



The Weblate Manual
4.6.2

Michal Čihař

20210508

?? ??

Weblate

??????

Weblate 1 [GNU gettext](#) [Android string resources](#)1
??????

?????
Component configuration ?

?:
../devel/integration

??????

?

?????
?:

- 1.????
- 2.????
- 3.????

?????

?????
???? 2.5 .
????

?:

 Your profile

[Languages](#)

[Preferences](#)

[Notifications](#)

[Account](#)

[Profile](#)

[Licenses](#)

[Audit log](#)

[API access](#)

Preferences

☐ Hide completed translations on the dashboard

Translation editor mode

Full editor

Zen editor mode

Top to bottom

Number of nearby strings

15

Number of nearby strings to show in each direction in the full editor.

☒ Show secondary translations in the Zen mode

☐ Hide source if a secondary translation exists

Editor link

Enter a custom URL to be used as link to the source code. You can use `{{branch}}` for branch, `{{filename}}` and `{{line}}` as filename and line placeholders.

Special characters

You can specify additional special visual keyboard characters to be shown while translating. It can be useful for characters you use frequently, but are hard to type on your keyboard.

Default dashboard view

☒ Watched translations

☐ Suggested translations

Save

#####:

> ##### Weblate

#####

#####

#####Weblate #####

#####: #####

#####

##: settings.py #####: ##### SINGLE_PROJECT ##### Weblate

#####

Component	Translated	Untranslated	Untranslated words	Checks	Suggestions
 WeblateOrg/Android — Czech   MIT	76%	3	3		
 WeblateOrg/Django — Hungarian   GPL-3.0	69%	8	109	1	
 WeblateOrg/Django — Czech   GPL-3.0	96%	1	12	4	
 WeblateOrg/Django — Hebrew   GPL-3.0	92%	2	15		
 WeblateOrg/Djangojs — Czech   GPL-3.0	✓				
 WeblateOrg/Djangojs — Hebrew   GPL-3.0	✓				
 WeblateOrg/Djangojs — Hungarian   GPL-3.0	96%	2	6		
 WeblateOrg/Language names — Czech   GPL-3.0	✓				
 WeblateOrg/Language names — Hungarian   GPL-3.0	81%	4	5		
 WeblateOrg/Language names — Hebrew   GPL-3.0	✓				
 WeblateOrg/WeblateOrg — Czech   GPL-3.0	✓				
 WeblateOrg/WeblateOrg — Hungarian   GPL-3.0	✓				
 WeblateOrg/WeblateOrg — Hebrew   GPL-3.0	✓				

[illegible]

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

 Your profile

[Languages](#)

[Preferences](#)

[Notifications](#)

[Account](#)

[Profile](#)

[Licenses](#)

[Audit log](#)

[API access](#)

Watched projects

☒ Automatically watch projects on contribution

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

Watched projects

Search...

Available:

WeblateOrg

Chosen:

WeblateOrg

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

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

Save

Notification settings

[Other projects](#)

[Watched projects](#)

[Managed projects](#)

Component wide notifications

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

Repository failure

Do not notify

Repository operation

Do not notify

Component locking

Do not notify

Changed license

Do not notify

Parse error

Do not notify

Comment on own translation

Instant notification

Mentioned in comment

Instant notification

New language

Do not notify

New translation component

Do not notify

New announcement

Instant notification

New alert

Do not notify

Translation notifications

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

New string

Do not notify

New contributor

Do not notify

New suggestion

Do not notify

New comment

Do not notify

Changed string

Do not notify

Translated string

Do not notify

Approved string

Do not notify

Pending suggestions

Do not notify

Strings needing action

Do not notify

Save

????

???? ????Weblate ??:
Weblate ? ??

?: ? Weblate ? GitLab ? GitHub?Google?Facebook?Bitbucket
? OAuth 2.0 ?

 Your profile

[Languages](#)

[Preferences](#)

[Notifications](#)

Account

[Profile](#)

[Licenses](#)

[Audit log](#)

[API access](#)

Account 

Username

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

Full name

E-mail

You can add another e-mail address below.
Your name and e-mail will appear as commit authorship.

[Save](#)

Current user identities 

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

Add new association

 **E-mail**

Removal

Account removal deletes all your private data.

[Remove my account](#)

User data

You can download all your private data.

[Download user data](#)

```

[REDACTED]
[REDACTED]ENABLE_AVATARS [REDACTED]https://gravatar.com/[REDACTED]

```

[illegible]

10

????

???
 ?????????
 ?????
 ???
 ???
 ???

W

Weblate

Dashboard

Projects

Languages

Checks

WebplateOrg / Django / Czech

translated 96%

Overview

Info

Search

Insights

Files

Tools

Manage

Share

Watching

Translation status

26 Strings

96%

183 Words

93%

Browse

Translate

Strings status

26

All strings — 183 words

Browse

Edit

Zen

25

Translated strings — 171 words

Browse

Edit

Zen

1

Strings needing action — 12 words

Browse

Edit

Zen

1

Not translated strings — 12 words

Browse

Edit

Zen

1

Strings needing action without suggestions — 12 words

Browse

Edit

Zen

3

Strings with any failing checks — 11 words

Browse

Edit

Zen

3

Translated strings with any failing checks — 11 words

Browse

Edit

Zen

1

Failed check: Unchanged translation — 4 words

Browse

Edit

Zen

1

Failed check: Mismatched full stop — 4 words

Browse

Edit

Zen

1

Failed check: Python format — 3 words

Browse

Edit

Zen

Other components

Component	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
Android MIT	76%	3	3			
Language names GPL-3.0						
Glossary WeblateOrg GPL-3.0						
Djangojs GPL-3.0						

Browse all components

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

11

??: ??????Weblate ?????????????????

[illegible]

□□□□

[illegible]

??:

report-source[?]

□ □ □ □ □

[illegible]

??:

variants[?] [?] [?] [?] [?]

???

QUESTION:

??:

labels

??






















































[illegible]

1	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
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[illegible]

???

```

1 webplate "car" 1 "cars" 2

```

[illegible]

??:

??????

W

Webate

Dashboard

Projects

Languages

Checks

+

WebateOrg / Django / Czech / Translate

translated 96%

<

<

1 / 1

>

>

Custom search

'%(count)s word'

Position and priority

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

%(count)s slovo

Czech, Few

%(count)s slova

Czech, Other

%(count)s slov

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

Needs editing

Save

Suggest

Skip

Nearby strings

Comments

Automatic suggestions

Other languages

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Glossary

English Czech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

webate/templates/translation.h

tml:149

String age

16 seconds ago

Source string age

16 seconds ago

Translation file

webate/locale/cs/LC_MESSAGE

S/django.po, string 5

Powered by Webate 4.6

About Webate

Legal

Contact

Documentation

Donate to Webate

13

????? ????????

????? 2.18 ??? : ?????? ????????? 2.18 ???
???????????????????? ?????????:

?????? ??????????

Alt+Home
Alt+End
Alt+PageUp or
Ctrl ↑ or
Alt ↑ or
Cmd ↑
Alt+PageDown or
Ctrl+↓ or
Alt+↓ or
Cmd+↓
Alt+Enter or
Ctrl+Enter or
Cmd+Enter
Ctrl+Shift+Enter or
Cmd+Shift+Enter
Ctrl+E or
Cmd+E
Ctrl+U or
Cmd+U
Ctrl+M or
Cmd+M
Ctrl+1 to Ctrl+9 or
Cmd+1 to Cmd+9
Ctrl+M+1 to 9 or
Cmd+M+1 to 9
Ctrl+I+1 to 9 or
Cmd+I+1 to 9
Ctrl+J or
Cmd+J
Ctrl+S or
Cmd+S
Ctrl+O or
Cmd+O
Ctrl+Y or
Cmd+Y

??

????????????????????
????????????????????
????????1 ?????????????

????????????????

????????????

????????????????????

????????????????????

????????????????????

???? ?????????: *????*

??

????????????????????????????

?????????????????? 1 ?????????

??????? ???????

????????????????

????????????

??? ???????

?????? ???????

??

3 ?????????????????:

??????? ?????????????

Weblate ????????????????? RTL ???????

SPECIAL_CHARS ???????

Webate

DashboardProjectsLanguagesChecks

WebateOrg / Django / Hebrew / Translate

translated 92%

<<1 / 26>>

All strings

Position and priority

Translation

English

Files

Hebrew

⌨

⬅NBS...-“”„’,-—ZWNJZWJLRMLRELPDFLROLRO

קבצים

Needs editing ⓘ5 / 100 · 5RTL

SaveSuggestSkip

Nearby strings 16

Comments

Automatic suggestions

Other languages 3

History

Language	Translated string
Czech	Soubory
Hungarian	Fájlok
English	Files

Glossary

EnglishHebrew

No related strings found in the glossary.

+ Add term to glossary

String information ⓘ

Screenshot context

No screenshot currently associated.

+ Add screenshot

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

No flags currently set.

Source string location

webate/templates/translation.html:45 · webate/trans/forms.py:1404

String age

31 seconds ago

Source string age

32 seconds ago

Translation file

webate/locale/he/LC_MESSAGES/django.po, string 1

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

????????

??

???? ID????msgctxt????

???? Webate ???: Visual context for strings

??

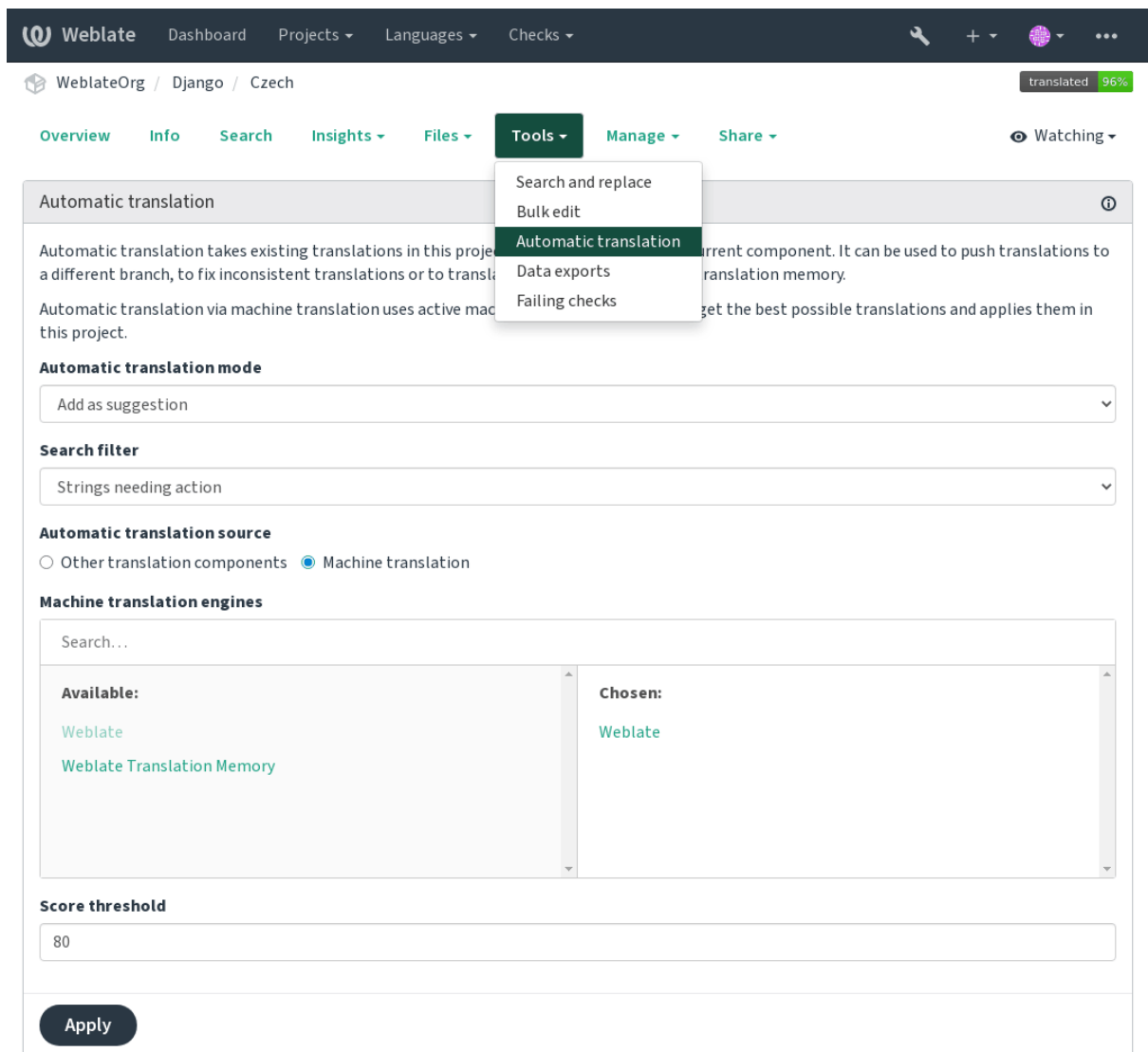
??: ???

?????????????????????????????????????: ?????

??

Webate ?????????????????????????????????????

15



2022-02-22 14:00:00:

Webblate 2022-02-22 14:00:00

2022-02-22 14:00:00

2022-02-22 14:00:00

2022-02-22 14:00:00: 2022-02-22 14:00:00

2022-02-22 14:00:00: 2022-02-22 14:00:00 WEB 2022-02-22 14:00:00

2022-02-22 14:00:00:

Keeping translations same across components

????

:

????

/

2

????

1

:

:

:

: labels

/

:

Bulk edit addon

????????????????

: Available options might be limited by *access control* settings.

????????

/

The first option is to download the file in the original format as it is stored in the repository. In this case, any pending changes in the translation are getting committed and the up-to-date file is yield without any conversions.

Weblate
 ↓

gettext PO

gettext ???? XLIFF

XLIFF 1.1

TermBase eXchange

Translation Memory eXchange

gettext MO

CSV

Excel Open XML

JSON

Android ?????

iOS ?????


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

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible][illegible]

Each project can include one or more glossaries as a shorthand for storing terminology. Glossary easily maintaining consistency of the translation.

4.5. Weblate

XX
XXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXX


Webplate
Dashboard
Projects
Languages
Checks

+

...

WebplateOrg /
Glossary
WebplateOrg /
Czech

translated 100%

Overview
Info
Search
Insights
Files
Tools
Share

Not watching

Translation status

2 Strings 100%

3 Words 100%

Add new glossary term

Browse

Translate

Strings status

2 All strings — 3 words Browse Edit Zen

2 Translated strings — 3 words Browse Edit Zen

Other components

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

Browse all components


Dashboard
Projects ▾
Languages ▾
Checks ▾
+ ▾

...


WeblateOrg / **Glossary** WeblateOrg / Czech / Browse
translated 100%

⏪
⏴
1 / 1
⏵
⏩

All strings ▾
Source string ▾
≡

+ Add new glossary term

English ▾	Czech
machine translation	strojový překlad
project	projekt

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

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

+

...

[WeblateOrg](#) /
[Glossary](#)
[WeblateOrg](#) /
[Czech](#) /
[Translate](#)

translated
100%

<
<
2 / 2
>
>

All strings

Source string

Zen

Glossary term

English

project

Czech

projekt

☐ Needs editing

Explanation

Additional explanation to clarify meaning or usage of the string.

Save

Suggest

Skip

Tools

Nearby strings
2

Comments

Automatic suggestions

Other languages

English

machine translation

project

Czech

strojový překlad

projekt

Delete string
Mark as read-only
Mark as forbidden translation
Mark as terminology
Add variant of this string

Glossary

English
Czech

project
projekt
WeblateOrg

Add term to glossary

String information

String age
5 seconds ago

Source string age
5 seconds ago

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

????????????????

Forbidden

4.5 Forbidden.

Flagging certain glossary term translations as `forbidden`, by bulk-editing, typing in the flag, or by using *Tools* ↓ *Mark as forbidden translation* means they are **not** to be used. Use this to clarify translation when some words are ambiguous or could have unexpected meanings.

Example:

Forbidden translation

Terminology

4.5 Terminology.

Flagging certain glossary terms as `terminology` by bulk-editing, typing in the flag, or or by using *Tools* ↓ *Mark as terminology* adds entries for them to all languages in the glossary. Use this for important terms that should be well thought out, and retain a consistent meaning across all languages.

Example:

Terminology

Variants

Variants are a generic way to group strings together. All term variants are listed in the glossary sidebar when translating.

Example: You can use this to add abbreviations or shorter expressions for a term.

Example:

variants

Long strings

Long strings

Long strings

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

Translation

English

Singular

Plural

Czech, One

Czech, Few

Czech, Other

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

Needs editing

Save

Suggest

Skip

Nearby strings 20

Comments

Automatic suggestions

Other languages 3

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Things to check

Python format 1

Following format strings are missing: %(count)s

Dismiss

☐ For all languages

Missing plurals 2

Some plural forms are not translated

Dismiss

☐ For all languages

Glossary

English

Czech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

String age

21 seconds ago

Source string age

22 seconds ago

Translation file

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

pending

????

???? Weblate ???

??:

AUTOFIX_LIST

????

Weblate ???

??:

CHECK_LIST? ?????????????????

???????

??

BBcode ???????

???? BBcode ???????

BBCode ???

??????BBCode ?????????????????????????????

??: ???BBCode ???

?????????

?????????? 2 ????????

???? 4.1 ???.

??

???: ??: *Weblate*
?????????

?????????????

???? 4.5 ???.

????????????????????????????????

?????check-glossary ??????????????????: ??????????????????????????????????:

??

????????????????????????????????Weblate ?????????????????????????????

??:

??? ?????????????????? ??????

??????

????????????????????

??

??

??????????

??

Weblate [??](#):
C [???](#) 'c-format'??Gettext [??](#) PO [??????](#) **xgettext**
[??](#)

[????????????????????????????????](#): *Additional info on source strings*[??Component configuration](#)
[??](#)

???: Weblate [????????????????????????????????](#) [??????????](#) [??????????](#)

[??](#):

Weblate
Dashboard
Projects
Languages
Checks

1/1

Custom search
'%(count)s word'

Position and priority

Zen

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

%(count)s slovo

Czech, Few

%(count)s slova

Czech, Other

%(count)s slov

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

☐ Needs editing

Save

Suggest

Skip

Nearby strings
20
Comments
Automatic suggestions
Other languages
3
History

No matching activity found.

