaları CSRF koruması gerektirmeyecek biçimde düzeltildi.

4.56.7 Weblate 0.3

8 Şubat 2012 tarihinde yayınlandı.

- Çoğul çevirilerde kaynağın daha iyi görüntülenmesi sağlandı.
- Sphinxs biçiminde yeni belgeler hazırlandı.
- Çeviri yapılırken ikincil dillerin görüntülenmesi sağlandı.
- Hata sayfası var olan projelerin listesini verecek biçimde geliştirildi.
- Her dil için istatistikler eklendi.

4.56.8 Weblate 0.2

7 Şubat 2012 tarihinde yayınlandı.

- Çeşitli formların doğrulaması iyileştirildi.
- Profili yükseltildiğinde kullanıcıların uyarılması sağlandı.
- Oturum açma adresinin anımsanması sağlandı.
- Çoğul biçimler yazılırken metin aralarının adlandırılması sağlandı.
- Çeviri alanının kendiliğinden genişletilmesi sağlandı.

4.56.9 Weblate 0.1

6 Şubat 2012 tarihinde yayınlandı.

- İlk sürüm.

W

wlc, [165](#)
wlc.config, [165](#)
wlc.main, [166](#)

HTTP Routing Table

/	GET /api/components/(string:project)/(string:component:slug)/, 140
ANY /, 112	GET /api/components/(string:project)/(string:component:slug)/, 136
/api	GET /api/components/(string:project)/(string:component:slug)/, 138
GET /api/, 114	GET /api/components/(string:project)/(string:component:slug)/, 138
/api/addons	GET /api/components/(string:project)/(string:component:slug)/, 137
GET /api/addons/, 152	GET /api/components/(string:project)/(string:component:slug)/, 136
GET /api/addons/(int:id)/, 152	GET /api/components/(string:project)/(string:component:slug)/, 140
PUT /api/addons/(int:id)/, 153	GET /api/components/(string:project)/(string:component:slug)/, 138
DELETE /api/addons/(int:id)/, 153	POST /api/components/(string:project)/(string:component:slug)/, 152
PATCH /api/addons/(int:id)/, 153	POST /api/components/(string:project)/(string:component:slug)/, 140
/api/changes	POST /api/components/(string:project)/(string:component:slug)/, 136
GET /api/changes/, 149	POST /api/components/(string:project)/(string:component:slug)/, 137
GET /api/changes/(int:id)/, 149	POST /api/components/(string:project)/(string:component:slug)/, 139
/api/component-lists	PUT /api/components/(string:project)/(string:component:slug)/, 135
GET /api/component-lists/, 153	DELETE /api/components/(string:project)/(string:component:slug)/, 135
GET /api/component-lists/(str:slug)/, 153	DELETE /api/components/(string:project)/(string:component:slug)/, 141
POST /api/component-lists/(str:slug)/components/, 154	PATCH /api/components/(string:project)/(string:component:slug)/, 133
PUT /api/component-lists/(str:slug)/, 153	/api/groups
DELETE /api/component-lists/(str:slug)/components/(string:component:slug)/, 154	GET /api/groups/, 118
PATCH /api/component-lists/(str:slug)/, 154	GET /api/groups/(int:id)/, 119
/api/components	POST /api/groups/, 118
GET /api/components/, 131	POST /api/groups/(int:id)/changes/, 121
GET /api/components/(string:project)/(string:component:slug)/, 131	POST /api/groups/(int:id)/components/, 120
GET /api/components/(string:project)/(string:component:slug)/, 135	
GET /api/components/(string:project)/(string:component:slug)/, 136	

POST /api/groups/(int:id)/languages/, 121 PUT /api/projects/(string:project)/, 126
 POST /api/groups/(int:id)/projects/, 121 DELETE /api/projects/(string:project)/, 126
 POST /api/groups/(int:id)/roles/, 120 PATCH /api/projects/(string:project)/, 126
 PUT /api/groups/(int:id)/, 120
 DELETE /api/groups/(int:id)/, 120
 DELETE /api/groups/(int:id)/componentlist/, 121
 DELETE /api/groups/(int:id)/components/(int:component_id)/, 120
 DELETE /api/groups/(int:id)/languages/(string:language_code_id)/, 121
 DELETE /api/groups/(int:id)/projects/(string:project_id)/, 121
 PATCH /api/groups/(int:id)/, 120

/api/roles

GET /api/roles/, 122
 POST /api/roles/, 122
 DELETE /api/roles/(int:id)/, 122
 PATCH /api/roles/(int:id)/, 122

/api/screenshots

GET /api/screenshots/, 150
 GET /api/screenshots/(int:id)/, 150
 GET /api/screenshots/(int:id)/file/, 150
 POST /api/screenshots/, 151
 POST /api/screenshots/(int:id)/file/, 150
 POST /api/screenshots/(int:id)/units/, 150
 PUT /api/screenshots/(int:id)/, 152
 DELETE /api/screenshots/(int:id)/, 152
 DELETE /api/screenshots/(int:id)/units/(int:unit_id)/, 151
 PATCH /api/screenshots/(int:id)/, 151

/api/languages

GET /api/languages/, 123
 GET /api/languages/(string:language)/, 123
 GET /api/languages/(string:language)/statistics/, 124
 POST /api/languages/, 123
 PUT /api/languages/(string:language)/, 124
 DELETE /api/languages/(string:language)/, 124
 PATCH /api/languages/(string:language)/, 124

/api/memory

GET /api/memory/, 147
 DELETE /api/memory/(int:memory_object_id)/, 147

/api/metrics

GET /api/metrics/, 155

/api/projects

GET /api/projects/, 125
 GET /api/projects/(string:project)/, 125
 GET /api/projects/(string:project)/changes/, 126
 GET /api/projects/(string:project)/components/, 127
 GET /api/projects/(string:project)/languages/, 130
 GET /api/projects/(string:project)/repositories/, 126
 GET /api/projects/(string:project)/statistics/, 131
 POST /api/projects/, 125
 POST /api/projects/(string:project)/components/, 128
 POST /api/projects/(string:project)/repositories/, 127

/api/tasks

GET /api/tasks/, 155
 GET /api/tasks/(str:uuid)/, 155

/api/translations

GET /api/translations/, 141
 GET /api/translations/(string:project)/(string:component)/, 141
 GET /api/translations/(string:project)/(string:component)/, 143
 GET /api/translations/(string:project)/(string:component)/, 145
 GET /api/translations/(string:project)/(string:component)/, 146
 GET /api/translations/(string:project)/(string:component)/, 146
 GET /api/translations/(string:project)/(string:component)/, 144
 POST /api/translations/(string:project)/(string:component)/, 144
 POST /api/translations/(string:project)/(string:component)/, 145
 POST /api/translations/(string:project)/(string:component)/, 146
 POST /api/translations/(string:project)/(string:component)/, 144

DELETE /api/translations/(string:project)/(string:component)/(string:language)/, 143
 POST /hooks/pagure/, 157

/api/units

GET /api/units/, 147
 GET /api/units/(int:id)/, 147
 PUT /api/units/(int:id)/, 148
 DELETE /api/units/(int:id)/, 149
 PATCH /api/units/(int:id)/, 148

/api/users

GET /api/users/, 115
 GET /api/users/(str:username)/, 115
 GET /api/users/(str:username)/notifications/, 117
 GET /api/users/(str:username)/notifications/(int:subscription_id)/, 117
 GET /api/users/(str:username)/statistics/, 117
 POST /api/users/, 115
 POST /api/users/(str:username)/groups/, 117
 POST /api/users/(str:username)/notifications/, 117
 PUT /api/users/(str:username)/, 116
 PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 118
 DELETE /api/users/(str:username)/, 117
 DELETE /api/users/(str:username)/groups/, 117
 DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 118
 PATCH /api/users/(str:username)/, 116
 PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 118

/exports

GET /exports/rss/, 159
 GET /exports/rss/(string:project)/, 159
 GET /exports/rss/(string:project)/(string:component)/, 159
 GET /exports/rss/(string:project)/(string:component)/(string:language)/, 159
 GET /exports/rss/language/(string:language)/, 159
 GET /exports/stats/(string:project)/(string:component)/, 158

/hooks

GET /hooks/update/(string:project)/, 156
 GET /hooks/update/(string:project)/(string:component)/, 156
 POST /hooks/azure/, 157
 POST /hooks/bitbucket/, 156
 POST /hooks/gitea/, 157
 POST /hooks/gitee/, 157
 POST /hooks/github/, 156

Semboller

.XML resource file
 file format, [93](#)

--add
 auto_translate komut satırı seçeneği, [393](#)

--addon
 install_addon komut satırı seçeneği, [399](#)

--age
 commit_pending komut satırı seçeneği, [394](#)

--author
 add_suggestions komut satırı seçeneği, [392](#)

--author-email
 wlc komut satırı seçeneği, [163](#)

--author-name
 wlc komut satırı seçeneği, [163](#)

--base-file-template
 import_project komut satırı seçeneği, [397](#)

--check
 importusers komut satırı seçeneği, [399](#)

--config
 wlc komut satırı seçeneği, [161](#)

--config-section
 wlc komut satırı seçeneği, [161](#)

--configuration
 install_addon komut satırı seçeneği, [399](#)

--convert
 wlc komut satırı seçeneği, [162](#)

--email
 createadmin komut satırı seçeneği, [395](#)

--file-format
 import_project komut satırı seçeneği, [397](#)

--force
 loadpo komut satırı seçeneği, [400](#)

--force-commit
 pushgit komut satırı seçeneği, [401](#)

--format
 wlc komut satırı seçeneği, [161](#)

--fuzzy
 wlc komut satırı seçeneği, [163](#)

--ignore
 import_json komut satırı seçeneği, [396](#)

--inconsistent
 auto_translate komut satırı seçeneği, [393](#)

--input
 wlc komut satırı seçeneği, [162](#)

--key
 wlc komut satırı seçeneği, [161](#)

--lang
 loadpo komut satırı seçeneği, [400](#)

--language-code
 list_translators komut satırı seçeneği, [400](#)

--language-map
 import_memory komut satırı seçeneği, [396](#)

--language-regex
 import_project komut satırı seçeneği, [397](#)

--license
 import_project komut satırı seçeneği, [397](#)

--license-url
 import_project komut satırı seçeneği, [397](#)

--main-component
 import_json komut satırı seçeneği, [396](#)

--method
 import_project komut satırı seçeneği, [397](#)

--mode
 auto_translate komut satırı seçeneği, [393](#)

--mt

auto_translate komut satırı seçeneği, 393
 --name
 createadmin komut satırı seçeneği, 395
 --name-template
 import_project komut satırı seçeneği, 397
 --new-base-template
 import_project komut satırı seçeneği, 397
 --no-password
 createadmin komut satırı seçeneği, 395
 --no-privs-update
 setupgroups komut satırı seçeneği, 401
 --no-projects-update
 setupgroups komut satırı seçeneği, 401
 --no-update
 setuplang komut satırı seçeneği, 402
 --output
 wlc komut satırı seçeneği, 162
 --overwrite
 auto_translate komut satırı seçeneği, 393
 wlc komut satırı seçeneği, 162
 --password
 createadmin komut satırı seçeneği, 395
 --project
 import_json komut satırı seçeneği, 396
 --source
 auto_translate komut satırı seçeneği, 393
 --threshold
 auto_translate komut satırı seçeneği, 393
 --update
 createadmin komut satırı seçeneği, 395
 import_json komut satırı seçeneği, 396
 install_addon komut satırı seçeneği, 399
 --url
 wlc komut satırı seçeneği, 161
 --user
 auto_translate komut satırı seçeneği, 393
 --username
 createadmin komut satırı seçeneği, 395
 --vcs
 import_project komut satırı seçeneği, 397