Browse all component changes

Glossary

EnglishCzech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

String age

16 seconds ago

Source string age

16 seconds ago

Translation file

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

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

AngularJS

AngularJS

```

Your balance is {{amount}} {{ currency }}
angularjs-format

```

:

AngularJS text interpolation

27

C

C 

```
There are %d apples
Your balance is %1$d %2$s
c-format
```

:

C format strings  C printf format

C#

C# 

```
There are {0} apples
c-sharp-format
```

:

C# String Format

ECMAScript

ECMAScript 

```
There are ${number} apples
es-format
```

:

Template literals

i18next

i18next 

4.0 .

```
There are {{number}} apples
There are $t(number) apples
i18next-interpolation
```

:

i18next interpolation

Java

Java 

```
There are %d apples
Your balance is %1$d %2$s
java-format
```

:

Java Format Strings

Java MessageFormat

Java `????? ?????????????????????`

```
?????????????There are {0} apples
?????????????  java-messageformat ??????????????????
                auto-java-messageformat ?????????????????????????????????
```

??:

Java MessageFormat

JavaScript ??

JavaScript `????????????????????`

```
?????????????There are %d apples
?????????????  javascript-format
```

??:

JavaScript formatting strings

Lua ??

Lua `????????????????`

```
?????????????There are %d apples
?????????????  lua-format
```

??:

Lua `??????`

`????????????????`

`????????????????????????????`

`?????? 4.0 ???.`

```
?????????????There are %number% apples
?????????????  percent-placeholders
```

Perl ??

Perl `????????????????`

```
?????????????There are %d apples
?????????????Your balance is %1$d %2$s
?????????????  perl-format
```

??:

Perl sprintf `?`Perl Format Strings

PHP

PHP 

```
There are %d apples
Your balance is %1$d %2$s
php-format
```

:

[PHP sprintf documentation](#)  [PHP Format Strings](#)

Python

Python 

```
There are {} apples
Your balance is {amount} {currency}
python-brace-format
```

:

[Python brace format](#)  [Python Format Strings](#)

Python

Python 

```
There are %d apples
Your balance is %(amount) %(currency)
python-format
```

:

[Python string formatting](#)  [Python Format Strings](#)

Qt

Qt 

```
There are %1 apples
qt-format
```

:

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

Qt

Qt 

```
There are %Ln apple(s)
qt-plural-format
```

:

[Qt i18n guide](#)

Ruby [?]

Ruby [?]

```
There are %d apples
Your balance is %1$f %2$s
Your balance is %+.2<amount>f %<currency>s
Your balance is %{amount} %{currency}
ruby-format
```

?

Ruby Kernel#sprintf

Scheme [?]

Scheme [?]

```
There are ~d apples
scheme-format
```

?

Srfi 28, Chicken Scheme format, Guile Scheme formatted output

Vue I18n [?]

Vue I18n [?]

```
There are {count} apples
Rails i18n [?] There are %{count} apples
@:message.dio @:message.the_world!
vue-format
```

?

Vue I18n Formatting [?] Vue I18n Linked locale messages

????

????

????VCS ????? VCS ?????

?????

????

Weblate ?????

1 ?????
????

?: ?????
????? ???? ????
???? ????
?

?

Keeping translations same across components

??????????

????????????????????????????????????

????? 3.7 ????.

??

?????????????????? 1 ???? 2 ?????????????????????????????1 ???????????????

????????? font-* ??????????: ?????????????????????????????ubuntu ?????? ??? 22
??????????????2 ? 500 ??????????????????????:

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

??? *Component configuration* ?? font-* ???
replacements: ???

??:

????????? ? ?????????????????? ? ??????????

\n ????

???? \n ?????????

?? \n ?????????????????????????????????????

?????????

????????????????????

??

??:

??? Wikipedia ??:

??????????

????????????????????

??3 ?????? (...) ?????????????? (. . .) ?????????

?????????????? 3 ?????????????????????????????????

??:

????? Wikipedia ??:

?????????

????????????????

??

??:

??? Wikipedia ??:

????

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

????

XXXXXXXXXXXX (...) XXXXX 3 XXXXX (...) XXXXXXXXXX

XXXXXXXXXXXXXXXXXX (...) XXXXXXXXXXXXXXX3 XXXXXX (. . .) XXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXUnicode XXX

??:

XXXXX Wikipedia ??:

???????

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXX 4.1 XXX.

XX

???????????

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XX

XX

?????????

XX

XXXXXX 4.1 XXX.

XX

XX

?????????

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XX

Python ? Gettext ???:

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

??

XXXXXX 3.9 XXX.

XX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXX XXXXXXX XXXXXXXXXXXXXXX

????

????????????????r"regexp" ?????????????????

????2~5 ?????????????????????????source:r"[2-5]" ???????

????????

??:

W

Weblate

Dashboard

Projects

Languages

Checks

+

...

WebateOrg / Django / Czech / Translate

translated 96%

K

<

1 / 1

>

>I

Custom search

'%(count)s word'

Not translated strings • state:empty

Strings needing action • state:<translated

Translated strings • state:>=translated

Strings marked for edit • state:needs-editing

Strings with suggestions • has:suggestion

Strings with variants • has:variant

Strings with labels • has:label

Strings with context • has:context

Strings needing action without suggestions • state:<translated AND NOT has:suggestion

Strings with comments • has:comment

Strings with any failing checks • has:check

Approved strings • state:approved

Strings waiting for review • state:translated

Zen

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

%(count)s slovo

Czech, Few

%(count)s slova

Czech, Other

%(count)s slov

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

☐ Needs editing

Save

Suggest

Skip

Nearby strings 20

Comments

Automatic suggestions

Other languages 3

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.h

tml:149

String age

16 seconds ago

Source string age

16 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGE

S/django.po, string 5

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

41

??????

????????????????????????????????????

W

Weblate

Dashboard

Projects

Languages

Checks

WebateOrg

Django

Czech

Translate

translated 96%

1 / 1

Not translated strings

state:empty

Zen

Position and priority

Position and priority

Position

Priority

Labels

Source string

Translated string

Age of string

Number of words

Number of comments

Number of failing checks

Key

Translation

English

The string uses three dots (...) instead of an ellipsis character (...)

Czech

Needs editing

Save

Suggest

Skip

Nearby strings 16

Comments

Automatic suggestions

Other languages 3

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

No flags currently set.

Source string location

weblate/checks/source.py:54

String age

18 seconds ago

Source string age

18 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGES/django.po, string 26

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

42

[illegible]

```
access control [REDACTED]
[REDACTED] [REDACTED] [REDACTED]
```

[illegible]

??:

XX Webplate XXXXXXXXXXXXXXXX
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XX
 XXXXXXXXXX XXXXXXXXXX
 XXXXXXXXXX XXXXXXXXXX
 XX

□□□□□□

□ □ □ □ □ □

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

SSH `git@github.com:weblate/weblate.git`

SSH `git@github.com:weblate/weblate.git` `git@github.com:weblate/weblate.git`

`git@github.com:weblate/weblate.git`

```
git@github.com:weblate/weblate.git upstream git@github.com:weblate/weblate.git msgmerge
git@github.com:weblate/weblate.git git@github.com:weblate/weblate.git git@github.com:weblate/weblate.git
git@github.com:weblate/weblate.git git@github.com:weblate/weblate.git git@github.com:weblate/weblate.git
git@github.com:weblate/weblate.git upstream git@github.com:weblate/weblate.git git@github.com:weblate/weblate.git
```

```
git@github.com:weblate/weblate.git git@github.com:weblate/weblate.git git@github.com:weblate/weblate.git Git exporter
git@github.com:weblate/weblate.git API git@github.com:weblate/weblate.git
```

```
# Commit all pending changes in Weblate, you can do this in the UI as well:
wlc commit
# Lock the translation in Weblate, again this can be done in the UI as well:
wlc lock
# Add Weblate as remote:
git remote add weblate https://hosted.weblate.org/git/project/component/
# You might need to include credentials in some cases:
git remote add weblate https://username:APIKEY@hosted.weblate.org/git/project/component/
# Update weblate remote:
git remote update weblate
# Merge Weblate changes:
git merge weblate/main
# Resolve conflicts:
edit ...
git add ...
...
git commit
# Push changes to upstream repository, Weblate will fetch merge from there:
git push
# Open Weblate for translation:
wlc unlock
```

Weblate `git@github.com:weblate/weblate.git`:

```
# Add and update Weblate remotes
git remote add weblate-one https://hosted.weblate.org/git/project/one/
git remote add weblate-second https://hosted.weblate.org/git/project/second/
git remote update weblate-one weblate-second
# Merge QA_4_7 branch:
git checkout QA_4_7
git merge weblate-one/QA_4_7
... # Resolve conflicts
git commit
# Merge main branch:
git checkout main
git merge weblate-second/main
... # Resolve conflicts
git commit
```

(`git@github.com:weblate/weblate.git`)

()

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

```
gettext PO [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]:
```

```

Webdate      Git      ??????????????      ??????????????????????????????????upstream      ?      Git      ??????????????????????
?????????????????????????????upstream      ?      Git      ???????????????? 2 ??????:      ??????????????????????????????:

```

```
# 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
```

??:

How to export the Git repository that Weblate uses? · Web ████████████████████ · Avoiding merge conflicts · Weblate ██████████

????????????????

Webplate **Project configuration** every **Component configuration**

Weblate

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

??:

Keeping translations same across components

How to translate multi-platform projects?

Weblate supports a wide range of file formats (see [Translations formats](#)) and the easiest approach is to use the native format for each platform.

Once you have added all platform translation files as components in one project (see [Adding translation projects and components](#)), you can utilize the translation propagation feature (turned on by default, and can be turned off in the [Component configuration](#)) to translate strings for all platforms at once.

??:

Keeping translations same across components

How to export the Git repository that Weblate uses?

There is nothing special about the repository, it lives under the `DATA_DIR` directory and is named `vcs/<project>/<component>/`. If you have SSH access to this machine, you can use the repository directly.

For anonymous access, you might want to run a Git server and let it serve the repository to the outside world.

Alternatively, you can use *Git exporter* inside Weblate to automate this.

What are the options for pushing changes back upstream?

This heavily depends on your setup, Weblate is quite flexible in this area. Here are examples of some workflows used with Weblate:

Weblate automatically pushes and merges changes (see [????????????????????????????????](#)).

You manually tell Weblate to push (it needs push access to the upstream repository).

Somebody manually merges changes from the Weblate git repository into the upstream repository.

Somebody rewrites history produced by Weblate (e.g. by eliminating merge commits), merges changes, and tells Weblate to reset the content in the upstream repository.

Of course you are free to mix all of these as you wish.

How can I limit Weblate access to only translations, without exposing source code to it?

You can use `git submodule` for separating translations from source code while still having them under version control.

1. Create a repository with your translation files.

2. Add this as a submodule to your code:

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

3. Link Weblate to this repository, it no longer needs access to the repository containing your source code.

4. You can update the main repository with translations from Weblate by:

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

Please consult the `git submodule` documentation for more details.

How can I check whether my Weblate is set up properly?

Weblate includes a set of configuration checks which you can see in the admin interface, just follow the *Performance report* link in the admin interface, or open the `/manage/performance/` URL directly.

Why are all commits committed by Weblate <noreply@weblate.org>?

This is the default committer name, configured when you create a translation component. You can change it in the administration at any time.

The author of every commit (if the underlying VCS supports it) is still recorded correctly as the user that made the translation.

??:

Component configuration

Usage

How do I review the translations of others?

There are several review based workflows available in Weblate, see [review workflows](#).

You can subscribe to any changes made in [a project](#) and then check others contributions as they come in by e-mail.

There is a review tool available at the bottom of the translation view, where you can choose to browse translations made by others since a given date.

Q:

[review workflow](#)

How do I provide feedback on a source string?

On context tabs below translation, you can use the *Comments* tab to provide feedback on a source string, or discuss it with other translators.

Q:

report-source[report source](#)

How can I use existing translations while translating?

Weblate [importing translations](#)

[importing translations](#) Weblate [importing translations](#)

Use the import functionality to load compendium as translations, suggestions or translations needing review. This is the best approach for a one-time translation using a compendium or a similar translation database.

You can set up *tmserver* with all databases you have and let Weblate use it. This is good when you want to use it several times during translation.

Another option is to translate all related projects in a single Weblate instance, which will make it automatically pick up translations from other projects as well.

Q:

[importing translations](#)

Does Weblate update translation files besides translations?

Weblate tries to limit changes in translation files to a minimum. For some file formats it might unfortunately lead to reformatting the file. If you want to keep the file formatted your way, please use a pre-commit hook for that.

Q:

updating-target-files

Where do language definitions come from and how can I add my own?

The basic set of language definitions is included within Weblate and Translate-toolkit. This covers more than 150 languages and includes info about plural forms or text direction.

You are free to define your own languages in the administrative interface, you just need to provide info about it.

Q:

[language definitions](#)

Can Weblate highlight changes in a fuzzy string?

Weblate supports this, however it needs the data to show the difference.

For Gettext PO files, you have to pass the parameter `--previous` to **msgmerge** when updating PO files, for example:

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

For monolingual translations, Weblate can find the previous string by ID, so it shows the differences automatically.

Why does Weblate still show old translation strings when I've updated the template?

Weblate does not try to manipulate the translation files in any way other than allowing translators to translate. So it also does not update the translatable files when the template or source code have been changed. You simply have to do this manually and push changes to the repository, Weblate will then pick up the changes automatically.

??: It is usually a good idea to merge changes done in Weblate before updating translation files, as otherwise you will usually end up with some conflicts to merge.

For example with gettext PO files, you can update the translation files using the **msgmerge** tool:

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

In case you want to do the update automatically, you can install add-on [POT to PO msgmerge](#).

??:

updating-target-files

Troubleshooting

Requests sometimes fail with "too many open files" error

This happens sometimes when your Git repository grows too much and you have many of them. Compressing the Git repositories will improve this situation.

The easiest way to do this is to run:

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

??:

`DATA_DIR`

When accessing the site I get a "Bad Request (400)" error

This is most likely caused by an improperly configured `ALLOWED_HOSTS`. It needs to contain all hostnames you want to access on your Weblate. For example:

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

??:

`ALLOWED_HOSTS`

What does mean "There are more files for the single language (en)"?

This typically happens when you have translation file for source language. Weblate keeps track of source strings and reserves source language for this. The additional file for same language is not processed.

```
#####
#####
#####
#####Weblate #####
```

??: You might get similar error message for other languages as well. In that case the most likely reason is that several files map to single language in Weblate.

This can be caused by using obsolete language codes together with new one (ja and jp for Japanese) or including both country specific and generic codes (fr and fr_FR). See [???????](#) for more details.

??

Does Weblate support other VCSes than Git and Mercurial?

Weblate currently does not have native support for anything other than *Git* (with extended support for *GitHub*, *Gerrit* and *Subversion*) and *Mercurial*, but it is possible to write backends for other VCSes.

You can also use *Git* [????](#) [????](#) in Git to access other VCSes.

Weblate also supports VCS-less operation, see [????](#) [????](#).

??: For native support of other VCSes, Weblate requires using distributed VCS, and could probably be adjusted to work with anything other than Git and Mercurial, but somebody has to implement this support.

??:

[??????????](#)

How does Weblate credit translators?

Every change made in Weblate is committed into VCS under the translators name. This way every single change has proper authorship, and you can track it down using the standard VCS tools you use for code.

Additionally, when the translation file format supports it, the file headers are updated to include the translator's name.

??:

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

Why does Weblate force showing all PO files in a single tree?

Weblate was designed in a way that every PO file is represented as a single component. This is beneficial for translators, so they know what they are actually translating.

[????](#) 4.2 **??:** [??](#)

Why does Weblate use language codes such sr_Latn or zh_Hant?

These are language codes defined by [RFC 4646](#) to better indicate that they are really different languages instead previously wrongly used modifiers (for @latin variants) or country codes (for Chinese).

Weblate still understands legacy language codes and will map them to current one - for example sr@latin will be handled as sr_Latn or zh@CN as zh_Hans.

??:

[????](#)

File formats

Weblate supports most translation format understood by [translate-toolkit](#), however each format being slightly different, some issues with formats that are not well tested can arise.

File formats:

Translation Related File Formats

Tip: When choosing a file format for your application, it's better to stick some well established format in the toolkit/platform you use. This way your translators can additionally use whatever tools they are used to, and will more likely contribute to your project.

Bilingual and monolingual formats

Both monolingual and bilingual formats are supported. Bilingual formats store two languages in single file—source and translation (typical examples are *GNU gettext*, *XLIFF* or *Apple iOS strings*). On the other side, monolingual formats identify the string by ID, and each language file contains only the mapping of those to any given language (typically *Android string resources*). Some file formats are used in both variants, see the detailed description below.

For correct use of monolingual files, Weblate requires access to a file containing complete list of strings to translate with their source—this file is called `strings.txt` within Weblate, though the naming might vary in your paradigm.

Additionally this workflow can be extended by utilizing `strings.xml` to include strings provided by developers, but not to be used as is in the final strings.

File format detection

Weblate can automatically detect several widespread file formats, but this detection can harm your performance and will limit features specific to given file format (for example automatic addition of new translations).