A

add_suggestions
 weblate admin command, 392
 add_suggestions komut satırı seçeneği
 --author, 392
 ADMINS
 setting, 217
 AKISMET_API_KEY
 setting, 349
 ALLOWED_HOSTS
 setting, 217
 Android
 file format, 77
 ANONYMOUS_USER_NAME
 setting, 349
 API, 111, 159, 164
 Apple strings
 file format, 78
 ARB
 file format, 79
 AUDITLOG_EXPIRY
 setting, 350
 AUTH_LOCK_ATTEMPTS
 setting, 350
 AUTH_TOKEN_VALID
 setting, 351
 auto_translate
 weblate admin command, 393
 auto_translate komut satırı seçeneği
 --add, 393
 --inconsistent, 393
 --mode, 393
 --mt, 393
 --overwrite, 393
 --source, 393
 --threshold, 393
 --user, 393
 AUTO_UPDATE
 setting, 350
 AUTOFIX_LIST
 setting, 351
 AVATAR_URL_PREFIX
 setting, 350

B

BACKGROUND_TASKS
 setting, 352
 BaseAddon (*weblate.addons.base* içindeki sınıf), 437
 BASIC_LANGUAGES
 setting, 352
 BITBUCKETSERVER_CREDENTIALS
 setting, 360
 BORG_EXTRA_ARGS
 setting, 352

C

CACHE_DIR
 setting, 353

- `can_install()` (*weblate.addons.base.BaseAddon sınıf yöntemi*), 437
 - `CELERY_BACKUP_OPTIONS`, 172, 190
 - `CELERY_BEAT_OPTIONS`, 172, 190
 - `CELERY_MAIN_OPTIONS`, 172, 190
 - `CELERY_MEMORY_OPTIONS`, 172, 190
 - `CELERY_NOTIFY_OPTIONS`, 172, 190
 - `celery_queues`
 - `weblate admin command`, 393
 - `CELERY_TRANSLATE_OPTIONS`, 172, 190
 - `changes`
 - `wlc komut satırı seçeneği`, 162
 - `CHECK_LIST`
 - setting, 353
 - `checkgit`
 - `weblate admin command`, 393
 - `cleanup`
 - `wlc komut satırı seçeneği`, 162
 - `cleanup_ssh_keys`
 - `weblate admin command`, 394
 - `cleanuptrans`
 - `weblate admin command`, 394
 - `Comma separated values`
 - file format, 80
 - `Command (wlc.main içindeki sınıf)`, 166
 - `COMMENT_CLEANUP_DAYS`
 - setting, 354
 - `commit`
 - `wlc komut satırı seçeneği`, 161
 - `commit_pending`
 - `weblate admin command`, 394
 - `commit_pending komut satırı seçeneği`
 - `--age`, 394
 - `COMMIT_PENDING_HOURS`
 - setting, 354
 - `commitgit`
 - `weblate admin command`, 394
 - `configure()` (*weblate.addons.base.BaseAddon yöntemi*), 437
 - `CONTACT_FORM`
 - setting, 354
 - `createadmin`
 - `weblate admin command`, 395
 - `createadmin komut satırı seçeneği`
 - `--email`, 395
 - `--name`, 395
 - `--no-password`, 395
 - `--password`, 395
 - `--update`, 395
 - `--username`, 395
 - `CSP_CONNECT_SRC`
 - setting, 353
 - `CSP_FONT_SRC`
 - setting, 353
 - `CSP_IMG_SRC`
 - setting, 353
 - `CSP_SCRIPT_SRC`
 - setting, 353
 - `CSP_STYLE_SRC`
 - setting, 353
 - `CSV`
 - file format, 80
- ## D
- `daily()` (*weblate.addons.base.BaseAddon yöntemi*), 437
 - `DATA_DIR`
 - setting, 354
 - `DATABASE_BACKUP`
 - setting, 355
 - `DATABASES`
 - setting, 217
 - `DEBUG`
 - setting, 217
 - `DEFAULT_ACCESS_CONTROL`
 - setting, 355
 - `DEFAULT_ADD_MESSAGE`
 - setting, 356
 - `DEFAULT_ADDON_MESSAGE`
 - setting, 356
 - `DEFAULT_ADDONS`
 - setting, 356
 - `DEFAULT_AUTO_WATCH`
 - setting, 356
 - `DEFAULT_COMMIT_MESSAGE`
 - setting, 356
 - `DEFAULT_COMMITER_EMAIL`
 - setting, 357
 - `DEFAULT_COMMITER_NAME`
 - setting, 357
 - `DEFAULT_DELETE_MESSAGE`
 - setting, 356
 - `DEFAULT_FROM_EMAIL`
 - setting, 217
 - `DEFAULT_LANGUAGE`
 - setting, 357
 - `DEFAULT_MERGE_MESSAGE`
 - setting, 356
 - `DEFAULT_MERGE_STYLE`
 - setting, 357
 - `DEFAULT_PAGE_LIMIT`
 - setting, 365
 - `DEFAULT_PULL_MESSAGE`
 - setting, 358
 - `DEFAULT_RESTRICTED_COMPONENT`
 - setting, 356
 - `DEFAULT_SHARED_TM`
 - setting, 357
 - `DEFAULT_TRANSLATION_PROPAGATION`
 - setting, 358
 - `download`
 - `wlc komut satırı seçeneği`, 162
 - `DTD`
 - file format, 81
 - `dualgual`
 - translation, 102

dump_memory
weblate admin command, 395

dumpuserdata
weblate admin command, 395

E

ENABLE_AVATARS
setting, 358

ENABLE_HOOKS
setting, 358

ENABLE_HTTPS
setting, 358

ENABLE_SHARING
setting, 358

EXTRA_HTML_HEAD
setting, 359

F

file format
.XML resource file, 93
Android, 77
Apple strings, 78
ARB, 79
Comma separated values, 80
CSV, 80
DTD, 81
gettext, 83
go-i18n, 84
gotext, 85
GWT properties, 85
i18next, 86
INI translations, 87, 88
Java properties, 88
Joomla translations, 89
JSON, 89
mi18n lang, 91
PHP strings, 92
PO, 83
Qt, 92
RC, 97
ResourceDictionary, 93
RESX, 93
Ruby YAML, 93
Ruby YAML Ain't Markup Language, 93
string resources, 77
TS, 92
WPF, 93
XLIFF, 97
XML, 82
YAML, 99
YAML Ain't Markup Language, 99

G

get() (*wlc. Weblate yöntemi*), 165

get_add_form() (*weblate.addons.base.BaseAddon sınıf yöntemi*), 437

GET_HELP_URL
setting, 359

get_settings_form() (*weblate.addons.base.BaseAddon yöntemi*), 437

gettext
file format, 83

GITEA_CREDENTIALS
setting, 359

GITHUB_CREDENTIALS
setting, 360

GITLAB_CREDENTIALS
setting, 359

go-i18n
file format, 84

GOOGLE_ANALYTICS_ID
setting, 360

gotext
file format, 85

GWT properties
file format, 85

H

HIDE_REPO_CREDENTIALS
setting, 360

HIDE_VERSION
setting, 361

I

i18next
file format, 86

import_demo
weblate admin command, 395

import_json
weblate admin command, 396

import_json komut satırı seçeneği
--ignore, 396
--main-component, 396
--project, 396
--update, 396

import_memory
weblate admin command, 396

import_memory komut satırı seçeneği
--language-map, 396

import_project
weblate admin command, 397

import_project komut satırı seçeneği
--base-file-template, 397
--file-format, 397
--language-regex, 397
--license, 397
--license-url, 397
--main-component, 397
--name-template, 397
--new-base-template, 397
--vcs, 397

importuserdata
weblate admin command, 399

importusers
weblate admin command, 399

importusers komut satırı seçeneği

- check, 399
- INI translations
 - file format, 87, 88
- install_addon
 - weblate admin command, 399
- install_addon komut satırı seçeneği
 - addon, 399
 - configuration, 399
 - update, 399
- INTERLEDGER_PAYMENT_POINTERS
 - setting, 361
- IP_BEHIND_REVERSE_PROXY
 - setting, 361
- IP_PROXY_HEADER
 - setting, 361
- IP_PROXY_OFFSET
 - setting, 362

J

- Java properties
 - file format, 88
- Joomla translations
 - file format, 89
- JSON
 - file format, 89

L

- LEGAL_TOS_DATE
 - setting, 362
- LEGAL_URL
 - setting, 362
- LICENSE_EXTRA
 - setting, 363
- LICENSE_FILTER
 - setting, 363
- LICENSE_REQUIRED
 - setting, 363
- LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
 - setting, 363
- list_languages
 - weblate admin command, 399
- list_translators
 - weblate admin command, 400
- list_translators komut satırı seçeneği
 - language-code, 400
- list_versions
 - weblate admin command, 400
- list-components
 - wlc komut satırı seçeneği, 161
- list-languages
 - wlc komut satırı seçeneği, 161
- list-projects
 - wlc komut satırı seçeneği, 161
- list-translations
 - wlc komut satırı seçeneği, 161
- load() (*wlc.config.WeblateConfig yöntemi*), 165
- loadpo

- weblate admin command, 400
- loadpo komut satırı seçeneği
 - force, 400
 - lang, 400
- LOCALIZE_CDN_PATH
 - setting, 364
- LOCALIZE_CDN_URL
 - setting, 364
- lock
 - wlc komut satırı seçeneği, 162
- lock_translation
 - weblate admin command, 400
- lock-status
 - wlc komut satırı seçeneği, 162
- LOGIN_REQUIRED_URLS
 - setting, 364
- LOGIN_REQUIRED_URLS_EXCEPTIONS
 - setting, 364
- ls
 - wlc komut satırı seçeneği, 161

M

- main() (*wlc.main modülü içinde*), 166
- MATOMO_SITE_ID
 - setting, 365
- MATOMO_URL
 - setting, 365
- mi18n lang
 - file format, 91
- modülü
 - wlc, 165
 - wlc.config, 165
 - wlc.main, 166
- monolingual
 - translation, 102
- move_language
 - weblate admin command, 401

N

- NEARBY_MESSAGES
 - setting, 365

O

- ortam değişkeni
 - CELERY_BACKUP_OPTIONS, 172, 190
 - CELERY_BEAT_OPTIONS, 172, 190
 - CELERY_MAIN_OPTIONS, 172, 190
 - CELERY_MEMORY_OPTIONS, 172, 190
 - CELERY_NOTIFY_OPTIONS, 172, 190
 - CELERY_TRANSLATE_OPTIONS, 172, 190
 - CLIENT_MAX_BODY_SIZE, 180
 - POSTGRES_ALTER_ROLE, 185
 - POSTGRES_CONN_MAX_AGE, 185
 - POSTGRES_DATABASE, 185
 - POSTGRES_DISABLE_SERVER_SIDE_CURSORS, 185
 - POSTGRES_HOST, 185
 - POSTGRES_PASSWORD, 185