Translation types capabilities

Capabilities of all supported formats:

Format	Lingual-ity ^{p. 53, 1}	Plurals ^{p. 53, 2}	Com-ments ^{p. 53, 3}	Context ^{p. 53, 4}	Loca-tion ^{p. 53, 5}	Flags ^{p. 53, 8}	Additional states ^{p. 53, 6}
<i>GNU gettext</i>	<code>gettext</code>	yes	yes	yes	yes	yes ⁹	needs editing
<i>Monolingual gettext</i>	mono	yes	yes	yes	yes	yes [?]	needs editing
<i>XLIFF</i>	both	yes	yes	yes	yes	yes ¹⁰	needs editing, approved
<i>Java properties</i>	both	no	yes	no	no	no	
<i>GWT translations</i>	mono	yes	yes	no	no	no	
<i>Joomla translations</i>	mono	no	yes	no	yes	no	
<i>Qt Linguist .ts</i>	both	yes	yes	no	yes	yes [?]	needs editing
<i>Android string resources</i>	mono	yes	yes ⁷	no	no	yes [?]	
<i>Apple iOS strings</i>	<code>strings</code>	no	yes	no	no	no	
<i>PHP files</i>	mono	no ¹¹	yes	no	no	no	
<i>JSON files</i>	mono	no	no	no	no	no	
<i>JSON files</i>	mono	yes	no	no	no	no	
<i>go-i18n JSON files</i>	mono	yes	no	no	no	no	
<i>ARB File</i>	mono	yes	yes	no	no	no	
<i>WebExtension JSON</i>	mono	yes	yes	no	no	no	

File format detection

Table 1 – Supported file formats							
Format	Lingual- ity ^{p. 53, 1}	Plurals ^{p. 53, 2}	Com- ments ^{p. 53, 3}	Context ^{p. 53, 4}	Loca- tion ^{p. 53, 5}	Flags ^{p. 53, 8}	Additional states ^{p. 53, 6}
.XML resource files	mono	no	yes	no	no	yes?	
CSV files	both	no	yes	yes	yes	no	needs editing
YAML files	mono	no	yes	no	no	no	
Ruby YAML files	mono	yes	yes	no	no	no	
DTD files	mono	no	no	no	no	no	
Flat XML files	mono	no	no	no	no	yes?	
Windows RC files	mono	no	yes	no	no	no	
Excel OpenXML	mono	no	yes	yes	yes	no	needs editing
.pot	mono	no	no	no	no	no	
.po							
Subtitle files	mono	no	no	no	yes	no	
HTML .i18n	mono	no	no	no	no	no	
OpenDocument Format	mono	no	no	no	no	no	
IDML Format	mono	no	no	no	no	no	
INI translations	mono	no	no	no	no	no	
Inno Setup INI	mono	no	no	no	no	no	
.termBase eXchange		no	yes	no	no	yes?	
.xlf	mono	no	no	no	no	no	

[Supported file formats](#)

[3.10](#) [.xlf](#).

Read-only strings from translation files will be included, but can not be edited in Weblate. This feature is natively supported by few formats ([XLIFF](#) and [Android string resources](#)), but can be emulated in others by adding a read-only flag, see [.xlf](#).

[.po](#): Bilingual and monolingual formats

Plurals are necessary to properly localize strings with variable count.

Comments can be used to pass additional info about the string to translate.

Context is used to differentiate identical strings used in different scopes (for example *Sun* can be used as an abbreviated name of the day "Sunday" or as the name of our closest star).

Location of a string in source code might help proficient translators figure out how the string is used.

[.po](#): [.po](#)

Additional states supported by the file format in addition to "Not translated" and "Translated".

The gettext type comments are used as flags.

The flags are extracted from the non-standard attribute `weblate-flags` for all XML based formats. Additionally `max-length:N` is supported through the `maxwidth` attribute as defined in the XLIFF standard, see [Specifying translation flags](#).

XML comment placed before the `<string>` element, parsed as a developer comment.

The plurals are supported only for Laravel which uses in string syntax to define them, see [Localization in Laravel](#).

GNU gettext

libre

Contextual info stored in the file is supported by adjusting its headers or linking to corresponding source files.

The bilingual gettext PO file typically looks like this:

```
#: 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"
```

Typical Weblate *Component configuration*

```

???????? po/*.po
???????? Empty
???????? po/messages.pot
???????? Gettext PO file

```

??:

```

devel/gettext?devel/sphinx?Gettext Wikipedia ??: ?PO Files?configure ???? ALL_LINGUAS ?????gettext
?????????LINGUAS ????MÓ ????POT ???? PO ????msgmerge?

```

Monolingual gettext

Some projects decide to use gettext as monolingual formats—they code just the IDs in their source code and the string then needs to be translated to all languages, including English. This is supported, though you have to choose this file format explicitly when importing components into Weblate.

The monolingual gettext PO file typically looks like this:

```
#: 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"
```

While the base language file will be:

```
#: 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"
```

Typical Weblate *Component configuration*

```
##### po/* .po
##### po/en.po
##### po/messages.pot
##### Gettext PO file (monolingual)
```

XLIFF

XML-based format created to standardize translation files, but in the end it is one of [many standards](#), in this area.

XML Localization Interchange File Format (XLIFF) is usually used as bilingual, but Weblate supports it as monolingual as well.

🔗:

XML Localization Interchange File Format (XLIFF) specification

Translation states

🔗 3.3 🔗: Weblate ignored the `state` attribute prior to the 3.3 release.

The `state` attribute in the file is partially processed and mapped to the "Needs edit" state in Weblate (the following states are used to flag the string as needing edit if there is a target present: `new`, `needs-translation`, `needs-adaptation`, `needs-l10n`). Should the `state` attribute be missing, a string is considered translated as soon as a `<target>` element exists.

If the translation string has `approved="yes"`, it will also be imported into Weblate as "Approved", anything else will be imported as "Waiting for review" (which matches the XLIFF specification).

While saving, Weblate doesn't add those attributes unless necessary:

The `state` attribute is only added in case string is marked as needing edit.

The `approved` attribute is only added in case string has been reviewed.

In other cases the attributes are not added, but they are updated in case they are present.

That means that when using the XLIFF format, it is strongly recommended to turn on the Weblate review process, in order to see and change the approved state of strings.

Similarly upon importing such files (in the upload form), you should choose *Import as translated* under *Processing of strings needing edit*.

🔗:

🔗

Whitespace and newlines in XLIFF

Generally types or amounts of whitespace is not differentiated between in XML formats. If you want to keep it, you have to add the `xml:space="preserve"` flag to the string.

🔗:

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

Specifying translation flags

You can specify additional translation flags (see [XXXXXXXXXXXX](#)) by using the `weblate-flags` attribute. Weblate also understands `maxwidth` and `font` attributes from the XLIFF specification:

```
<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>
```

The `font` attribute is parsed for font family, size and weight, the above example shows all of that, though only font family is required. Any whitespace in the font family is converted to underscore, so `Source Sans Pro` becomes `Source_Sans_Pro`, please keep that in mind when naming the font group (see [XXXXXXXX](#)).

XXXX

Weblate identifies the units in the XLIFF file by `resname` attribute in case it is present and falls back to `id` (together with `file` tag if present).

The `resname` attribute is supposed to be human friendly identifier of the unit making it more suitable for Weblate to display instead of `id`. The `resname` has to be unique in the whole XLIFF file. This is required by Weblate and is not covered by the XLIFF standard - it does not put any uniqueness restrictions on this attribute.

Typical Weblate *Component configuration* for bilingual XLIFF

XXXXXXXX	<code>localizations/*.xliff</code>
XXXXXXXXXXXXXXXXXXXX	<i>Empty</i>
XXXXXXXXXXXXXXXXXXXX	<code>localizations/en-US.xliff</code>
XXXXXX	<i>XLIFF Translation File</i>

Typical Weblate *Component configuration* for monolingual XLIFF

File mask	<code>localizations/*.xliff</code>
XXXXXXXXXXXXXXXXXXXX	<code>localizations/en-US.xliff</code>
XXXXXXXXXXXXXXXXXXXX	<code>localizations/en-US.xliff</code>
XXXXXX	<i>XLIFF Translation File</i>

XX:

XLIFF Wikipedia XX: [XXLIFF](#)font attribute in XLIFF 1.2 [maxwidth](#) attribute in XLIFF 1.2

Java properties

Native Java format for translations.

Java properties are usually used as monolingual translations.

Weblate supports ISO-8859-1, UTF-8 and UTF-16 variants of this format. All of them support storing all Unicode characters, it is just differently encoded. In the ISO-8859-1, the Unicode escape sequences are used (for example `zkou\u0161ka`), all others encode characters directly either in UTF-8 or UTF-16.

XX: Loading escape sequences works in UTF-8 mode as well, so please be careful choosing the correct encoding set to match your application needs.

Typical Weblate *Component configuration*

XXXXXXXX	<code>src/app/Bundle_*.properties</code>
XXXXXXXXXXXXXXXXXXXX	<code>src/app/Bundle.properties</code>
XXXXXXXXXXXXXXXXXXXX	<i>Empty</i>
XXXXXX	<i>Java Properties (ISO-8859-1)</i>

XX:

Java properties Wikipedia XX: [Mozilla and Java properties files](#)[Java](#) [XXXX](#) [XXXXXXXXXX](#)[XXXXXXXXXXXXXXXXXXXX](#)

GWT [Properties](#)

Native GWT format for translations.

GWT properties are usually used as monolingual translations.

Typical Weblate	Component configuration
Properties	src/app/Bundle_*.properties
Properties	src/app/Bundle.properties
Properties	Empty
Properties	GWT Properties

[Properties](#):

[GWT localization guide](#)[GWT Internationalization Tutorial](#)[Mozilla and Java properties files](#)[Java Properties](#)

INI translations

[Properties](#) 4.1 [Properties](#).

INI file format for translations.

INI translations are usually used as monolingual translations.

Typical Weblate	Component configuration
Properties	language/*.ini
Properties	language/en.ini
Properties	Empty
Properties	INI File

[Properties](#): Weblate only extracts keys from sections within an INI file. In case your INI file lacks sections, you might want to use [Joomla translations](#) or [Java properties](#) instead.

[Properties](#):

[INI Files](#)[Java properties](#)[Joomla translations](#)[Inno Setup INI Properties](#)

Inno Setup INI [Properties](#)

[Properties](#) 4.1 [Properties](#).

[Properties](#) Inno Setup INI [Properties](#)

Inno Setup INI [Properties](#)

[Properties](#): The only notable difference to [INI translations](#) is in supporting %n and %t placeholders for line break and tab.

Typical Weblate	Component configuration
Properties	language/*.isl
Properties	language/en.isl
Properties	Empty
Properties	Inno Setup INI File

[Properties](#): Only Unicode files (.isl) are currently supported, ANSI variant (.isl) is currently not supported.

[Properties](#):

[INI Files](#)[Joomla translations](#)[INI translations](#)

Joomla translations

2.12

Native Joomla format for translations.

Joomla translations are usually used as monolingual translations.

```
Typical Weblate Component configuration
language/*com_foobar.ini
language/en-GBcom_foobar.ini
Empty
Joomla Language File
```

2.12:

Specification of Joomla language files Mozilla and Java properties filesINI translationsInno Setup INI

Qt Linguist .ts

Translation format used in Qt based applications.

Qt Linguist files are used as both bilingual and monolingual translations.

```
Typical Weblate Component configuration when using as bilingual
i18n/app.*.ts
Empty
i18n/app.de.ts
Qt Linguist Translation File
```

```
Typical Weblate Component configuration when using as monolingual
i18n/app.*.ts
i18n/app.en.ts
i18n/app.en.ts
Qt Linguist Translation File
```

2.12:

Qt Linguist manual Qt .tsBilingual and monolingual formats

Android string resources

Android specific file format for translating applications.

Android string resources are monolingual, the `res/values/strings.xml` is stored in a different location from the others

```
Typical Weblate Component configuration
res/values*/strings.xml
res/values/strings.xml
Empty
Android String Resource
```

2.12:

Android string resources documentation Android string resources

Android *string-array* structures are not currently supported. To work around this, you can break your string arrays apart:

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

become:

```
<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>
```

The *string-array* that points to the *string* elements should be stored in a different file, and not be made available for translation.

This script may help pre-process your existing strings.xml files and translations: <https://gist.github.com/paour/11291062>

Apple iOS strings

Apple specific file format for translating applications, used for both iOS and iPhone/iPad application translations.

Apple iOS strings are usually used as bilingual translations.

Typical Weblate *Component configuration*

```
Resources/*.lproj/Localizable.strings
Resources/en.lproj/Localizable.strings or Resources/Base.lproj/
Localizable.strings
Empty
iOS Strings (UTF-8)
```

🔗:

Apple "strings files" documentation 🔗 Mac OSX strings

PHP 🔗🔗🔗

PHP translations are usually monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Example file:

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

Typical Weblate *Component configuration*

```
lang/*/texts.php
lang/en/texts.php
lang/en/texts.php
PHP strings
```

Laravel PHP 🔗🔗🔗

🔗🔗🔗 4.1 🔗🔗.

The Laravel PHP localization files are supported as well with plurals:

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

🔗:

PHP🔗Localization in Laravel

JSON files

2.0.

2.16: Since Weblate 2.16 and with [translate-toolkit](#) at-least 2.2.4, nested structure JSON files are supported as well.

4.3: The structure of JSON file is properly preserved even for complex situations which were broken in prior releases.

JSON format is used mostly for translating applications implemented in JavaScript.

Weblate currently supports several variants of JSON translations:

Simple key / value files, used for example by *vue-i18n* or *react-intl*.

Files with nested keys.

JSON i18next files

go-i18n JSON files

WebExtension JSON

ARB File

JSON translations are usually monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Example file:

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

Nested files are supported as well (see above for requirements), such a file can look like:

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

: The *JSON file* and *JSON nested structure file* can both handle same type of files. Both preserve existing JSON structure when translating.

The only difference between them is when adding new strings using Weblate. The nested structure format parses the newly added key and inserts the new string into the matching structure. For example `app.name` key is inserted as:

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

Typical Weblate *Component configuration*

	<code>langs/translation-*.json</code>
	<code>langs/translation-en.json</code>
	<code>Empty</code>
	<i>JSON nested structure file</i>

:

JSONJSON

JSON i18next files

2.17 Since Weblate 2.17 and with [translate-toolkit](#) at-least 2.2.5, i18next JSON files with plurals are supported as well.

[i18next](#) is an internationalization framework written in and for JavaScript. Weblate supports its localization files with features such as plurals.

i18next translations are monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

2.17: Weblate [2.17](#) i18next JSON v3 [2.17](#) v2 [2.17](#) v1 [2.17](#)

Example file:

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

Typical Weblate *Component configuration*

```
langfiles      langs/*.json
default_lang   langs/en.json
empty          Empty
i18next_json   i18next JSON file
```

2.17:

JSON [i18next JSON Format](#) [2.17](#) [2.17](#)

go-i18n JSON files

4.1.

go-i18n translations are monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

2.17: Weblate supports the go-i18n JSON v1 format, for flat JSON formats please use [JSON files](#). The v2 format with hash is currently not supported.

Typical Weblate *Component configuration*

```
langfiles      langs/*.json
default_lang   langs/en.json
empty          Empty
go_i18n_json   go-i18n JSON file
```

2.17:

JSON [go-i18n](#) [2.17](#) [2.17](#)

ARB File

4.1.

ARB translations are monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Typical Weblate *Component configuration*

```
langfiles      lib/l10n/intl_*.arb
default_lang   lib/l10n/intl_en.arb
empty          Empty
arb_file       ARB file
```

??:

JSON?Application Resource Bundle Specification ?Internationalizing Flutter apps ?JSON

WebExtension JSON

???? 2.16 ????: This is supported since Weblate 2.16 and with [translate-toolkit](#) at-least 2.2.4.

File format used when translating extensions for Mozilla Firefox or Google Chromium.

?: While this format is called JSON, its specification allows to include comments, which are not part of JSON specification. Weblate currently does not support file with comments.

Example file:

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

Typical Weblate *Component configuration*

```
???? _locales/*/messages.json
???? _locales/en/messages.json
???? Empty
???? WebExtension JSON file
```

??:

JSON?Google chrome.i18n ?Mozilla Extensions Internationalization

.XML resource files

???? 2.3 ???.

A .XML resource (.resx) file employs a monolingual XML file format used in Microsoft .NET applications. It is interchangeable with .resw, when using identical syntax to .resx.

Typical Weblate *Component configuration*

```
???? Resources/Language.*.resx
???? Resources/Language.resx
???? Empty
???? .NET resource file
```

??:

.NET Resource files (.resx)?ref:addon-weblate.cleanup.generic

CSV files

2.4

CSV files can contain a simple list of source and translation. Weblate supports the following files:

Files with header defining fields (location, source, target, ID, fuzzy, context, translator_comments, developer_comments). This is the recommended approach, as it is the least error prone. Choose *CSV file* as a file format.

Files with two fields—source and translation (in this order). Choose *Simple CSV file* as a file format.

Headerless files with fields in order defined by the [translate-toolkit](#): location, source, target, ID, fuzzy, context, translator_comments, developer_comments. Choose *CSV file* as a file format.

Remember to define `CSV` when your files are monolingual (see *Bilingual and monolingual formats*).

Warning: The CSV format currently automatically detects the dialect of the CSV file. In some cases the automatic detection might fail and you will get mixed results. This is especially true for CSV files with newlines in the values. As a workaround it is recommended to omit quoting characters.

Example file:

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

Typical Weblate *Component configuration* for bilingual CSV

```
locale/*.csv
Empty
locale/en.csv
CSV file
```

Typical Weblate *Component configuration* for monolingual CSV

```
locale/*.csv
locale/en.csv
locale/en.csv
CSV
```

Warning:
CSV

YAML files

2.9

The plain YAML files with string keys and values. Weblate also extract strings from lists or dictionaries.

Example of a YAML file:

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

Typical Weblate *Component configuration*

```
translations/messages.*.yaml
translations/messages.en.yaml
Empty
YAML file
```

Warning:
YAML
Ruby YAML files

Ruby YAML files

2.9.2022.

Ruby i18n YAML files with language as root node.

Example Ruby i18n YAML file:

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

Typical Weblate *Component configuration*

```
translations/translations/messages.*.yaml
translations/translations/messages.en.yaml
Empty
Ruby YAML file
```

YAML:

YAML files

DTD files

2.18.2022.

Example DTD file:

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

Typical Weblate *Component configuration*

```
locale/*.dtd
locale/en.dtd
Empty
DTD file
```

DTD:

Mozilla DTD format

Flat XML files

3.9.2022.

Example of a flat XML file:

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

Typical Weblate *Component configuration*

```
locale/*.xml
locale/en.xml
Empty
Flat XML file
```

XML:

Flat XML

Windows RC files

Webblate 4.1 beta: Support for Windows RC files has been rewritten.

beta: Support for this format is currently in beta, feedback from testing is welcome.

Example Windows RC file:

```
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
```

Typical Weblate *Component configuration*

```
lang/*.*rc
lang/en-US.rc
lang/en-US.rc
RC file
```

beta:

Windows RC files

Webblate 4.1 beta

Webblate 3.5 beta.

Metadata used for publishing apps in various app stores can be translated. Currently the following tools are compatible:

Triple-T gradle-play-publisher

Fastlane

F-Droid

The metadata consists of several textfiles, which Weblate will present as separate strings to translate.