POSTGRES_PASSWORD_FILE, 185
 POSTGRES_PORT, 185
 POSTGRES_SSL_MODE, 185
 POSTGRES_USER, 185
 REDIS_DB, 186
 REDIS_HOST, 186
 REDIS_PASSWORD, 186
 REDIS_PASSWORD_FILE, 186
 REDIS_PORT, 186
 REDIS_TLS, 186
 REDIS_VERIFY_SSL, 186
 ROLLBAR_ENVIRONMENT, 188
 ROLLBAR_KEY, 188
 SENTRY_DSN, 189
 SENTRY_ENVIRONMENT, 189
 SOCIAL_AUTH_SLACK_SECRET, 184
 WEB_WORKERS, 172, 190
 WEBLATE_ADD_ADDONS, 189
 WEBLATE_ADD_APPS, 189
 WEBLATE_ADD_AUTOFIX, 189
 WEBLATE_ADD_CHECK, 189
 WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTED, 176
 WEBLATE_ADMIN_EMAIL, 172--174
 WEBLATE_ADMIN_NAME, 172--174
 WEBLATE_ADMIN_PASSWORD, 168, 172--174
 WEBLATE_ADMIN_PASSWORD_FILE, 174
 WEBLATE_AKISMET_API_KEY, 178, 409
 WEBLATE_ALLOWED_HOSTS, 174, 217, 221, 222, 371
 WEBLATE_API_RATELIMIT_ANON, 114, 179
 WEBLATE_API_RATELIMIT_USER, 114, 179
 WEBLATE_AUTH_LDAP_BIND_DN, 180
 WEBLATE_AUTH_LDAP_BIND_PASSWORD, 180
 WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE, 180
 WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFUSING, 180
 WEBLATE_AUTH_LDAP_SERVER_URI, 180
 WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 180
 WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 180
 WEBLATE_AUTH_LDAP_USER_SEARCH, 180
 WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 180
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 181
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 181
 WEBLATE_AUTO_UPDATE, 188
 WEBLATE_AVATAR_URL_PREFIX, 179
 WEBLATE_BASIC_LANGUAGES, 178
 WEBLATE_BITBUCKETSERVER_HOST, 177
 WEBLATE_BITBUCKETSERVER_TOKEN, 177
 WEBLATE_BITBUCKETSERVER_USERNAME, 177
 WEBLATE_BORG_EXTRA_ARGS, 179
 WEBLATE_CONTACT_FORM, 174
 WEBLATE_CORS_ALLOWED_ORIGINS, 180
 WEBLATE_CSP_CONNECT_SRC, 178
 WEBLATE_CSP_FONT_SRC, 178
 WEBLATE_CSP_IMG_SRC, 178
 WEBLATE_CSP_SCRIPT_SRC, 178
 WEBLATE_CSP_STYLE_SRC, 178
 WEBLATE_DATABASE_BACKUP, 186
 WEBLATE_DEBUG, 173, 241
 WEBLATE_DEFAULT_ACCESS_CONTROL, 177
 WEBLATE_DEFAULT_AUTO_WATCH, 178
 WEBLATE_DEFAULT_COMMITER_EMAIL, 177
 WEBLATE_DEFAULT_COMMITER_NAME, 177
 WEBLATE_DEFAULT_FROM_EMAIL, 174
 WEBLATE_DEFAULT_PULL_MESSAGE, 177
 WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 177
 WEBLATE_DEFAULT_SHARED_TM, 177
 WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 177
 WEBLATE_EMAIL_BACKEND, 188
 WEBLATE_EMAIL_HOST, 187
 WEBLATE_EMAIL_HOST_PASSWORD, 187
 WEBLATE_EMAIL_HOST_PASSWORD_FILE, 187
 WEBLATE_EMAIL_HOST_USER, 187
 WEBLATE_EMAIL_PORT, 187, 188
 WEBLATE_EMAIL_USE_SSL, 187, 188
 WEBLATE_EMAIL_USE_TLS, 187, 188
 WEBLATE_ENABLE_AVATARS, 179
 WEBLATE_ENABLE_HOOKS, 179
 WEBLATE_ENABLE_HTTPS, 175, 251
 WEBLATE_ENABLE_SHARING, 179
 WEBLATE_EXTRA_HTML_HEAD, 179
 WEBLATE_GET_HELP_URL, 188
 WEBLATE_GITEA_HOST, 177
 WEBLATE_GITEA_TOKEN, 177
 WEBLATE_GITEA_USERNAME, 177
 WEBLATE_GITHUB_HOST, 176, 241
 WEBLATE_GITHUB_TOKEN, 176
 WEBLATE_GITHUB_USERNAME, 176
 WEBLATE_GITLAB_HOST, 176
 WEBLATE_GITLAB_TOKEN, 176
 WEBLATE_GITLAB_USERNAME, 176
 WEBLATE_GOOGLE_ANALYTICS_ID, 176
 WEBLATE_GPG_IDENTITY, 178
 WEBLATE_HIDE_VERSION, 178
 WEBLATE_INTERLEDGER_PAYMENT_POINTERS, 178
 WEBLATE_IP_PROXY_HEADER, 175
 WEBLATE_LEGAL_URL, 188
 WEBLATE_LICENSE_FILTER, 178
 WEBLATE_LICENSE_REQUIRED, 178
 WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 179
 WEBLATE_LOCALIZE_CDN_PATH, 189
 WEBLATE_LOCALIZE_CDN_URL, 189