Typical Weblate *Component configuration*

```
fastlane/android/metadata/*
fastlane/android/metadata/en-US
fastlane/android/metadata/en-US
App store metadata files
```

beta: In case you don't want to translate certain strings (for example changelogs), mark them read-only (see [Webblate 4.1 beta](#)). This can be automated by the [beta](#).

Subtitle files

Webblate 3.7 beta.

Weblate [Webblate 3.7 beta](#)

SubRip subtitle file (*.srt)

MicroDVD subtitle file (*.sub)

Advanced Substation Alpha subtitles file (*.ass)

Substation Alpha subtitle file (*.ssa)

Typical Weblate *Component configuration*

```
##### path/*.srt
##### path/en.srt
##### path/en.srt
##### SubRip subtitle file
```

##:

Subtitles

Excel Open XML

3.2 ##.

Excel Open XML (.xlsx) files can be imported and exported.

When uploading XLSX files for translation, be aware that only the active worksheet is considered, and there must be at least a column called `source` (which contains the source string) and a column called `target` (which contains the translation). Additionally there should be the column called `context` (which contains the context path of the translation string). If you use the XLSX download for exporting the translations into an Excel workbook, you already get a file with the correct file format.

HTML

4.1 ##.

##: Support for this format is currently in beta, feedback from testing is welcome.

The translatable content is extracted from the HTML files and offered for the translation.

##:

HTML

#####

4.6 ##.

##: Support for this format is currently in beta, feedback from testing is welcome.

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

3 ##:

#####

DokuWiki #####

MediaWiki #####

##:

Simple Text Documents

OpenDocument Format

4.1

Support for this format is currently in beta, feedback from testing is welcome.

The translatable content is extracted from the OpenDocument files and offered for the translation.

OpenDocument Format

IDML Format

4.1

Support for this format is currently in beta, feedback from testing is welcome.

The translatable content is extracted from the Adobe InDesign Markup Language files and offered for the translation.

TermBase eXchange

4.5

TBX XML

Typical Weblate <i>Component configuration</i>	
	tbx/*.tbx
	Empty
	Empty
	TermBase eXchange

TBX Wikipedia <https://en.wikipedia.org/wiki/TermBase_eXchange>_`\_TBX`

Most formats supported by [translate-toolkit](#) which support serializing can be easily supported, but they did not (yet) receive any testing. In most cases some thin layer is needed in Weblate to hide differences in behavior of different [translate-toolkit](#) storages.

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

Translation Related File Formats

Weblate [Git](#) [GitHub](#) [Gerrit](#) [Subversion](#) [Mercurial](#)

VCS Weblate <https://github.com/WeblateOrg/weblate.git> URL: [URL](#)

Hosted Weblate

[illegible]

GitHub `weblate` `5`

SSH [?] [?] [?] [?] [?]

```

#####
SSH #####SSH ? upstream #####Weblate
SSH #####: Weblate SSH #####

```

📄: GitHub 📄📄📄📄📄📄 1 📄📄📄📄📄📄📄📄📄📄: *GitHub 📄📄📄📄 📄📄 Hosted Weblate 📄📄📄📄📄📄📄📄📄📄*

```

[?]Webplate [?]: SSH [?]
[?]Webplate [?]

```

W Weblate

DashboardProjectsLanguagesChecks

Manage / SSH keys

Weblate statusBackupsTranslation memoryPerformance reportSSH keysAlertsRepositoriesUsersAppearance

ToolsBilling

Public SSH key

Webblate currently uses this SSH key:

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAQCdKuMKvxqBGmPWC0Tnn7az4C/VTyPhqICr8FxBobfhVGHaVufSxcFi3WmGvsq3+sVXvqPoiIG9nEU11StMje8KBwNGLghtvHM

Download private key

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8

Add host key

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

Hostname

Hostname

Port

Port

Submit

Weblate SSH

Weblate [About](#)

[SSH keys](#) Weblate

SSH:

SSH: Weblate SSH

SSH

Weblate SSH

`gitlab.com` `ssh-rsa` `AAAAB3NzaC1yc2EAAAADAQABAAQCA...` `ssh-rsa` `AAAAB3NzaC1yc2EAAAADAQABAAQCA...`

Weblate

Dashboard

Projects

Languages

Checks

Manage / SSH keys

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

Weblate status

Backups

Translation memory

Performance report

SSH keys

Alerts

Repositories

Users

Appearance

Tools

Billing

Public SSH key

Weblate currently uses this SSH key:

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCA...

Download private key

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8

Add host key

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

Hostname

github.com

Port

Port

Submit

GitHub

```
SSH [REDACTED]: SSH [REDACTED] SSH [REDACTED] GitHub
[REDACTED] 1 [REDACTED]
```

```

##### push ##### upstream
#####

```

```

$ curl -s https://api.github.com/user/13280763/tokens | jq
{"tokens": [{"token": "gho_1234567890123456789012345678901234567890", "label": "command-line use"}]}

```

```

XXXXXXXXXXXXXXXXXXXXWeblate XXXXXXXXXXXXXXXXXXXX Weblate XXXXXXXX SSH XXXXXX: Weblate SSH
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Hosted Weblate XXXXXXXXXXXXXXXX
weblate XXXXXXXXXXXXXXXX

```

??:

Hosted Weblate [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)

Weblate URL

```

##### @webate :// project/component#####
(2) #####VCS#####

```

??: Removing main component also removes linked components.

Webplate URL

□□□□:

????????????????????

1

Webplate 1: *Git exporter*

`??????: Git ?`

HTTPS [?][?][?][?][?]

000000 HTTPS 0000000000000000 URL 0000000000000000000000000000 URL
 00000000000000000000 Weblate 000000000000000000 URL 0000000000000000

```

##### GitHub URL #####: https://user:your_access_token@github.com/
WeblateOrg/weblate.git

```

URL `https://user%40example.com:%24password%23@bitbucket.org/...`

□□□□□□□

????? ????????? HTTP/HTTPS VCS ?????????????????????VCS ?????????????

```

#####cURL documentation ##### http_proxy? https_proxy???? all_proxy
#####VCS #####:

```

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

```

[??: ????Webplate ??: ???? HOME=$DATA_DIR/home`?:
DATA DIR?Webplate ? Git ?

```

??:

[The cURL manpage](#) [Git config documentation](#)

Git

##:

#####

Git

Git

##: upstream #####

Git

Weblate ##### VCS ##### HOME=\$DATA_DIR/home ##: DATA_DIR#####
DATA_DIR/home/.git #####

Git

remote helpers #####
#####Bazaar ? Mercurial ##### GitHub #####GitHub: git-remote-hg ? git-remote-bzr
#####: ~/bin#####
Weblate #####
Bazaar ##### Launchpad ## gnuhello #####:

bzr::lp:gnuhello

Mercurial ##### selenic.com ? hello #####:

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

##: Git ##### Mercurial ##### tip
#####

GitHub

2.3 ##.

#####GitHub API ##### Git #####

Git #####GitHub ## ##### Git #####

##:

Pushing changes from Weblate

GitHub

GitHub ##### 1

#####API #####

##:

GITHUB_USERNAME?GITHUB_TOKEN?GITHUB_CREDENTIALS

GitLab

3.9

GitLab API `gitlab.com` `Git`

Git `Git` `GitLab`

##:

Pushing changes from Weblate

GitLab `gitlab.com` `gitlab.com`

GitLab `gitlab.com` 1 `gitlab.com`

`API`

##:

`GITLAB_USERNAME` `GITLAB_TOKEN` `GITLAB_CREDENTIALS`

Pagure

4.3.2

Pagure API `pagure.org` `Git`

Git `Git` `Pagure`

##:

Pushing changes from Weblate

`pagure.org` Pagure `pagure.org`

Pagure `pagure.org` 1 `pagure.org`

`API`

##:

`PAGURE_USERNAME` `PAGURE_TOKEN` `PAGURE_CREDENTIALS`

Gerrit

2.2

`git-review` `Gerrit` `Git`

Gerrit

Mercurial

2.1

Mercurial `Weblate` 1 `VCS`

##: Mercurial `Weblate`

##:

`mercurial` `weblate`

Subversion

2.8.2.

Weblate `git-svn` `subversion` `Git` `Subversion` `Perl`

URL: Weblate `Subversion` `branches/``tags/``trunk/` `git-svn` `documentation` `URL`

2.19: `DATA_DIR` `DATA_DIR`

Subversion

Weblate `DATA_DIR` `'svn'` `$HOME` `DATA_DIR`

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

URL:

`DATA_DIR`

`DATA_DIR`

3.8.2.

Weblate `VCS` `Weblate`

Weblate `Git` `VCS` `Weblate`

Weblate's REST API

2.6: REST API `Weblate 2.6`

API `/api/` `URL` `Django REST framework` `Weblate`

`API`

`API` `100`

ANY /

API `Web`

format -- `Accept` `REST` `json` `api`

Accept -- `Accept`

Authorization -- `Authorization`

Content-Type -- `Accept`

Allow -- `HTTP` `JSON`

detail (`string`) -- `200 OK` `HTTP`

count (`int`) -- `count`

next (*string*) -- ?????? ????????????? URL
previous (*string*) -- ?????? ????????????? URL
results (*array*) -- ?????? ??????
url (*string*) -- ????????? API ????????????? URL
web_url (*string*) -- ????????? Web ????????????? URL
 200 OK -- ?????????????
 201 Created -- when a new object was created successfully
 204 No Content -- when an object was created successfully
 400 Bad Request -- ?????? ?????????????
 403 Forbidden -- ?????????????
 429 Too Many Requests -- ?????????????

????

?????:

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

????:

```
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 ??:

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

????????

POST ????????????? application/x-www-form-urlencoded????? JSON? application/json????????

???? ?????:

```
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 ??????:

(XXXXXXXXXX)

```
{
  "projects": "http://example.com/api/projects/",
  "components": "http://example.com/api/components/",
  "translations": "http://example.com/api/translations/",
  "languages": "http://example.com/api/languages/"
}
```

????

XXXX 4.0 ????.

GET /api/users/
XX

??:

???? XXXXXXXXXXXXXXXXXXXX GET /api/users/(str:username)/ XXXXXXXXXXXXXXXX

POST /api/users/
XXXXXXXXXXXXXXXXXXXX
XXXXXX

username (string) -- XXXXX

full_name (string) -- XXXXXXXXXXXX

email (string) -- XXXXXXXXXXXXXXXX

is_superuser (boolean) -- XXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXX

is_active (boolean) -- XXXXXXXXXXXXXXXX XXXXXXXX

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

username (string) -- XXXXXXXXXXXX
XXXXXX JSON XXXXXXXX

username (string) -- XXXXXXXXXXXX

full_name (string) -- XXXXXXXXXXXX

email (string) -- XXXXXXXXXXXXXXXX

is_superuser (boolean) -- XXXXXXXXXXXXXXXXXXXXXXXX

is_active (boolean) -- XXXXXXXXXXXXXXXX

date_joined (string) -- XXXXXXXX

groups (array) -- XXXXXXXXXXXXXXXXXXXXXXXX: GET /api/groups/(int:id) /

JSON XXXXXX:

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

PUT /api/users/(str: username) /
XXXXXXXXXXXXXXXXXXXX
XXXXXX

username (string) -- XXXXXXXXXXXX

```

##### JSON #####
username(string)-- #####
full_name(string)-- #####
email(string)-- #####
is_superuser(boolean)-- #####
is_active(boolean)-- #####
date_joined(string)-- #####
PATCH /api/users/(str: username) /
#####
#####
username(string)-- #####
##### JSON #####
username(string)-- #####
full_name(string)-- #####
email(string)-- #####
is_superuser(boolean)-- #####
is_active(boolean)-- #####
date_joined(string)-- #####
DELETE /api/users/(str: username) /
#####
#####
username(string)-- #####
POST /api/users/(str: username) /groups/
#####
#####
username(string)-- #####
#### #####
string group_id-- ##### ID
GET /api/users/(str: username) /statistics/
#####
#####
username(string)-- #####
##### JSON #####
translated(int)-- #####
suggested(int)-- #####
uploaded(int)-- #####
commented(int)-- #####
languages(int)-- #####
GET /api/users/(str: username) /notifications/
#####
#####
username(string)-- #####
POST /api/users/(str: username) /notifications/
#####
#####
username(string)-- #####
##### JSON #####
notification(string)-- #####
scope(int)-- #####
frequency(int)-- #####
GET /api/users/(str: username) /notifications/
int: subscription_id/ #####
#####

```

```

username(string)-- ?????????
subscription_id(int)-- ????? ID
PUT /api/users/(str: username)/notifications/
int: subscription_id/ ?????????????????????????????????
????
username(string)-- ?????????
subscription_id(int)-- ????? ID
???? JSON ?????
notification(string)-- ?????
scope(int)-- ?????????
frequency(int)-- ?????
PATCH /api/users/(str: username)/notifications/
int: subscription_id/ ?????????????????????????????????
????
username(string)-- ?????????
subscription_id(int)-- ????? ID
???? JSON ?????
notification(string)-- ?????
scope(int)-- ?????????
frequency(int)-- ?????
DELETE /api/users/(str: username)/notifications/
int: subscription_id/ ?????????????????????????????????
????
username(string)-- ?????????
subscription_id-- ?????
subscription_id-- int

```

????

???? 4.0 ???.

```

GET /api/groups/
????????????????????????????????????????????????????????????????????????????????????

```

??:

```

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

```

```

POST /api/groups/
????????????????
????

```

```

name(string)-- ?????
project_selection(int)-- ?????????????????????????????
language_selection(int)-- ?????????????????????

```

```

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

```

```

id(int)-- ??? ID
???? JSON ?????

```

```

name(string)-- ?????
project_selection(int)-- ?? ?????????????
language_selection(int)-- ?? ?????

```

```

roles(array)-- ??????????: GET /api/roles/(int:id)/
projects(array)-- ??????????: GET /api/projects/(string:project)/
components(array)-- ??????????: GET /api/components/(string:project)/
(string:component)/

```

componentlist (array) -- `????????????? ?????: GET /api/component-lists/`
(*str:slug*)/

JSON `?????`:

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

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

id (*int*) -- `???? ID`
`????? JSON` `?????`

name (*string*) -- `?????`

project_selection (*int*) -- `?? ?????`

language_selection (*int*) -- `?? ?????`

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

id (*int*) -- `???? ID`
`????? JSON` `?????`

name (*string*) -- `?????`

project_selection (*int*) -- `?? ?????`

language_selection (*int*) -- `?? ?????`

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

id (*int*) -- `???? ID`

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

id (*int*) -- `???? ID`
`???? ?????`

string *role_id* -- `????? ID`

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

id (*int*) -- `???? ID`
`???? ?????`

string *component_id* -- `????? ID`

name (*string*) -- `string`
permissions (*array*) -- `array`
JSON `object`:

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

PUT `/api/roles/{id}` /
`PUT`
`PUT`
id (*int*) -- `int` ID
`JSON` `object`
name (*string*) -- `string`
permissions (*array*) -- `array`
PATCH `/api/roles/{id}` /
`PATCH`
`PATCH`
id (*int*) -- `int` ID
`JSON` `object`
name (*string*) -- `string`
permissions (*array*) -- `array`
DELETE `/api/roles/{id}` /
`DELETE`
`DELETE`
id (*int*) -- `int` ID

`??`

GET `/api/languages/`
`GET`
`??`:
`GET /api/languages/{language}` /
POST `/api/languages/`
`POST`
`POST`
code (*string*) -- `string`
name (*string*) -- `string`
direction (*string*) -- `string`
plural (*object*) -- `object`
GET `/api/languages/{language}` /
`GET`
`GET`
language (*string*) -- `string`
`JSON` `object`
code (*string*) -- `string`
direction (*string*) -- `string`
plural (*object*) -- `object`
aliases (*array*) -- `array`
JSON `object`:

```
{
  "code": "en",
  "direction": "ltr",
  "name": "English",
  "plural": {
    "id": 75,
    "source": 0,
    "number": 2,
    "formula": "n != 1",
    "type": 1
  },
  "aliases": [
    "english",
    "en_en",
    "base",
    "source",
    "eng"
  ],
  "url": "http://example.com/api/languages/en/",
  "web_url": "http://example.com/languages/en/",
  "statistics_url": "http://example.com/api/languages/en/statistics/"
}
```

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

????????????????

??????

language (string) -- ??????

?????? JSON ???????

name (string) -- ???

direction (string) -- ???????

plural (object) -- ?????????

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

????????????????

??????

language (string) -- ??????

?????? JSON ???????

name (string) -- ???

direction (string) -- ???????

plural (object) -- ?????????

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

????????????

??????

language (string) -- ??????

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

????????????????

??????

language (string) -- ??????

?????? JSON ???????

total (int) -- ???????

total_words (int) -- ??????

last_change (timestamp) -- ?????????

recent_changes (int) -- ??????

translated (int) -- ?????????

translated_percent (float) -- ?????????

translated_words (int) -- ?????????

translated_words_percent (int) -- ?????????

translated_chars (int) -- ?????????

```
translated_chars_percent (int) -- ??????????
total_chars (int) -- ?????
fuzzy (int) -- ??????????????????
fuzzy_percent (int) -- ??????????????????
failing (int) -- ??????????
failing -- ??????????
```

??????

```
GET /api/projects/
????????????????????????????
```

??:

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

```
POST /api/projects/
????? 3.9 ???.
```

```
????????????????????
?????
```

```
name (string) -- ?????
slug (string) -- ??????????
web (string) -- ?????? Web ???
```

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

```
project (string) -- ?????? URL ???
????? JSON ?????
```

```
name (string) -- ?????
slug (string) -- ??????????
web (string) -- ?????? Web ???
```

```
components_list_url (string) -- ?????????? ?? URL??? GET /api/projects/
(string:project)/components/
```

```
repository_url (string) -- ??????????????URL??? GET /api/projects/
(string:project)/repository/
```

```
changes_list_url (string) -- ?????????? URL??? GET /api/projects/(string:project)/
changes/
```

```
translation_review (boolean) -- ?????
```

```
source_review (boolean) -- ??????????
```

```
set_language_team (boolean) -- "Language-Team" ?????
```

```
enable_hooks (boolean) -- ?????
```

```
instructions (string) -- ?????????
```

```
language_aliases (string) -- ?????
```

JSON ?????:

```
{
  "name": "Hello",
  "slug": "hello",
  "url": "http://example.com/api/projects/hello/",
  "web": "https://weblate.org/",
  "web_url": "http://example.com/projects/hello/"
}
```

```
PATCH /api/projects/(string: project) /
????? 4.3 ???.
```

```
PATCH ??????????????????
?????
```

```

project (string) -- URL
component (string) -- URL
PUT /api/projects/(string: project) /
4.3
PUT
project (string) -- URL
DELETE /api/projects/(string: project) /
3.9
project (string) -- URL
GET /api/projects/(string: project) /changes/
GET /api/changes/
project (string) -- URL
JSON
results (array) -- GET /api/changes/(int:id) /
GET /api/projects/(string: project) /repository/
VCS
GET /api/components/(string:project)/(string:component)/repository/
project (string) -- URL
JSON
needs_commit (boolean) --
needs_merge (boolean) -- upstream
needs_push (boolean) --
JSON:

```

```

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

```

```

POST /api/projects/(string: project) /repository/
VCS
project (string) -- URL
JSON
operation (string) -- : push, pull, commit, reset, cleanup, file-sync
JSON
result (boolean) --
CURL:

```

```

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

```

JSON:

```

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

```

(`)`

(XXXXXXXXXX)

```
Content-Length: 20
{"operation":"pull"}
```

JSON XXXXXXX:

```
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/
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXX
```

```
project (string) -- XXXXXX URL XXXX
XXXXXX JSON XXXXXXX
```

```
results (array) -- XXXXXXX XXXXXXXXXXXXXXX: GET /api/components/(string:project)/
(string:component)/
```

```
POST /api/projects/(string: project)/components/
XXXXXX 3.9 XXXX.
```

```
XXXXXX 4.3 XXXX: zipfile XXXX docfile XXXXXXXVCS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX: XXXX
XXXXXX
```

```
XXXXXX 4.6 XXXX: The cloned repositories are now automatically shared within a project using Weblate XXXX URL.
Use disable_autoshare to turn off this.
```

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
XXX: 1 XX VCS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXWeblate XXXX URL XXXXXXX
```

```
XX: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX task_url
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
XXXXXX
```

```
project (string) -- XXXXXX URL XXXX
XXXXXX XXXXXXX
```

```
file zipfile -- XXXXXXXXXXXXX Weblate XXXXXXXXXXXX ZIP XXXX
```

```
file docfile -- XXXXXXXXXXXXX
```

```
boolean disable_autoshare -- Disables automatic repository sharing via Weblate XXXX URL.
```

```
XXXXXX JSON XXXXXXX
```

```
result (object) -- XXXXXXXXXXXXXXX XXXXXXXXXXXXX: GET /api/components/(string:project)/
(string:component)/
```

JSON can not be used when uploading the files using the zipfile and docfile parameters. The data has to be uploaded as *multipart/form-data*.

CURL XXXX XXXXXXX:

```
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 XXXXXXX:

```
curl \
  --data-binary '{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "slug": "weblate",
    "repo": "file:///home/nijel/work/weblate-hello",
    "template": "",
    "new_base": "",
    "vcs": "git"
  }' \
  -H "Content-Type: application/json" \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

JSON ????????:

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

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "vcs": "git"
}
```

JSON ????????:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
```

(?????????)

(XXXXXXXXXXXX)

```
        "direction": "ltr",
        "name": "English",
        "url": "http://example.com/api/languages/en/",
        "web_url": "http://example.com/languages/en/"
    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
},
"repo": "file:///home/nijel/work/weblate-hello",
"template": "",
"new_base": "",
"url": "http://example.com/api/components/hello/weblate/",
"vcs": "git",
"web_url": "http://example.com/projects/hello/weblate/"
}
```

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

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXX 3.8 XXX.

XXXXXX

project (*string*) -- XXXXXX URL XXXX

XXXXXX JSON XXXXXX

results (*array*) -- XXXXXXXXXXXXXXXX

language (*string*) -- XXX

code (*string*) -- XXXXX

total (*int*) -- XXXXXX

translated (*int*) -- XXXXXXXXXXXX

translated_percent (*float*) -- XXXXXXXXXXXXX

total_words (*int*) -- XXXXX

translated_words (*int*) -- XXXXXXXXXXXX

words_percent (*float*) -- XXXXXXXXXXXX

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

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXX 3.8 XXX.

XXXXXX

project (*string*) -- XXXXXX URL XXXX

XXXXXX JSON XXXXXX

total (*int*) -- XXXXXX

translated (*int*) -- XXXXXXXXXXXX

translated_percent (*float*) -- XXXXXXXXXXXXX

total_words (*int*) -- XXXXX

translated_words (*int*) -- XXXXXXXXXXXX

words_percent (*float*) -- XXXXXXXXXXXX

???????

GET /api/components/
????????????????????????

??:

??????? ???? GET /api/components/(string:project)/
(string:component)/ ?????

GET /api/components/(string: project) /
string: component/ ?????
????

project (string) -- ????? URL ?????

component (string) -- ????? URL ????
???? JSON ?????

project (object) -- ?????: GET /api/projects/(string:project) /

name (string) -- ?????

slug (string) -- ?????

vcs (string) -- ?????

repo (string) -- ?????

git_export (string) -- ????? URL

branch (string) -- ?????

push_branch (string) -- ????? push

filemask (string) -- File mask

template (string) -- ?????

edit_template (string) -- ?????

intermediate (string) -- ?????

new_base (string) -- ?????

file_format (string) -- ?????

license (string) -- ?????

agreement (string) -- ?????

new_lang (string) -- ?????

language_code_style (string) -- ????? ?????

source_language (object) -- ?????: GET /api/languages/
(string:language) /

push (string) -- ????? URL

check_flags (string) -- ?????

priority (string) -- ?????

enforced_checks (string) -- ?????

restricted (string) -- ?????

repoweb (string) -- ????? ?????

report_source_bugs (string) -- ?????

merge_style (string) -- ??? ?????

commit_message (string) -- Commit, add, delete, merge and addon messages

add_message (string) -- Commit, add, delete, merge and addon messages

delete_message (string) -- Commit, add, delete, merge and addon messages

merge_message (string) -- Commit, add, delete, merge and addon messages

addon_message (string) -- Commit, add, delete, merge and addon messages

allow_translation_propagation (string) -- ?????

enable_suggestions (string) -- ?????

suggestion_voting (string) -- ?????

```

suggestion_autoaccept (string) -- ????????
push_on_commit (string) -- ????????????????
commit_pending_age (string) -- ????????????????
auto_lock_error (string) -- ????????
language_regex (string) -- ????????
variant_regex (string) -- ????????????????
repository_url (string) -- ????? ???? URL???: GET /api/components/
(string:project)/(string:component)/repository/
translations_url (string) -- ??????? URL???: GET /api/components/
(string:project)/(string:component)/translations/
lock_url (string) -- ????????????? URL???: GET /api/components/(string:project)/
(string:component)/lock/
changes_list_url (string) -- ??????? URL???: GET /api/components/
(string:project)/(string:component)/changes/
task_url (string) -- ????????? ???? URL?????????????: GET /api/tasks/(str:uuid)/
JSON ?????:

```

```

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
      "name": "English",
      "url": "http://example.com/api/languages/en/",
      "web_url": "http://example.com/languages/en/"
    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
  },
  "source_language": {
    "code": "en",
    "direction": "ltr",
    "name": "English",
    "url": "http://example.com/api/languages/en/",
    "web_url": "http://example.com/languages/en/"
  },
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "url": "http://example.com/api/components/hello/weblate/",
  "vcs": "git",
  "web_url": "http://example.com/projects/hello/weblate/"
}

```

```

PATCH /api/components/(string: project) /
string: component / PATCH ????????????????????????
?????
project (string) -- ??????? URL ?????
component (string) -- ????????? URL ?????
source_language (string) -- ????????????????????????
????? JSON ?????

```

name (string) -- ?????????
slug (string) -- ??????????????
repo (string) -- VCS ????? URL
CURL :

```

curl \
  --data-binary '{"name": "new name"}' \
  -H "Content-Type: application/json" \
  -H "Authorization: Token TOKEN" \
  PATCH http://example.com/api/projects/hello/components/

```

JSON :

```

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

{
  "name": "new name"
}

```

JSON :

```

HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "new name",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
      "name": "English",
      "url": "http://example.com/api/languages/en/",
      "web_url": "http://example.com/languages/en/"
    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
  },
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "url": "http://example.com/api/components/hello/weblate/",
  "vcs": "git",
  "web_url": "http://example.com/projects/hello/weblate/"
}

```

PUT /api/components/ (string: *project*) /
string: *component* / **PUT** ??????????????????
 ?????

```

project (string) -- ?????? URL ????
component (string) -- ???????? URL ????
?????? JSON ??????
branch (string) -- VCS ??????
file_format (string) -- ??????????
filemask (string) -- ??????????????????
name (string) -- ?????????
slug (string) -- ??????????????
repo (string) -- VCS ?????? URL
template (string) -- base file for monolingual translations
new_base (string) -- base file for adding new translations
vcs (string) -- version control system

DELETE /api/components/ (string: project) /
string: component/ ???? 3.9 ??.
Deletes a component.
?????

project (string) -- ?????? URL ????
component (string) -- ???????? URL ????
GET /api/components/ (string: project) /
string: component/changes/ Returns a list of component changes. This is essentially a component scoped
GET /api/changes/ accepting same params.
?????

project (string) -- ?????? URL ????
component (string) -- ???????? URL ????
?????? JSON ??????
results (array) -- ?????????????????????: GET /api/changes/(int:id)/
GET /api/components/ (string: project) /
string: component/screenshots/ Returns a list of component screenshots.
?????

project (string) -- ?????? URL ????
component (string) -- ???????? URL ????
?????? JSON ??????
results (array) -- array of component screenshots; see GET /api/screenshots/\(int:id\)/
GET /api/components/ (string: project) /
string: component/lock/ Returns component lock status.
?????

project (string) -- ?????? URL ????
component (string) -- ???????? URL ????
?????? JSON ??????
locked (boolean) -- whether component is locked for updates
JSON ?????:

```

```

{
  "locked": false
}

```

```

POST /api/components/ (string: project) /
string: component/lock/ Sets component lock status.

Response is same as GET /api/components/\(string:project\)/\(string:component\)/lock/.
?????

project (string) -- ?????? URL ????
component (string) -- ???????? URL ????
?????? JSON ??????
lock -- Boolean whether to lock or not.

```

CURL `curl`:

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

JSON `{}:`

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

{"lock": true}
```

JSON `{}:`

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{"locked":true}
```

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

string: *component/repository/* Returns information about VCS repository status.

The response is same as for *GET /api/projects/(string:project)/repository/*.
`{}:`

project (string) -- `{}:` URL `{}:`

component (string) -- `{}:` URL `{}:`
`{}:` JSON `{}:`

needs_commit (boolean) -- `{}:`

needs_merge (boolean) -- `{}:` upstream `{}:`

needs_push (boolean) -- `{}:`

remote_commit (string) -- Remote commit information

status (string) -- VCS repository status as reported by VCS

merge_failure -- Text describing merge failure or null if there is none

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

string: *component/repository/* Performs the given operation on a VCS repository.

See *POST /api/projects/(string:project)/repository/* for documentation.
`{}:`

project (string) -- `{}:` URL `{}:`

component (string) -- `{}:` URL `{}:`
`{}:` JSON `{}:`

operation (string) -- `{}:`: push, pull, commit, reset, cleanup `{}:`
`{}:` JSON `{}:`

result (boolean) -- `{}:`

CURL `curl`:

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

JSON `{}:`

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

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

JSON **????????**:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{"result": true}
```

GET /api/components/(string: project) /
string: *component/monolingual_base/* Downloads base file for monolingual translations.
??????

project (*string*) -- **??????** URL **????**

component (*string*) -- **??????** URL **????**

GET /api/components/(string: project) /
string: *component/new_template/* Downloads template file for new translations.
??????

project (*string*) -- **??????** URL **????**

component (*string*) -- **??????** URL **????**

GET /api/components/(string: project) /
string: *component/translations/* Returns a list of translation objects in the given component.
??????

project (*string*) -- **??????** URL **????**

component (*string*) -- **??????** URL **????**
?????? JSON **??????**

results (*array*) -- array of translation objects; see *GET /api/translations/(string:project)/ (string:component)/(string:language)/*

POST /api/components/(string: project) /
string: *component/translations/* Creates new translation in the given component.
??????

project (*string*) -- **??????** URL **????**

component (*string*) -- **??????** URL **????**
?????? JSON **??????**

language_code (*string*) -- translation language code; see *GET /api/languages/ (string:language)/*
?????? JSON **??????**

result (*object*) -- new translation object created

CURL **??:**

```
curl \
  -d language_code=cs \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

JSON **????????**:

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

{"language_code": "cs"}
```

JSON ????????:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
  "failing_checks": 0,
  "failing_checks_percent": 0,
  "failing_checks_words": 0,
  "filename": "po/cs.po",
  "fuzzy": 0,
  "fuzzy_percent": 0.0,
  "fuzzy_words": 0,
  "have_comment": 0,
  "have_suggestion": 0,
  "is_template": false,
  "is_source": false,
  "language": {
    "code": "cs",
    "direction": "ltr",
    "name": "Czech",
    "url": "http://example.com/api/languages/cs/",
    "web_url": "http://example.com/languages/cs/"
  },
  "language_code": "cs",
  "id": 125,
  "last_author": null,
  "last_change": null,
  "share_url": "http://example.com/engage/hello/cs/",
  "total": 4,
  "total_words": 15,
  "translate_url": "http://example.com/translate/hello/weblate/cs/",
  "translated": 0,
  "translated_percent": 0.0,
  "translated_words": 0,
  "url": "http://example.com/api/translations/hello/weblate/cs/",
  "web_url": "http://example.com/projects/hello/weblate/cs/"
}
```

GET /api/components/(string: project) /
string: *component/statistics/* Returns paginated statistics for all translations within component.

?????? 2.7 ???
 ??????

project (*string*) -- ??????? URL ?????

component (*string*) -- ??????? URL ?????
 ?????? JSON ???????

results (*array*) -- array of translation statistics objects; see *GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/*

GET /api/components/(string: project) /
string: *component/links/* ?????????????????????????????????

?????? 4.5 ???.

????

project (*string*) -- ?????? URL ????

component (*string*) -- ?????? URL ????
???? JSON ?????

projects (*array*) -- ??????; ??: *GET /api/projects/(string:project)/*

POST /api/components/(string: project) /
string: *component/links/ ??????????????????????*

???? 4.5 ??.
????

project (*string*) -- ?????? URL ????

component (*string*) -- ?????? URL ????
???? ?????

string project_slug -- ???????????

DELETE /api/components/(string: project) /
string: *component/links/string: project_slug/ ??????????????????????*

???? 4.5 ??.
????

project (*string*) -- ?????? URL ????

component (*string*) -- ?????? URL ????

project_slug (*string*) -- ???????????????

??

GET /api/translations/

Returns a list of translations.

??:

Translation object attributes are documented at *GET /api/translations/(string:project)/ (string:component)/(string:language)/*.

GET /api/translations/(string: project) /
string: *component/string: language/* Returns information about a translation.

????

project (*string*) -- ?????? URL ????

component (*string*) -- ?????? URL ????

language (*string*) -- Translation language code
???? JSON ?????

component (*object*) -- component object; see *GET /api/components/(string:project)/ (string:component)/*

failing_checks (*int*) -- ???????????

failing_checks_percent (*float*) -- ???????????

failing_checks_words (*int*) -- ???????????

filename (*string*) -- translation filename

fuzzy (*int*) -- ???????????????????

fuzzy_percent (*float*) -- ???????????????????

fuzzy_words (*int*) -- number of words in fuzzy (marked for edit) strings

have_comment (*int*) -- number of strings with comment

have_suggestion (*int*) -- number of strings with suggestion

is_template (*boolean*) -- ???????????????

language (*object*) -- ??????????????: *GET /api/languages/(string:language) /*

language_code (*string*) -- language code used in the repository; this can be different from language code in the language object

last_author (*string*) -- name of last author

last_change (*timestamp*) -- last change timestamp
revision (*string*) -- revision hash for the file
share_url (*string*) -- URL for sharing leading to engagement page
total (*int*) -- `??????`
total_words (*int*) -- `??????`
translate_url (*string*) -- URL for translating
translated (*int*) -- `??????????`
translated_percent (*float*) -- `??????????????`
translated_words (*int*) -- `??????????`
repository_url (*string*) -- URL to repository status; see `GET /api/translations/(string:project)/(string:component)/(string:language)/repository/`
file_url (*string*) -- URL to file object; see `GET /api/translations/(string:project)/(string:component)/(string:language)/file/`
changes_list_url (*string*) -- URL to changes list; see `GET /api/translations/(string:project)/(string:component)/(string:language)/changes/`
units_list_url (*string*) -- URL to strings list; see `GET /api/translations/(string:project)/(string:component)/(string:language)/units/`
JSON `??????`:

```

{
  "component": {
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "new_base": "",
    "project": {
      "name": "Hello",
      "slug": "hello",
      "source_language": {
        "code": "en",
        "direction": "ltr",
        "name": "English",
        "url": "http://example.com/api/languages/en/",
        "web_url": "http://example.com/languages/en/"
      },
      "url": "http://example.com/api/projects/hello/",
      "web": "https://weblate.org/",
      "web_url": "http://example.com/projects/hello/"
    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "slug": "weblate",
    "template": "",
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "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",

```

`(??????????)`

(XXXXXXXXXXXX)