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS,	182	WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY,	182
176		WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET,	182
WEBLATE_LOGLEVEL,	173	WEBLATE_SOCIAL_AUTH_FEDORA,	184
WEBLATE_LOGLEVEL_DATABASE,	173	WEBLATE_SOCIAL_AUTH_GITEA_API_URL,	183
WEBLATE_NO_EMAIL_AUTH,	185	WEBLATE_SOCIAL_AUTH_GITEA_KEY,	183
WEBLATE_PAGURE_HOST,	177	WEBLATE_SOCIAL_AUTH_GITEA_SECRET,	183
WEBLATE_PAGURE_TOKEN,	177	WEBLATE_SOCIAL_AUTH_GITHUB_KEY,	182
WEBLATE_PAGURE_USERNAME,	177	WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY,	182
WEBLATE_PRIVACY_URL,	188	WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME,	182
WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN,	179	WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET,	182
WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE,	179	WEBLATE_SOCIAL_AUTH_GITHUB_SECRET,	182
WEBLATE_RATELIMIT_ATTEMPTS,	178, 411	WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID,	182
WEBLATE_RATELIMIT_LOCKOUT,	178	WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY,	182
WEBLATE_RATELIMIT_WINDOW,	178	WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET,	182
WEBLATE_REGISTRATION_ALLOW_BACKENDS,	174	WEBLATE_SOCIAL_AUTH_GITLAB_API_URL,	183
WEBLATE_REGISTRATION_OPEN,	174	WEBLATE_SOCIAL_AUTH_GITLAB_KEY,	183
WEBLATE_REGISTRATION_REBIND,	174	WEBLATE_SOCIAL_AUTH_GITLAB_SECRET,	183
WEBLATE_REMOVE_ADDONS,	189	WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY,	182
WEBLATE_REMOVE_APPS,	189	WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET,	182
WEBLATE_REMOVE_AUTOFIX,	189	WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED,	182
WEBLATE_REMOVE_CHECK,	189	WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED,	182
WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS,	176	WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL,	183
WEBLATE_REQUIRE_LOGIN,	176, 370	WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM,	183
WEBLATE_SAML_IDP_ENTITY_ID,	184	WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URI,	183
WEBLATE_SAML_IDP_IMAGE,	184	WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE,	183
WEBLATE_SAML_IDP_TITLE,	184	WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY,	183
WEBLATE_SAML_IDP_URL,	184	WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY,	183
WEBLATE_SAML_IDP_X509CERT,	184	WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET,	183
WEBLATE_SECURE_PROXY_SSL_HEADER,	175, 176	WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE,	183
WEBLATE_SERVER_EMAIL,	174	WEBLATE_SOCIAL_AUTH_OIDC_KEY,	184
WEBLATE_SERVICE,	172, 190	WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT,	184
WEBLATE_SILENCED_SYSTEM_CHECKS,	178, 248		
WEBLATE_SIMPLIFY_LANGUAGES,	177		
WEBLATE_SITE_DOMAIN,	173, 219, 237, 371		
WEBLATE_SITE_TITLE,	173		
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY,	183		
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET,	183		
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY,	183		
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET,	183		
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TITLE,	183		
WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY,	182		
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY,	182		
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET,	182		
WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET,			

WEBLATE_SOCIAL_AUTH_OIDC_SECRET, 184
 WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY, 184
 WEBLATE_SOCIAL_AUTH_OPENINFRA, 184
 WEBLATE_SOCIAL_AUTH_OPENSUSE, 184
 WEBLATE_SOCIAL_AUTH_SLACK_KEY, 184
 WEBLATE_SOCIAL_AUTH_UBUNTU, 184
 WEBLATE_SSH_EXTRA_ARGS, 179
 WEBLATE_STATUS_URL, 188
 WEBLATE_TIME_ZONE, 175
 WEBLATE_UNUSED_ALERT_DAYS, 179
 WEBLATE_URL_PREFIX, 178
 WEBLATE_WEBSITE_REQUIRED, 178
 WEBLATE_WORKERS, 172, 190
 WL_BRANCH, 346
 WL_COMPONENT_NAME, 346
 WL_COMPONENT_SLUG, 346
 WL_COMPONENT_URL, 347
 WL_ENGAGE_URL, 347
 WL_FILE_FORMAT, 346
 WL_FILEMASK, 346
 WL_LANGUAGE, 346
 WL_NEW_BASE, 346
 WL_PATH, 346
 WL_PREVIOUS_HEAD, 346
 WL_PROJECT_NAME, 347
 WL_PROJECT_SLUG, 346
 WL_REPO, 346
 WL_TEMPLATE, 346
 WL_VCS, 346

P

PAGURE_CREDENTIALS
 setting, 365
 PHP strings
 file format, 92
 PIWIK_SITE_ID
 setting, 365
 PIWIK_URL
 setting, 365
 PO
 file format, 83
 post() (*wlc. Weblate yöntemi*), 165
 post_add() (*weblate.addons.base.BaseAddon yöntemi*), 437
 post_commit() (*weblate.addons.base.BaseAddon yöntemi*), 437
 post_push() (*weblate.addons.base.BaseAddon yöntemi*), 437
 post_update() (*weblate.addons.base.BaseAddon yöntemi*), 437
 pre_commit() (*weblate.addons.base.BaseAddon yöntemi*), 438
 pre_push() (*weblate.addons.base.BaseAddon yöntemi*), 438
 pre_update() (*weblate.addons.base.BaseAddon yöntemi*), 438

PRIVACY_URL
 setting, 366
 PRIVATE_COMMIT_EMAIL_OPT_IN
 setting, 366
 PRIVATE_COMMIT_EMAIL_TEMPLATE
 setting, 366
 PROJECT_BACKUP_KEEP_COUNT
 setting, 366
 PROJECT_BACKUP_KEEP_DAYS
 setting, 367
 PROJECT_NAME_RESTRICT_RE
 setting, 367
 PROJECT_WEB_RESTRICT_HOST
 setting, 367
 PROJECT_WEB_RESTRICT_NUMERIC
 setting, 367
 PROJECT_WEB_RESTRICT_RE
 setting, 367
 pull
 wlc komut satırı seçeneği, 162
 push
 wlc komut satırı seçeneği, 162
 pushgit
 weblate admin command, 401
 pushgit komut satırı seçeneği
 --force-commit, 401
 Python, 164
 Python Geliştirme Önerileri
 PEP 484, 433, 448

Q

Qt
 file format, 92

R

RATELIMIT_ATTEMPTS
 setting, 368
 RATELIMIT_LOCKOUT
 setting, 368
 RATELIMIT_WINDOW
 setting, 368
 RC
 file format, 97
 REDIS_PASSWORD, 186
 register_command() (*wlc.main modülü içinde*), 166
 REGISTRATION_ALLOW_BACKENDS
 setting, 368
 REGISTRATION_CAPTCHA
 setting, 369
 REGISTRATION_EMAIL_MATCH
 setting, 369
 REGISTRATION_OPEN
 setting, 369
 REGISTRATION_REBIND
 setting, 369
 repo
 wlc komut satırı seçeneği, 162

REPOSITORY_ALERT_THRESHOLD
 setting, 369
REQUIRE_LOGIN
 setting, 370
reset
 wlc komut satırı seçeneği, 162
ResourceDictionary
 file format, 93
REST, 111
RESX
 file format, 93
RFC
 RFC 5646, 77
Ruby YAML
 file format, 93
Ruby YAML Ain't Markup Language
 file format, 93