```
    "direction": "ltr",
    "name": "Czech",
    "url": "http://example.com/api/languages/cs/",
    "web_url": "http://example.com/languages/cs/"
  },
  "language_code": "cs",
  "last_author": "Weblate Admin",
  "last_change": "2016-03-07T10:20:05.499",
  "revision": "7ddfafe6daaf57fc8654cc852ea6be212b015792",
  "share_url": "http://example.com/engage/hello/cs/",
  "total": 4,
  "total_words": 15,
  "translate_url": "http://example.com/translate/hello/weblate/cs/",
  "translated": 4,
  "translated_percent": 100.0,
  "translated_words": 15,
  "url": "http://example.com/api/translations/hello/weblate/cs/",
  "web_url": "http://example.com/projects/hello/weblate/cs/"
}
```

DELETE /api/translations/(string: *project*) /
string: *component*/**string:** *language*/ XXXXX 3.9 XXX.

Deletes a translation.

XXXXXX

project (*string*) -- XXXXXX URL XXXX

component (*string*) -- XXXXXXXX URL XXXX

language (*string*) -- Translation language code

GET /api/translations/(string: *project*) /
string: *component*/**string:** *language*/**changes/** Returns a list of translation changes. This is essentially a translations-scoped **GET /api/changes/** accepting the same parameters.

XXXXXX

project (*string*) -- XXXXXX URL XXXX

component (*string*) -- XXXXXXXX URL XXXX

language (*string*) -- Translation language code

XXXXXX JSON XXXXXXXX

results (*array*) -- XXXXXXXXXXXXXXXXXXXXXXXX: **GET /api/changes/(int:id) /**

GET /api/translations/(string: *project*) /
string: *component*/**string:** *language*/**units/** Returns a list of translation units.

XXXXXX

project (*string*) -- XXXXXX URL XXXX

component (*string*) -- XXXXXXXX URL XXXX

language (*string*) -- Translation language code

q (*string*) -- Search query string XX (optional)

XXXXXX JSON XXXXXXXX

results (*array*) -- array of component objects; see **GET /api/units/(int:id) /**

POST /api/translations/(string: *project*) /
string: *component*/**string:** *language*/**units/** Add new monolingual unit.

XXXXXX

project (*string*) -- XXXXXX URL XXXX

component (*string*) -- XXXXXXXX URL XXXX

language (*string*) -- Translation language code

XXXXXX JSON XXXXXXXX

key (*string*) -- Name of translation unit

value (*string*) -- The translation unit value

XX:

XXXXXXXXXXadding-new-strings

POST /api/translations/ (string: project) /
string: component/string: language/autotranslate/ Trigger automatic translation.
?????

project (string) -- ?????? URL ?????

component (string) -- ?????? URL ?????

language (string) -- Translation language code
????? JSON ??????

mode (string) -- ??????

filter_type (string) -- Automatic translation filter type

auto_source (string) -- ??????

component (string) -- ???

engines (string) -- ??????

threshold (string) -- ??????

GET /api/translations/ (string: project) /
string: component/string: language/file/ Download current translation file as it is stored in the VCS
(without the **format** parameter) or converted to another format (see [????????](#)).

??: This API endpoint uses different logic for output than rest of API as it operates on whole file rather than on data.
Set of accepted format parameter differs and without such parameter you get translation file as stored in VCS.

??? ?????

format -- File format to use; if not specified no format conversion happens; supported file formats: po, mo, xiff, xiff11, tbx, csv, xlsx, json, aresource, strings
?????

project (string) -- ?????? URL ?????

component (string) -- ?????? URL ?????

language (string) -- Translation language code

POST /api/translations/ (string: project) /
string: component/string: language/file/ Upload new file with translations.
?????

project (string) -- ?????? URL ?????

component (string) -- ?????? URL ?????

language (string) -- Translation language code
????? ?????

string conflicts -- How to deal with conflicts (ignore, replace-translated or replace-approved)

file file -- Uploaded file

string email -- ????????????

string author -- ?????

string method -- ?????? ??????translate, approve, suggest, fuzzy, replace, source, add?????: ??????

string fuzzy -- Fuzzy (marked for edit) strings processing (*empty*, process, approve)

CURL ??:

```
curl -X POST \
  -F file=@strings.xml \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/translations/hello/android/cs/file/
```

GET /api/translations/ (string: project) /
string: component/string: language/repository/ Returns information about VCS repository status.

The response is same as for **GET /api/components/ (string:project) / (string:component) / repository/**.

?????

project (string) -- ?????? URL ?????

```

component (string) -- URL
language (string) -- Translation language code
POST /api/translations/ (string: project) /
string: component/string: language/repository/ VCS URL
See POST /api/projects/\(string:project\)/repository/ for documentation.
JSON

project (string) -- URL
component (string) -- URL
language (string) -- Translation language code
JSON
operation (string) -- push, pull, commit, reset, cleanup
JSON
result (boolean) --
GET /api/translations/ (string: project) /
string: component/string: language/statistics/ Returns detailed translation statistics.
2.7.
JSON

project (string) -- URL
component (string) -- URL
language (string) -- Translation language code
JSON
code (string) --
failing (int) -- number of failing checks
failing_percent (float) -- percentage of failing checks
fuzzy (int) --
fuzzy_percent (float) --
total_words (int) --
translated_words (int) --
last_author (string) -- name of last author
last_change (timestamp) -- date of last change
name (string) --
total (int) --
translated (int) --
translated_percent (float) --
url (string) -- URL to access the translation (engagement URL)
url_translate (string) -- URL to access the translation (real translation URL)

```

Units

A *unit* is a single piece of a translation which pairs a source string with a corresponding translated string and also contains some related metadata. The term is derived from the [Translate Toolkit](#) and XLIFF.

2.10.

GET /api/units/
Returns list of translation units.

JSON:

Unit object attributes are documented at [GET /api/units/\(int:id\)/](#).

GET /api/units/(int: id) /
4.3: The `target` and `source` are now arrays to properly handle plural strings.

Returns information about translation unit.

JSON

id (*int*) -- Unit ID

????? JSON ??????

translation (*string*) -- URL of a related translation object
source (*array*) -- ??
previous_source (*string*) -- previous source string used for fuzzy matching
target (*array*) -- target string
id_hash (*string*) -- unique identifier of the unit
content_hash (*string*) -- unique identifier of the source string
location (*string*) -- location of the unit in source code
context (*string*) -- translation unit context
note (*string*) -- translation unit note
flags (*string*) -- translation unit flags
state (*int*) -- unit state, 0 - not translated, 10 - needs editing, 20 - translated, 30 - approved, 100 - read only
fuzzy (*boolean*) -- ????? "fuzzy" ??????????????????
translated (*boolean*) -- ??????????????
approved (*boolean*) -- ??????????????
position (*int*) -- unit position in translation file
has_suggestion (*boolean*) -- ??????????????
has_comment (*boolean*) -- ??????????????
has_failing_check (*boolean*) -- ??????????????????????
num_words (*int*) -- number of source words
priority (*int*) -- translation priority; 100 is default
id (*int*) -- unit identifier
explanation (*string*) -- String explanation, available on source units, see [Additional info on source strings](#)
extra_flags (*string*) -- ??????????????????????: ??????????????????
web_url (*string*) -- URL where the unit can be edited
source_unit (*string*) -- Source unit link; see [GET /api/units/\(int:id\)/](#)

PATCH /api/units/(int: id) /
????? 4.3 ???.

????????????????????
??????

id (*int*) -- Unit ID
????? JSON ??????

state (*int*) -- unit state, 0 - not translated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see [??????](#))

target (*array*) -- target string

explanation (*string*) -- String explanation, available on source units, see [Additional info on source strings](#)

extra_flags (*string*) -- ??????????????????????: ??????????????????

PUT /api/units/(int: id) /
????? 4.3 ???.

????????????????????
??????

id (*int*) -- Unit ID
????? JSON ??????

state (*int*) -- unit state, 0 - not translated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see [??????](#))

target (*array*) -- target string

explanation (*string*) -- String explanation, available on source units, see [Additional info on source strings](#)

extra_flags (*string*) -- ??????????????????????: ??????????????????

DELETE /api/units/(int: id) /
????? 4.3 ???.

????????????
????

id(*int*) -- Unit ID

??

???? 2.10 ??.

GET /api/changes/

???? 4.1 ???: ?????????? 4.1 ??????????

Returns a list of translation changes.

??:

Change object attributes are documented at [GET /api/changes/\(int:id\)/](#).

??? ?????

user(*string*) -- Username of user to filters

action(*int*) -- Action to filter, can be used several times

timestamp_after(*timestamp*) -- ISO 8601 formatted timestamp to list changes after

timestamp_before(*timestamp*) -- ISO 8601 formatted timestamp to list changes before

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

Returns information about translation change.

????

id(*int*) -- Change ID

???? JSON ?????

unit(*string*) -- URL of a related unit object

translation(*string*) -- URL of a related translation object

component(*string*) -- URL of a related component object

user(*string*) -- URL of a related user object

author(*string*) -- URL of a related author object

timestamp(*timestamp*) -- event timestamp

action(*int*) -- numeric identification of action

action_name(*string*) -- text description of action

target(*string*) -- event changed text or detail

id(*int*) -- change identifier

????????

???? 2.14 ??.

GET /api/screenshots/

Returns a list of screenshot string information.

??:

Screenshot object attributes are documented at [GET /api/screenshots/\(int:id\)/](#).

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

Returns information about screenshot information.

????

id(*int*) -- Screenshot ID

???? JSON ?????

name(*string*) -- name of a screenshot

component(*string*) -- URL of a related component object

file_url(*string*) -- URL to download a file; see [GET /api/screenshots/\(int:id\)/file/](#)

units(*array*) -- link to associated source string information; see [GET /api/units/\(int:id\)/](#)

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

Download the screenshot image.


```

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

id (int) -- Screenshot ID
?????? JSON ???????

name (string) -- name of a screenshot

component (string) -- URL of a related component object

file_url (string) -- URL to download a file; see GET /api/screenshots/(int:id)/file/

units (array) -- link to associated source string information; see GET /api/units/(int:id)/

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

id (int) -- Screenshot ID

?????

?????? 4.4.1 ????

GET /api/addons/
????????????????????

??:

????? ?????????????? GET /api/addons/(int:id)/ ??????????????????

GET /api/addons/(int: id) /
????????????????????????????
??????

id (int) -- ???? ID
?????? JSON ???????

name (string) -- ???????

component (string) -- URL of a related component object

configuration (object) -- ???????????????

POST /api/components/(string: project) /
string: component/addons/ ??????????????????
??????

project_slug (string) -- ??????????????

component_slug (string) -- ???????????????
?????? JSON ???????

name (string) -- ???????

configuration (object) -- ???????????????

PATCH /api/addons/(int: id) /
????????????????????????????
??????

id (int) -- ???? ID
?????? JSON ???????

configuration (object) -- ???????????????

PUT /api/addons/(int: id) /
????????????????????????????
??????

id (int) -- ???? ID
?????? JSON ???????

configuration (object) -- ???????????????

DELETE /api/addons/(int: id) /
????????????????
??????

id (int) -- ???? ID

```

????????? ???

????? 4.0 ???.

GET /api/component-lists/
????????????????

??:

??????? ??????????????GET /api/component-lists/(str:slug)/????????????????

GET /api/component-lists/(str: slug) /
??????? ??????????????
??????

slug(string)-- ??????? ?????????
?????? JSON ???????

name(string)-- ??????? ???????

slug(string)-- ??????? ?????????

show_dashboard(boolean)-- ?????????????????

components(array)-- ??????????????????: GET /api/components/(string:project)/
(string:component)/

auto_assign(array)-- ?????????

PUT /api/component-lists/(str: slug) /
??????? ??????????????
??????

slug(string)-- ??????? ?????????
?????? JSON ???????

name(string)-- ??????? ???????

slug(string)-- ??????? ?????????

show_dashboard(boolean)-- ?????????????????

PATCH /api/component-lists/(str: slug) /
??????? ??????????????
??????

slug(string)-- ??????? ?????????
?????? JSON ???????

name(string)-- ??????? ???????

slug(string)-- ??????? ?????????

show_dashboard(boolean)-- ?????????????????

DELETE /api/component-lists/(str: slug) /
??????? ??????????????
??????

slug(string)-- ??????? ?????????

POST /api/component-lists/(str: slug) /components/
???????????????????? ??????????????
??????

slug(string)-- ??????? ?????????
????? ???????

string component_id-- ??????? ID

DELETE /api/component-lists/(str: slug) /components/
str: component_slug ??????? ?????????????????????
??????

slug(string)-- ??????? ?????????

component_slug(string)-- ?????????????

???

4.5: API

???

4.4.

GET /api/tasks/
?

GET /api/tasks/(str: uuid) /
?
?

uuid(string) -- UUID
JSON

completed(boolean) --

progress(int) --

result(object) --

log(string) --

?????

Notification hooks allow external applications to notify Weblate that the VCS repository has been updated.

You can use repository endpoints for projects, components and translations to update individual repositories; see `POST /api/projects/(string:project)/repository/` for documentation.

GET /hooks/update/(string: project) /
string: component/ 2.6: Please use `POST /api/components/(string:project)/(string:component)/repository/` instead which works properly with authentication for ACL limited projects.

Triggers update of a component (pulling from VCS and scanning for translation changes).

GET /hooks/update/(string: project) /
2.6: Please use `POST /api/projects/(string:project)/repository/` instead which works properly with authentication for ACL limited projects.

Triggers update of all components in a project (pulling from VCS and scanning for translation changes).

POST /hooks/github/

Special hook for handling GitHub notifications and automatically updating matching components.

: GitHub includes direct support for notifying Weblate: enable Weblate service hook in repository settings and set the URL to the URL of your Weblate installation.

:

Automatically receiving changes from GitHub

For instruction on setting up GitHub integration

<https://docs.github.com/en/github/extending-github/about-webhooks>

Generic information about GitHub Webhooks

`ENABLE_HOOKS`

For enabling hooks for whole Weblate

POST /hooks/gitlab/

Special hook for handling GitLab notifications and automatically updating matching components.

:

Automatically receiving changes from GitLab

For instruction on setting up GitLab integration

<https://docs.gitlab.com/ce/user/project/integrations/webhooks.html>

Generic information about GitLab Webhooks

`ENABLE_HOOKS`

For enabling hooks for whole Weblate

POST /hooks/bitbucket/

Special hook for handling Bitbucket notifications and automatically updating matching components.

??:

Automatically receiving changes from Bitbucket

For instruction on setting up Bitbucket integration

<https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/>

Generic information about Bitbucket Webhooks

[ENABLE_HOOKS](#)

For enabling hooks for whole Weblate

POST /hooks/pagure/

????? 3.3 ???.

Pagure ???

??:

Pagure ??????????????????

Pagure ??????????????

https://docs.pagure.org/pagure/usage/using_webhooks.html

Pagure WEB ??????????????????

[ENABLE_HOOKS](#)

For enabling hooks for whole Weblate

POST /hooks/azure/

????? 3.8 ???.

Special hook for handling Azure Repos notifications and automatically updating matching components.

??:

Automatically receiving changes from Azure Repos

For instruction on setting up Azure integration

<https://docs.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops>

Generic information about Azure Repos Web Hooks

[ENABLE_HOOKS](#)

For enabling hooks for whole Weblate

POST /hooks/gitea/

????? 3.9 ???.

Special hook for handling Gitea Webhook notifications and automatically updating matching components.

??:

Automatically receiving changes from Gitea Repos

For instruction on setting up Gitea integration

<https://docs.gitea.io/en-us/webhooks/>

Generic information about Gitea Webhooks

[ENABLE_HOOKS](#)

For enabling hooks for whole Weblate

POST /hooks/gitee/

????? 3.9 ???.

Special hook for handling Gitee Webhook notifications and automatically updating matching components.

??:

Automatically receiving changes from Gitee Repos

For instruction on setting up Gitee integration

<https://gitee.com/help/categories/40>

Generic information about Gitee Webhooks

[ENABLE_HOOKS](#)

For enabling hooks for whole Weblate

Exports

Weblate provides various exports to allow you to further process the data.

GET /exports/stats/(string: project) /

string: component /

??? ?????

format (string) -- Output format: either json or csv

????? 2.6 ????: Please use `GET /api/components/(string:project)/(string:component)/statistics/` and `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/` instead; it allows access to ACL controlled projects as well.

Retrieves statistics for given component in given format.

Example request:

```
GET /exports/stats/weblate/main/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

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

RSS

Changes in translations are exported in RSS feeds.

GET /exports/rss/(string: project) /
string: component/string: language/ Retrieves RSS feed with recent changes for a translation.

GET /exports/rss/(string: project) /
string: component/ Retrieves RSS feed with recent changes for a component.

GET /exports/rss/(string: project) /
Retrieves RSS feed with recent changes for a project.

GET /exports/rss/language/(string: language) /
Retrieves RSS feed with recent changes for a language.

GET /exports/rss/
Retrieves RSS feed with recent changes for Weblate instance.

Example:

RSS Wikipedia

Weblate

Weblate 2.7: Weblate 2.7 wlc API

Installation

Weblate Python wlc

```
pip3 install wlc
```

Docker

Weblate Docker

Docker Hub: <https://hub.docker.com/r/weblate/wlc>

Example:

```
docker pull weblate/wlc
```

Docker Weblate localhost API API URL API_KEY

Example:

```
docker run --rm weblate/wlc [WLC_ARGS]
```

Example:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

You might want to pass your `API_KEY` to the Docker container, the easiest approach is to add your current directory as `/home/weblate` volume:

```
docker run --volume $PWD:/home/weblate --rm weblate/wlc show
```

????

wlc `???` `~/config/weblate` `????????????????????` `???????` for other locations
`???.????????????????????`:

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

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

`??????????????` `????????????????????`

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

??:

`???????`

??

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

`????????????????????????????????`

??

Weblate `???????` Python `??????????????` *Weblate's REST API* `??????` Weblate `????????????????????????????`
`????????????????????????` `??????????????wlc` `????????????????????wlc` `??????????????`

??

`????????????????????????????????` Weblate `????????????????????????????????`

--format {csv,json,text,html}
`??????????????`

--url URL
API `?` URL `????????????????????????????`: `??????????URL` `????` /api/ `??????`: https://
hosted.weblate.org/api/`?`

--key KEY
`????` API `????` `????????????????????????????`: `??????????` Weblate `????????????????`

--config PATH
`????????????????`: `??????????`

--config-section SECTION
`??????????????` `????????????????`: `??????????`

?????

`????????????`

version
`????????????????`

list-languages
Weblate `????????????????`

list-projects
Weblate `????????????????`

list-components
Weblate `????????????????`

list-translations
Weblate `??????????????`

??????

1.6: .ini

1.6.

Windows

XDG XDG_CONFIG_HOME XDG_CONFIG_DIRS
Windows APPDATA

[weblate] --config-section:

key

Weblate API

url

API URL http://127.0.0.1:8000/api/

translation

-

INI:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

API [keys]

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

VCS .weblate wlc

?

:

```
$ wlc version
version: 0.1
```

:

```
$ 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/
```

wlc:

```
$ 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+
```

(?)

(XXXXXXXXXXXX)

```
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/
```

XX:

```
$ wlc commit
```

Weblate Python API

Python API XXXXXXXXXXXXXXXXXXXXXXXXXXXX *Weblate :* (wlc) XXXXXXXXXXXXXXXXXXXX

```
pip install wlc
```

wlc

WeblateException

exception `wlc.WeblateException`
XXXXXXXXXXXXXXXXXXXX

Weblate

class `wlc.Weblate` (*key="", url=None, config=None*)
XXXXXX

key (*str*) -- XXXX key

url (*str*) -- API XXXXX URLXXXXXXXXXXXXXXXXXXXX

config (`wlc.config.WeblateConfig`) -- XXXXXXXXXXXXXXXXXXXXXXXX

API XXXXXXX XXXXXAPI XXXXXXX API URL XXXXXXX

get (*path*)
XXXXXX

path (*str*) -- XXXXXXXX
XXXXXX

object

XXX API GET XXXXXXXXXXXXXXX

post (*path, **kwargs*)
XXXXXX

path (*str*) -- XXXXXXXX
XXXXXX

object

XXX API GET XXXXXXXXXXXXXXX

wlc.config

WeblateConfig

```
class wlc.config.WeblateConfig (section='wlc')
    """
    section (str) -- """
    XDG """
    load (path=None)
    """
    path (str) -- """
    """XDG  /etc/xdg/wlc  wlc ~/
    config/wlc """
```

wlc.main

```
wlc.main.main (settings=None, stdout=None, args=None)
    """
    settings (list) -- """
    stdout (object) -- """ stdout """ sys.stdout """
    args (list) -- """ sys.args """
    """ """
    @wlc.main.register_command (command)
    main() """main() """
```

Command

```
class wlc.main.Command (args, config, stdout=None)
    """
    """
```

"""

Weblate """

Docker """

With dockerized Weblate deployment you can get your personal Weblate instance up and running in seconds. All of Weblate's dependencies are already included. PostgreSQL is set up as the default database.

Hardware requirements

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

2 GB of RAM

2 CPU cores

1 GB of storage space

The more memory the better - it is used for caching on all levels (filesystem, database and Weblate).

Many concurrent users increases the amount of needed CPU cores. For hundreds of translation components at least 4 GB of RAM is recommended.

The typical database storage usage is around 300 MB per 1 million hosted words. Storage space needed for cloned repositories varies, but Weblate tries to keep their size minimal by doing shallow clones.

??: Actual requirements for your installation of Weblate vary heavily based on the size of the translations managed in it.

??

The following examples assume you have a working Docker environment, with `docker-compose` installed. Please check the Docker documentation for instructions.

1. Clone the weblate-docker repo:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Create a `docker-compose.override.yml` file with your settings. See [Docker environment variables](#) for full list of environment variables.

```
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
```

??: If `WEBLATE_ADMIN_PASSWORD` is not set, the admin user is created with a random password shown on first startup.

The provided example makes Weblate listen on port 80, edit the port mapping in the `docker-compose.override.yml` file to change it.

3. Start Weblate containers:

```
docker-compose up
```

Enjoy your Weblate deployment, it's accessible on port 80 of the `weblate` container.

???? 2.15-2 ??: The setup has changed recently, priorly there was separate web server container, since 2.15-2 the web server is embedded in the Weblate container.

???? 3.7.1-6 ??: In July 2019 (starting with the 3.7.1-6 tag), the containers are not running as a root user. This has changed the exposed port from 80 to 8080.

??:

Invoking management commands

Docker container with HTTPS support

Please see [??](#) for generic deployment instructions, this section only mentions differences compared to it.

Using own SSL certificates

3.8-3

In case you have own SSL certificate you want to use, simply place the files into the Weblate data volume (see *Docker container volumes*):

ssl/fullchain.pem containing the certificate including any needed CA certificates

ssl/privkey.pem containing the private key

Both of these files must be owned by the same user as the one starting the docker container and have file mask set to 600 (readable and writable only by the owning user).

Additionally, Weblate container will now accept SSL connections on port 4443, you will want to include the port forwarding for HTTPS in docker compose override:

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

If you already host other sites on the same server, it is likely ports 80 and 443 are used by a reverse proxy, such as NGINX. To pass the HTTPS connection from NGINX to the docker container, you can use the following configuration:

```
server {
    listen 443;
    listen [::]:443;

    server_name <SITE_URL>;
    ssl_certificate /etc/letsencrypt/live/<SITE>/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/<SITE>/privkey.pem;

    location / {
        proxy_set_header HOST $host;
        proxy_set_header X-Forwarded-Proto https;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Host $server_name;
        proxy_pass https://127.0.0.1:<EXPOSED_DOCKER_PORT>;
    }
}
```

Replace <SITE_URL>, <SITE> and <EXPOSED_DOCKER_PORT> with actual values from your environment.

Automatic SSL certificates using Let's Encrypt

In case you want to use Let's Encrypt automatically generated SSL certificates on public installation, you need to add a reverse HTTPS proxy an additional Docker container, [https-portal](#) will be used for that. This is made use of in the docker-compose-https.yml file. Then create a docker-compose-https.override.yml file with your settings:

```
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'
```

Whenever invoking **docker-compose** you need to pass both files to it, and then do:

```
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
```

Upgrading the Docker container

Usually it is good idea to only update the Weblate container and keep the PostgreSQL container at the version you have, as upgrading PostgreSQL is quite painful and in most cases does not bring many benefits.

You can do this by sticking with the existing docker-compose and just pull the latest images and then restart:

```
docker-compose stop
docker-compose pull
docker-compose up
```

The Weblate database should be automatically migrated on first startup, and there should be no need for additional manual actions.

!!! Upgrades across 3.0 are not supported by Weblate. If you are on 2.x series and want to upgrade to 3.x, first upgrade to the latest 3.0.1-x (at time of writing this it is the 3.0.1-7) image, which will do the migration and then continue upgrading to newer versions.

You might also want to update the `docker-compose` repository, though it's not needed in most case. Please beware of PostgreSQL version changes in this case as it's not straightforward to upgrade the database, see [GitHub issue](#) for more info.

?????? ??

After container setup, you can sign in as *admin* user with password provided in `WEBLATE_ADMIN_PASSWORD`, or a random password generated on first start if that was not set.

To reset *admin* password, restart the container with `WEBLATE_ADMIN_PASSWORD` set to new password.

!!:

`WEBLATE_ADMIN_PASSWORD`?`WEBLATE_ADMIN_NAME`?`WEBLATE_ADMIN_EMAIL`

Number of processes and memory consumption

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

In case you have a lot of CPU cores and hit out of memory issues, try reducing number of workers:

```
environment:
  WEBLATE_WORKERS: 2
```

You can also fine-tune individual worker categories:

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

!!:

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

Scaling horizontally

4.6

⚠: This feature is a technology preview.

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

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

You can find example setup in the `docker-compose` repo as [docker-compose-split.yml](#).

Docker environment variables

Many of Weblate's [vars](#) can be set in the Docker container using environment variables:

Generic settings

WEBLATE_DEBUG

Configures Django debug mode using `DEBUG`.

⚠:

```
environment:
  WEBLATE_DEBUG: 1
```

⚠:

[vars](#) [vars](#)

WEBLATE_LOGLEVEL

Configures the logging verbosity.

WEBLATE_SITE_TITLE

Changes the site-title shown in the header of all pages.

WEBLATE_SITE_DOMAIN

[vars](#)

⚠:

[vars](#)`SITE_DOMAIN`

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Configures the site-admin's name and e-mail. It is used for both `ADMINS` setting and creating `admin` user (see [WEBLATE_ADMIN_PASSWORD](#) for more info on that).

⚠:

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

⚠:

[vars](#) [vars](#)`ADMINS`

WEBLATE_ADMIN_PASSWORD

Sets the password for the `admin` user.

If not set and `admin` user does not exist, it is created with a random password shown on first container startup.

If not set and `admin` user exists, no action is performed.

If set the `admin` user is adjusted on every container startup to match `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME` and `WEBLATE_ADMIN_EMAIL`.

??: It might be a security risk to store password in the configuration file. Consider using this variable only for initial setup (or let Weblate generate random password on initial startup) or for password recovery.

??:

?????? ?? `WEBLATE_ADMIN_PASSWORD` `WEBLATE_ADMIN_NAME` `WEBLATE_ADMIN_EMAIL`

WEBLATE_SERVER_EMAIL

WEBLATE_DEFAULT_FROM_EMAIL

Configures the address for outgoing e-mails.

??:

????????

WEBLATE_CONTACT_FORM

Configures contact form behavior, see `CONTACT_FORM`.

WEBLATE_ALLOWED_HOSTS

Configures allowed HTTP hostnames using `ALLOWED_HOSTS`.

Defaults to * which allows all hostnames.

?:

environment:

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

??:

ALLOWED_HOSTS `????????????????????????????????`

WEBLATE_REGISTRATION_OPEN

Configures whether registrations are open by toggling `REGISTRATION_OPEN`.

?:

environment:

WEBLATE_REGISTRATION_OPEN: `0`

WEBLATE_REGISTRATION_ALLOW_BACKENDS

Configure which authentication methods can be used to create new account via `REGISTRATION_ALLOW_BACKENDS`.

?:

environment:

WEBLATE_REGISTRATION_OPEN: `0`

WEBLATE_REGISTRATION_ALLOW_BACKENDS: `azuread-oauth2,azuread-tenant-oauth2`

WEBLATE_TIME_ZONE

Configures the used time zone in Weblate, see `TIME_ZONE`.

??: To change the time zone of the Docker container itself, use the `TZ` environment variable.

?:

environment:

WEBLATE_TIME_ZONE: `Europe/Prague`

WEBLATE_ENABLE_HTTPS

Makes Weblate assume it is operated behind a reverse HTTPS proxy, it makes Weblate use HTTPS in e-mail and API links or set secure flags on cookies.

???: Please see `ENABLE_HTTPS` documentation for possible caveats.

??: This does not make the Weblate container accept HTTPS connections, you need to configure that as well, see *Docker container with HTTPS support* for examples.

?:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

??:

`ENABLE_HTTPS` `WEBLATE_SECURE_PROXY_SSL_HEADER`

WEBLATE_IP_PROXY_HEADER

Lets Weblate fetch the IP address from any given HTTP header. Use this when using a reverse proxy in front of the Weblate container.

Enables `IP_BEHIND_REVERSE_PROXY` and sets `IP_PROXY_HEADER`.

?: The format must conform to Django's expectations. Django [transforms](#) raw HTTP header names as follows:

converts all characters to uppercase

replaces any hyphens with underscores

prepends HTTP_ prefix

So X-Forwarded-For would be mapped to HTTP_X_FORWARDED_FOR.

?:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

A tuple representing a HTTP header/value combination that signifies a request is secure. This is needed when Weblate is running behind a reverse proxy doing SSL termination which does not pass standard HTTPS headers.

?:

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

??:

`SECURE_PROXY_SSL_HEADER`

WEBLATE_REQUIRE_LOGIN

Enables `REQUIRE_LOGIN` to enforce authentication on whole Weblate.

?:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

Adds URL exceptions for authentication required for the whole Weblate installation using `LOGIN_REQUIRED_URLS_EXCEPTIONS`.

You can either replace whole settings, or modify default value using ADD and REMOVE variables.

WEBLATE_GOOGLE_ANALYTICS_ID

Configures ID for Google Analytics by changing `GOOGLE_ANALYTICS_ID`.

WEBLATE_GITHUB_USERNAME

Configures GitHub username for GitHub pull-requests by changing `GITHUB_USERNAME`.

??:

GitHub

WEBLATE_GITHUB_TOKEN

4.3

Configures GitHub personal access token for GitHub pull-requests via API by changing `GITHUB_TOKEN`.

?:

GitHub

WEBLATE_GITLAB_USERNAME

Configures GitLab username for GitLab merge-requests by changing `GITLAB_USERNAME`

??:

GitLab

WEBLATE_GITLAB_TOKEN

Configures GitLab personal access token for GitLab merge-requests via API by changing *GITLAB_TOKEN*

??:

GitLab

WEBLATE_PAGURE_USERNAME

PAGURE_USERNAME ??????Pagure merge-requests ?? Pagure ????????????

??:

Pagure

WEBLATE_PAGURE_TOKEN

PAGURE_TOKEN ??????API ??? Pagure ????? ??????????Pagure ?????????? ????????????

??:

Pagure

WEBLATE_SIMPLIFY_LANGUAGES

Configures the language simplification policy, see *SIMPLIFY_LANGUAGES*.

WEBLATE_DEFAULT_ACCESS_CONTROL

Configures the default *???????* for new projects, see *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Configures the default value for *???????* for new components, see *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Configures the default value for *???????????????* for new components, see *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

Configures *DEFAULT_COMMITER_EMAIL*.

WEBLATE_DEFAULT_COMMITER_NAME

Configures *DEFAULT_COMMITER_NAME*.

WEBLATE_DEFAULT_SHARED_TM

DEFAULT_SHARED_TM ?????

WEBLATE_AKISMET_API_KEY

Configures the Akismet API key, see *AKISMET_API_KEY*.

WEBLATE_GPG_IDENTITY

Configures GPG signing of commits, see *WEBLATE_GPG_IDENTITY*.

??:

Signing Git commits with GnuPG

WEBLATE_URL_PREFIX

Configures URL prefix where Weblate is running, see *URL_PREFIX*.

WEBLATE_SILENCED_SYSTEM_CHECKS

Configures checks which you do not want to be displayed, see *SILENCED_SYSTEM_CHECKS*.

WEBLATE_CSP_SCRIPT_SRC

WEBLATE_CSP_IMG_SRC

WEBLATE_CSP_CONNECT_SRC

WEBLATE_CSP_STYLE_SRC

WEBLATE_CSP_FONT_SRC

Allows to customize Content-Security-Policy HTTP header.

??:

????? ??????? ??????CSP_SCRIPT_SRC?CSP_IMG_SRC?CSP_CONNECT_SRC?CSP_STYLE_SRC?CSP_FONT_SRC

WEBLATE_LICENSE_FILTER

LICENSE_FILTER ?????

WEBLATE_LICENSE_REQUIRED

LICENSE_REQUIRED ???

WEBLATE_WEBSITE_REQUIRED

WEBSITE_REQUIRED ???

□□□□□□□□□

```
environment:
  WEBLATE_MT_MYMEMORY_ENABLED: 1
```

WEBLATE_MT_GLOSBE_ENABLED
Glosbe [????????](#)

```
environment:
  WEBLATE_MT_GLOSBE_ENABLED: 1
```

WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED
Microsoft Terminology Service [????????](#)

```
environment:
  WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED: 1
```

WEBLATE_MT_SAP_BASE_URL
WEBLATE_MT_SAP_SANDBOX_APIKEY
WEBLATE_MT_SAP_USERNAME
WEBLATE_MT_SAP_PASSWORD
WEBLATE_MT_SAP_USE_MT
Configures *SAP Translation Hub* machine translation.

```
environment:
  WEBLATE_MT_SAP_BASE_URL: "https://example.hana.ondemand.com/
  ↪translationhub/api/v1/"
  WEBLATE_MT_SAP_USERNAME: "user"
  WEBLATE_MT_SAP_PASSWORD: "password"
  WEBLATE_MT_SAP_USE_MT: 1
```

Authentication settings

LDAP

WEBLATE_AUTH_LDAP_SERVER_URI
WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE
WEBLATE_AUTH_LDAP_USER_ATTR_MAP
WEBLATE_AUTH_LDAP_BIND_DN
WEBLATE_AUTH_LDAP_BIND_PASSWORD
WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS
WEBLATE_AUTH_LDAP_USER_SEARCH
WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER
WEBLATE_AUTH_LDAP_USER_SEARCH_UNION
WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER
LDAP authentication configuration.

Example for direct bind:

```
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
```

Example for search and bind:

```
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
```

Example for union search and bind:

```
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
```

Example with search and bind against Active Directory:

```
environment:
  WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS: 0
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH: CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER: (sAMAccountName=%(user)s)
```

??:

LDAP ??