S

save_state() (weblate.addons.base.BaseAddon
 yöntemi), 438
SECRET_KEY
 setting, 218
SENTRY_DSN
 setting, 370
SERVER_EMAIL
 setting, 218
SESSION_COOKIE_AGE_AUTHENTICATED
 setting, 370
SESSION_ENGINE
 setting, 217
setting
 ADMINS, 217
 AKISMET_API_KEY, 349
 ALLOWED_HOSTS, 217
 ANONYMOUS_USER_NAME, 349
 AUDITLOG_EXPIRY, 350
 AUTH_LOCK_ATTEMPTS, 350
 AUTH_TOKEN_VALID, 351
 AUTO_UPDATE, 350
 AUTOFIX_LIST, 351
 AVATAR_URL_PREFIX, 350
 BACKGROUND_TASKS, 352
 BASIC_LANGUAGES, 352
 BITBUCKETSERVER_CREDENTIALS, 360
 BORG_EXTRA_ARGS, 352
 CACHE_DIR, 353
 CHECK_LIST, 353
 COMMENT_CLEANUP_DAYS, 354
 COMMIT_PENDING_HOURS, 354
 CONTACT_FORM, 354
 CSP_CONNECT_SRC, 353
 CSP_FONT_SRC, 353
 CSP_IMG_SRC, 353
 CSP_SCRIPT_SRC, 353
 CSP_STYLE_SRC, 353
 DATA_DIR, 354
 DATABASE_BACKUP, 355

DATABASES, 217
DEBUG, 217
DEFAULT_ACCESS_CONTROL, 355
DEFAULT_ADD_MESSAGE, 356
DEFAULT_ADDON_MESSAGE, 356
DEFAULT_ADDONS, 356
DEFAULT_AUTO_WATCH, 356
DEFAULT_COMMIT_MESSAGE, 356
DEFAULT_COMMITTER_EMAIL, 357
DEFAULT_COMMITTER_NAME, 357
DEFAULT_DELETE_MESSAGE, 356
DEFAULT_FROM_EMAIL, 217
DEFAULT_LANGUAGE, 357
DEFAULT_MERGE_MESSAGE, 356
DEFAULT_MERGE_STYLE, 357
DEFAULT_PAGE_LIMIT, 365
DEFAULT_PULL_MESSAGE, 358
DEFAULT_RESTRICTED_COMPONENT, 356
DEFAULT_SHARED_TM, 357
DEFAULT_TRANSLATION_PROPAGATION,
 358
ENABLE_AVATARS, 358
ENABLE_HOOKS, 358
ENABLE_HTTPS, 358
ENABLE_SHARING, 358
EXTRA_HTML_HEAD, 359
GET_HELP_URL, 359
GITEA_CREDENTIALS, 359
GITHUB_CREDENTIALS, 360
GITLAB_CREDENTIALS, 359
GOOGLE_ANALYTICS_ID, 360
HIDE_REPO_CREDENTIALS, 360
HIDE_VERSION, 361
INTERLEDGER_PAYMENT_POINTERS, 361
IP_BEHIND_REVERSE_PROXY, 361
IP_PROXY_HEADER, 361
IP_PROXY_OFFSET, 362
LEGAL_TOS_DATE, 362
LEGAL_URL, 362
LICENSE_EXTRA, 363
LICENSE_FILTER, 363
LICENSE_REQUIRED, 363
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH,
 363
LOCALIZE_CDN_PATH, 364
LOCALIZE_CDN_URL, 364
LOGIN_REQUIRED_URLS, 364
LOGIN_REQUIRED_URLS_EXCEPTIONS, 364
MATOMO_SITE_ID, 365
MATOMO_URL, 365
NEARBY_MESSAGES, 365
PAGURE_CREDENTIALS, 365
PIWIK_SITE_ID, 365
PIWIK_URL, 365
PRIVACY_URL, 366
PRIVATE_COMMIT_EMAIL_OPT_IN, 366
PRIVATE_COMMIT_EMAIL_TEMPLATE, 366
PROJECT_BACKUP_KEEP_COUNT, 366

- PROJECT_BACKUP_KEEP_DAYS, 367
- PROJECT_NAME_RESTRICT_RE, 367
- PROJECT_WEB_RESTRICT_HOST, 367
- PROJECT_WEB_RESTRICT_NUMERIC, 367
- PROJECT_WEB_RESTRICT_RE, 367
- RATELIMIT_ATTEMPTS, 368
- RATELIMIT_LOCKOUT, 368
- RATELIMIT_WINDOW, 368
- REGISTRATION_ALLOW_BACKENDS, 368
- REGISTRATION_CAPTCHA, 369
- REGISTRATION_EMAIL_MATCH, 369
- REGISTRATION_OPEN, 369
- REGISTRATION_REBIND, 369
- REPOSITORY_ALERT_THRESHOLD, 369
- REQUIRE_LOGIN, 370
- SECRET_KEY, 218
- SENTRY_DSN, 370
- SERVER_EMAIL, 218
- SESSION_COOKIE_AGE_AUTHENTICATED, 370
- SESSION_ENGINE, 217
- SIMPLIFY_LANGUAGES, 370
- SINGLE_PROJECT, 371
- SITE_DOMAIN, 370
- SITE_TITLE, 371
- SPECIAL_CHARS, 371
- SSH_EXTRA_ARGS, 371
- STATUS_URL, 372
- SUGGESTION_CLEANUP_DAYS, 372
- UNUSED_ALERT_DAYS, 372
- UPDATE_LANGUAGES, 372
- URL_PREFIX, 372
- VCS_API_DELAY, 373
- VCS_BACKENDS, 373
- VCS_CLONE_DEPTH, 373
- WEBLATE_ADDONS, 374
- WEBLATE_EXPORTERS, 375
- WEBLATE_FORMATS, 375
- WEBLATE_GPG_IDENTITY, 375
- WEBLATE_MACHINERY, 375
- WEBSITE_REQUIRED, 375
- setupgroups
 - weblate admin command, 401
- setupgroups komut satırı seçeneği
 - no-privs-update, 401
 - no-projects-update, 401
- setuplang
 - weblate admin command, 402
- setuplang komut satırı seçeneği
 - no-update, 402
- show
 - wlc komut satırı seçeneği, 161
- SIMPLIFY_LANGUAGES
 - setting, 370
- SINGLE_PROJECT
 - setting, 371
- SITE_DOMAIN
 - setting, 370
- SITE_TITLE
 - setting, 371
- SPECIAL_CHARS
 - setting, 371
- SSH_EXTRA_ARGS
 - setting, 371
- stats
 - wlc komut satırı seçeneği, 162
- STATUS_URL
 - setting, 372
- store_post_load() (weblate.addons.base.BaseAddon yöntemi), 438
- string resources
 - file format, 77
- SUGGESTION_CLEANUP_DAYS
 - setting, 372
- T**
 - translation
 - dualgual, 102
 - monolingual, 102
 - TS
 - file format, 92
- U**
 - unit_pre_create() (weblate.addons.base.BaseAddon yöntemi), 438
 - unlock
 - wlc komut satırı seçeneği, 162
 - unlock_translation
 - weblate admin command, 401
 - UNUSED_ALERT_DAYS
 - setting, 372
 - UPDATE_LANGUAGES
 - setting, 372
 - updatechecks
 - weblate admin command, 402
 - updategit
 - weblate admin command, 402
 - upload
 - wlc komut satırı seçeneği, 162
 - URL_PREFIX
 - setting, 372
- V**
 - VCS_API_DELAY
 - setting, 373
 - VCS_BACKENDS
 - setting, 373
 - VCS_CLONE_DEPTH
 - setting, 373
 - version
 - wlc komut satırı seçeneği, 161
- W**
 - WEB_WORKERS, 172, 190
 - Weblate (wlc içindeki sınıf), 165
 - weblate admin command

- add_suggestions, 392
 - auto_translate, 393
 - celery_queues, 393
 - checkgit, 393
 - cleanup_ssh_keys, 394
 - cleanuptrans, 394
 - commit_pending, 394
 - commitgit, 394
 - createadmin, 395
 - dump_memory, 395
 - dumpuserdata, 395
 - import_demo, 395
 - import_json, 396
 - import_memory, 396
 - import_project, 397
 - importuserdata, 399
 - importusers, 399
 - install_addon, 399
 - list_languages, 399
 - list_translators, 400
 - list_versions, 400
 - loadpo, 400
 - lock_translation, 400
 - move_language, 401
 - pushgit, 401
 - setupgroups, 401
 - setuplang, 402
 - unlock_translation, 401
 - updatechecks, 402
 - updategit, 402
 - WEBLATE_ADDONS
 - setting, 374
 - WEBLATE_ADMIN_EMAIL, 172--174
 - WEBLATE_ADMIN_NAME, 172--174
 - WEBLATE_ADMIN_PASSWORD, 168, 172--174
 - WEBLATE_ADMIN_PASSWORD_FILE, 174
 - WEBLATE_AKISMET_API_KEY, 409
 - WEBLATE_ALLOWED_HOSTS, 217, 221, 222, 371
 - WEBLATE_API_RATELIMIT_ANON, 114
 - WEBLATE_API_RATELIMIT_USER, 114
 - WEBLATE_AUTH_LDAP_BIND_PASSWORD, 180
 - WEBLATE_DEBUG, 241
 - WEBLATE_EMAIL_HOST_PASSWORD, 187
 - WEBLATE_EMAIL_PORT, 187, 188
 - WEBLATE_EMAIL_USE_SSL, 187, 188
 - WEBLATE_EMAIL_USE_TLS, 187, 188
 - WEBLATE_ENABLE_HTTPS, 251
 - WEBLATE_EXPORTERS
 - setting, 375
 - WEBLATE_FORMATS
 - setting, 375
 - WEBLATE_GITHUB_HOST, 241
 - WEBLATE_GPG_IDENTITY
 - setting, 375
 - WEBLATE_LOCALIZE_CDN_PATH, 189
 - WEBLATE_MACHINERY
 - setting, 375
 - WEBLATE_RATELIMIT_ATTEMPTS, 411
 - WEBLATE_REQUIRE_LOGIN, 370
 - WEBLATE_SECURE_PROXY_SSL_HEADER, 175
 - WEBLATE_SERVICE, 172
 - WEBLATE_SILENCED_SYSTEM_CHECKS, 248
 - WEBLATE_SITE_DOMAIN, 219, 237, 371
 - WEBLATE_WORKERS, 172, 190
 - WeblateConfig (*wlc.config* içindeki sınıf), 165
 - WeblateException, 165
 - WEBSITE_REQUIRED
 - setting, 375
 - wlc, 159
 - modülü, 165
 - wlc komut satırı seçeneği
 - author-email, 163
 - author-name, 163
 - config, 161
 - config-section, 161
 - convert, 162
 - format, 161
 - fuzzy, 163
 - input, 162
 - key, 161
 - method, 163
 - output, 162
 - overwrite, 162
 - url, 161
 - changes, 162
 - cleanup, 162
 - commit, 161
 - download, 162
 - list-components, 161
 - list-languages, 161
 - list-projects, 161
 - list-translations, 161
 - lock, 162
 - lock-status, 162
 - ls, 161
 - pull, 162
 - push, 162
 - repo, 162
 - reset, 162
 - show, 161
 - stats, 162
 - unlock, 162
 - upload, 162
 - version, 161
 - wlc.config
 - modülü, 165
 - wlc.main
 - modülü, 166
 - WPF
 - file format, 93
- ## X
- XLIFF
 - file format, 97
 - XML
 - file format, 82

Y

YAML

file format, [99](#)

YAML Ain't Markup Language

file format, [99](#)