GitHub

```
WEBLATE_SOCIAL_AUTH_GITHUB_KEY
WEBLATE_SOCIAL_AUTH_GITHUB_SECRET
GitHub ?? ?????
```

Bitbucket

```
WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY
WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET
Bitbucket ?? ?????
```

Facebook

```
WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY
WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET
Facebook OAuth 2 ?????
```

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 ?????
```

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY

WEBLATE_SOCIAL_AUTH_GITLAB_SECRET

WEBLATE_SOCIAL_AUTH_GITLAB_API_URL

GitLab OAuth 2 [?????](#)

Azure Active Directory

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET

Enables Azure Active Directory authentication, see *Microsoft Azure Active Directory*.

Azure Active Directory with Tenant support

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID

Enables Azure Active Directory authentication with Tenant support, see *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

Enables Keycloak authentication, see [documentation](#).

Linux vendors

You can enable authentication using Linux vendors authentication services by setting following variables to any value.

WEBLATE_SOCIAL_AUTH_FEDORA

WEBLATE_SOCIAL_AUTH_OPENSUSE

WEBLATE_SOCIAL_AUTH_UBUNTU

Slack

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Enables Slack authentication, see *Slack*.

SAML

Self-signed SAML keys are automatically generated on first container startup. In case you want to use own keys, place the certificate and private key in `/app/data/ssl/saml.crt` and `/app/data/ssl/saml.key`.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT

SAML Identity Provider settings, see [SAML](#).

Other authentication settings

WEBLATE_NO_EMAIL_AUTH

Disables e-mail authentication when set to any value.

PostgreSQL database setup

The database is created by `docker-compose.yml`, so these settings affect both Weblate and PostgreSQL containers.

Env:

Weblate

POSTGRES_PASSWORD

PostgreSQL password.

POSTGRES_USER

PostgreSQL username.

POSTGRES_DATABASE

PostgreSQL database name.

POSTGRES_HOST

PostgreSQL server hostname or IP address. Defaults to `database`.

POSTGRES_PORT

PostgreSQL server port. Defaults to `none` (uses the default value).

POSTGRES_SSL_MODE

Configure how PostgreSQL handles SSL in connection to the server, for possible choices see [SSL Mode Descriptions](#)

POSTGRES_ALTER_ROLE

Configures name of role to alter during migrations, see [PostgreSQL](#) *Weblate*.

Database backup settings

Env:

Dumped data for backups

WEBLATE_DATABASE_BACKUP

Configures the daily database dump using `DATABASE_BACKUP`. Defaults to `plain`.

Caching server setup

Using Redis is strongly recommended by Weblate and you have to provide a Redis instance when running Weblate in Docker.

Env:

REDIS_HOST

The Redis server hostname or IP address. Defaults to `cache`.

REDIS_PORT

The Redis server port. Defaults to `6379`.

REDIS_DB

The Redis database number, defaults to `1`.

REDIS_PASSWORD

The Redis server password, not used by default.

REDIS_TLS

Enables using SSL for Redis connection.

REDIS_VERIFY_SSL

Can be used to disable SSL certificate verification for Redis connection.

Email server setup

To make outgoing e-mail work, you need to provide a mail server.

Example TLS configuration:

```
environment:
  WEBLATE_EMAIL_HOST: smtp.example.com
  WEBLATE_EMAIL_HOST_USER: user
  WEBLATE_EMAIL_HOST_PASSWORD: pass
```

Example SSL configuration:

```
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
```

??:

????????

WEBLATE_EMAIL_HOST

Mail server hostname or IP address.

??:

WEBLATE_EMAIL_PORT?WEBLATE_EMAIL_USE_SSL?WEBLATE_EMAIL_USE_TLS?EMAIL_HOST

WEBLATE_EMAIL_PORT

Mail server port, defaults to 25.

??:

EMAIL_PORT

WEBLATE_EMAIL_HOST_USER

????????

??:

EMAIL_HOST_USER

WEBLATE_EMAIL_HOST_PASSWORD

????????

??:

EMAIL_HOST_PASSWORD

WEBLATE_EMAIL_USE_SSL

Whether to use an implicit TLS (secure) connection when talking to the SMTP server. In most e-mail documentation, this type of TLS connection is referred to as SSL. It is generally used on port 465. If you are experiencing problems, see the explicit TLS setting *WEBLATE_EMAIL_USE_TLS*.

??:

WEBLATE_EMAIL_PORT?WEBLATE_EMAIL_USE_TLS?EMAIL_USE_SSL

WEBLATE_EMAIL_USE_TLS

Whether to use a TLS (secure) connection when talking to the SMTP server. This is used for explicit TLS connections, generally on port 587 or 25. If you are experiencing connections that hang, see the implicit TLS setting *WEBLATE_EMAIL_USE_SSL*.

??:

WEBLATE_EMAIL_PORT?WEBLATE_EMAIL_USE_SSL?EMAIL_USE_TLS

WEBLATE_EMAIL_BACKEND

Configures Django back-end to use for sending e-mails.

??:

`?????????EMAIL_BACKEND`

?????

WEBLATE_GET_HELP_URL

`GET_HELP_URL` **????**

WEBLATE_STATUS_URL

`STATUS_URL` **????**

WEBLATE_LEGAL_URL

`LEGAL_URL` **????**

Error reporting

It is recommended to collect errors from the installation systematically, see [??? ????????](#).

To enable support for Rollbar, set the following:

ROLLBAR_KEY

Your Rollbar post server access token.

ROLLBAR_ENVIRONMENT

Your Rollbar environment, defaults to `production`.

To enable support for Sentry, set following:

SENTRY_DSN

Your Sentry DSN.

SENTRY_ENVIRONMENT

Your Sentry Environment (optional).

CDN **????**

WEBLATE_LOCALIZE_CDN_URL

WEBLATE_LOCALIZE_CDN_PATH

???? 4.2.1 **???**.

Configuration for *JavaScript* **????** *CDN*.

The `WEBLATE_LOCALIZE_CDN_PATH` is path within the container. It should be stored on the persistent volume and not in the transient storage.

One of possibilities is storing that inside the Weblate data dir:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

??: You are responsible for setting up serving of the files generated by Weblate, it only does stores the files in configured location.

??:

`weblate-cdn`**?**`LOCALIZE_CDN_URL`**?**`LOCALIZE_CDN_PATH`

Changing enabled apps, checks, addons or autofixes

3.8-5

The built-in configuration of enabled checks, addons or autofixes can be adjusted by the following variables:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

:

```
environment:
  WEBLATE_REMOVE_AUTOFIX: weblate.trans.autofixes.whitespace.
  ↳ SameBookendingWhitespace
  WEBLATE_ADD_ADDONS: customize.addons.MyAddon,customize.addons.OtherAddon
```

:

`CHECK_LIST``AUTOFIX_LIST``WEBLATE_ADDONS``INSTALLED_APPS`

WEBLATE_WORKERS

4.6.1

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

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

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

These variables allow you to adjust Celery worker options. It can be useful to adjust concurrency (`--concurrency 16`) or use different pool implementation (`--pool=gevent`).

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

:

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

:

Celery worker options

UWSGI_WORKERS

Configure how many uWSGI workers should be executed.

It defaults to `WEBLATE_WORKERS`.

:

```
environment:
  UWSGI_WORKERS: 32
```

WEBLATE_SERVICE

Defines which services should be executed inside the container. Use this for *Scaling horizontally*.

Following services are defined:

`celery-beat`

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

`celery-backup`

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

`celery-celery`

Generic Celery worker.

`celery-memory`

Translation memory Celery worker.

`celery-notify`

Notifications Celery worker.

`celery-translate`

Automatic translation Celery worker.

`web`

Web `?????`

Docker container volumes

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

The data volume is used to store Weblate persistent data such as cloned repositories or to customize Weblate installation.

The placement of the Docker volume on host system depends on your Docker configuration, but usually it is stored in `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/`. In the container it is mounted as `/app/data`.

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

??:

[Docker volumes documentation](#)

Further configuration customization

You can further customize Weblate installation in the data volume, see *Docker container volumes*.

Custom configuration files

You can additionally override the configuration in `/app/data/settings-override.py` (see *Docker container volumes*). This is executed at the end of built-in settings, after all environment settings are loaded, and you can adjust or override them.

Replacing logo and other static files

`?????` 3.8-5 `???`.

The static files coming with Weblate can be overridden by placing into `/app/data/python/customize/static` (see *Docker container volumes*). For example creating `/app/data/python/customize/static/favicon.ico` will replace the favicon.

???: The files are copied to the corresponding location upon container startup, so a restart of Weblate is needed after changing the content of the volume.

Alternatively you can also include own module (see *Customizing Weblate*) and add it as separate volume to the Docker container, for example:

```

weblate:
  volumes:
    - weblate-data:/app/data
    - ../weblate_customization/weblate_customization:/app/data/python/
    ↪weblate_customization
  environment:
    WEBLATE_ADD_APPS: weblate_customization

```

Adding own Python modules

3.8-5

You can place own Python modules in `/app/data/python/` (see *Docker container volumes*) and they can be then loaded by Weblate, most likely by using *Custom configuration files*.

??:

Customizing Weblate

Select your machine - local or cloud providers

With Docker Machine you can create your Weblate deployment either on your local machine, or on any large number of cloud-based deployments on e.g. Amazon AWS, Greenhost, and many other providers.

Installing on Debian and Ubuntu

Hardware requirements

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

2 GB of RAM

2 CPU cores

1 GB of storage space

The more memory the better - it is used for caching on all levels (filesystem, database and Weblate).

Many concurrent users increases the amount of needed CPU cores. For hundreds of translation components at least 4 GB of RAM is recommended.

The typical database storage usage is around 300 MB per 1 million hosted words. Storage space needed for cloned repositories varies, but Weblate tries to keep their size minimal by doing shallow clones.

?: Actual requirements for your installation of Weblate vary heavily based on the size of the translations managed in it.

??

System requirements

Install the dependencies needed to build the Python modules (see [????????](#)):

```

apt install \
  libxml2-dev libxslt-dev libfreetype6-dev libjpeg-dev libz-dev libyaml-
  ↪dev \
  libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev libacl1-dev libssl-
  ↪dev \
  build-essential python3-gdbm python3-dev python3-pip python3-virtualenv
  ↪virtualenv git

```

Install wanted optional dependencies depending on features you intend to use (see [????????](#)):

```

apt install tesseract-ocr libtesseract-dev liblptonica-dev

```

Optionally install software for running production server, see [Docker](#), [Weblate Docker](#), [Celery Docker](#). Depending on size of your installation you might want to run these components on dedicated servers.

The local installation instructions:

```
# Web server option 1: NGINX and uWSGI
apt install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
apt install apache2 libapache2-mod-wsgi

# Caching backend: Redis
apt install redis-server

# Database server: PostgreSQL
apt install postgresql postgresql-contrib

# SMTP server
apt install exim4
```

Python modules

!!!: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv [User Guide](#).

1.Create the virtualenv for Weblate:

```
virtualenv --python=python3 ~/weblate-env
```

2.Activate the virtualenv for Weblate:

```
. ~/weblate-env/bin/activate
```

3.Install Weblate including all dependencies:

```
pip install Weblate
```

4.Install database driver:

```
pip install psycopg2-binary
```

5.Install wanted optional dependencies depending on features you intend to use (some might require additional system libraries, check [Docker](#)):

```
pip install ruamel.yaml aidon boto3 zeep chardet tesseract
```

Configuring Weblate

!!!: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **weblate** command as `~/weblate-env/bin/weblate`.

1.Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.

2.!!! settings.py [Docker](#) [Docker](#)

3.Create the database and its structure for Weblate (the example settings use PostgreSQL, check [Weblate Docker](#) for production ready setup):

```
weblate migrate
```

4.Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see [gettext](#) and [django](#)):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see [django](#) [compressor](#)):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See [Celery](#) [django](#) [worker](#) for more info:

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

8. Start the development server (see [django](#) for production setup):

```
weblate runserver
```

After installation

Congratulations, your Weblate server is now running and you can start using it.

You can now access Weblate on <http://localhost:8000/>.

Login with admin credentials obtained during installation or register with new users.

You can now run Weblate commands using **weblate** command when Weblate virtualenv is active, see [Management commands](#).

You can stop the test server with Ctrl+C.

Review potential issues with your installation either on [/manage/performance/](#) URL or using **weblate check --deploy**, see [django](#).

Adding translation

1. Open the admin interface (<http://localhost:8000/create/project/>) and create the project you want to translate. See [Project configuration](#) for more details.

All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See [Component configuration](#) for more details.

The important fields here are: Component name, VCS repository address and mask for finding translatable files. Weblate supports a wide range of formats including gettext PO files, Android resource strings, iOS string properties, Java properties or Qt Linguist files, see [django](#) for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Installing on SUSE and openSUSE

Hardware requirements

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

2 GB of RAM

2 CPU cores

1 GB of storage space

The more memory the better - it is used for caching on all levels (filesystem, database and Weblate).

Many concurrent users increases the amount of needed CPU cores. For hundreds of translation components at least 4 GB of RAM is recommended.

The typical database storage usage is around 300 MB per 1 million hosted words. Storage space needed for cloned repositories varies, but Weblate tries to keep their size minimal by doing shallow clones.

!!!: Actual requirements for your installation of Weblate vary heavily based on the size of the translations managed in it.

??

System requirements

Install the dependencies needed to build the Python modules (see [??????????](#)):

```
zypper install \
    libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel
↪libyaml-devel \
    cairo-devel typelib-1_0-Pango-1_0 gobject-introspection-devel libacl-
↪devel \
    python3-pip python3-virtualenv python3-devel git
```

Install wanted optional dependencies depending on features you intend to use (see [????????????](#)):

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

Optionally install software for running production server, see [??????](#), [Weblate ????????????](#), [Celery ?????????????? ????](#). Depending on size of your installation you might want to run these components on dedicated servers.

The local installation instructions:

```
# Web server option 1: NGINX and uWSGI
zypper install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
zypper install apache2 apache2-mod_wsgi

# Caching backend: Redis
zypper install redis-server

# Database server: PostgreSQL
zypper install postgresql postgresql-contrib

# SMTP server
zypper install postfix
```

Python modules

!!!: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv [User Guide](#).

1.Create the virtualenv for Weblate:

```
virtualenv --python=python3 ~/weblate-env
```

2.Activate the virtualenv for Weblate:

```
. ~/weblate-env/bin/activate
```

3.Install Weblate including all dependencies:

```
pip install Weblate
```

4.Install database driver:

```
pip install psycopg2-binary
```

5.Install wanted optional dependencies depending on features you intend to use (some might require additional system libraries, check [????????????](#)):

```
pip install ruamel.yaml aeidon boto3 zeep chardet tesseract
```

Configuring Weblate

❗: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **weblate** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. `settings.py` `#####`
`#####`
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check [Weblate #####](#) for production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see [???????](#) and [????????????](#)):

```
weblate collectstatic
```

- 6.Compress JavaScript and CSS files (optional, see [?????? ?????](#)):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See [Celery `worker` for more info:](#)

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

8. Start the development server (see [??????](#) for production setup):

```
weblate runserver
```

After installation

Congratulations, your Weblate server is now running and you can start using it.

You can now access Weblate on <http://localhost:8000/>.

Login with admin credentials obtained during installation or register with new users.

You can now run Weblate commands using **weblate** command when Weblate virtualenv is active, see [Management commands](#).

You can stop the test server with Ctrl+C.

Review potential issues with your installation either on `/manage/performance/` URL or using **weblate check --deploy**, see [????????](#).

Adding translation

1. Open the admin interface (<http://localhost:8000/create/project/>) and create the project you want to translate. See [Project configuration](#) for more details.

All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See [Component configuration](#) for more details.

The important fields here are: Component name, VCS repository address and mask for finding translatable files. Weblate supports a wide range of formats including gettext PO files, Android resource strings, iOS string properties, Java properties or Qt Linguist files, see [Supported file formats](#) for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Installing on RedHat, Fedora and CentOS

Hardware requirements

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

2 GB of RAM

2 CPU cores

1 GB of storage space

The more memory the better - it is used for caching on all levels (filesystem, database and Weblate).

Many concurrent users increases the amount of needed CPU cores. For hundreds of translation components at least 4 GB of RAM is recommended.

The typical database storage usage is around 300 MB per 1 million hosted words. Storage space needed for cloned repositories varies, but Weblate tries to keep their size minimal by doing shallow clones.

??: Actual requirements for your installation of Weblate vary heavily based on the size of the translations managed in it.

??

System requirements

Install the dependencies needed to build the Python modules (see [Dependencies](#)):

```
dnf install \
    libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
    ↳ libyaml-devel \
    cairo-devel pango-devel gobject-introspection-devel libacl-devel \
    python3-pip python3-virtualenv python3-devel git
```

Install wanted optional dependencies depending on features you intend to use (see [Optional dependencies](#)):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-devel
```

Optionally install software for running production server, see [Deployment](#), [Weblate Deployment](#), [Celery Deployment](#). Depending on size of your installation you might want to run these components on dedicated servers.

The local installation instructions:

```
# 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
```

([Dependencies](#))

[illegible]

```
# Caching backend: Redis
dnf install redis

# Database server: PostgreSQL
dnf install postgresql postgresql-contrib

# SMTP server
dnf install postfix
```

Python modules

!!!: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv [User Guide](#).

1. Create the virtualenv for Weblate:

```
virtualenv --python=python3 ~/weblate-env
```

2. Activate the virtualenv for Weblate:

```
. ~/weblate-env/bin/activate
```

3.Install Weblate including all dependencies:

```
pip install Weblate
```

4.Install database driver:

```
pip install psycopg2-binary
```

5. Install wanted optional dependencies depending on features you intend to use (some might require additional system libraries, check [here](#)):

```
pip install ruamel.yaml aeidon boto3 zeep chardet tesseract
```

Configuring Weblate

!!!: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **weblate** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.

[illegible]

3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check [Weblate](#) `??????????` for production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see [???????](#) and [????????????](#)):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see [?????? ???? ????](#)):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See [Celery](#) [\[1\]](#) for more info:

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

8. Start the development server (see [\[2\]](#) for production setup):

```
weblate runserver
```

After installation

Congratulations, your Weblate server is now running and you can start using it.

You can now access Weblate on `http://localhost:8000/`.

Login with admin credentials obtained during installation or register with new users.

You can now run Weblate commands using **weblate** command when Weblate virtualenv is active, see [Management commands](#).

You can stop the test server with Ctrl+C.

Review potential issues with your installation either on `/manage/performance/` URL or using **weblate check --deploy**, see [\[3\]](#).

Adding translation

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See [Project configuration](#) for more details.

All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See [Component configuration](#) for more details.

The important fields here are: Component name, VCS repository address and mask for finding translatable files. Weblate supports a wide range of formats including gettext PO files, Android resource strings, iOS string properties, Java properties or Qt Linguist files, see [\[4\]](#) for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Installing on macOS

Hardware requirements

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

2 GB of RAM

2 CPU cores

1 GB of storage space

The more memory the better - it is used for caching on all levels (filesystem, database and Weblate).

Many concurrent users increases the amount of needed CPU cores. For hundreds of translation components at least 4 GB of RAM is recommended.

The typical database storage usage is around 300 MB per 1 million hosted words. Storage space needed for cloned repositories varies, but Weblate tries to keep their size minimal by doing shallow clones.

[1]: Actual requirements for your installation of Weblate vary heavily based on the size of the translations managed in it.



System requirements

Install the dependencies needed to build the Python modules (see [Dependencies](#)):

```
brew install python pango cairo gobject-introspection libffi glib libyaml  
pip3 install virtualenv
```

Make sure pip will be able to find the libffi version provided by homebrew — this will be needed during the installation build step.

```
export PKG_CONFIG_PATH="/usr/local/opt/libffi/lib/pkgconfig"
```

Install wanted optional dependencies depending on features you intend to use (see [Optional dependencies](#)):

```
brew install tesseract
```

Optionally install software for running production server, see [Deployment](#), [Weblate \[Deployment\]\(#\)](#), [Celery \[Deployment\]\(#\)](#). Depending on size of your installation you might want to run these components on dedicated servers.

The local installation instructions:

```
# Web server option 1: NGINX and uWSGI  
brew install nginx uwsgi  
  
# Web server option 2: Apache with ``mod_wsgi``  
brew install httpd  
  
# Caching backend: Redis  
brew install redis  
  
# Database server: PostgreSQL  
brew install postgresql
```

Python modules

Tip: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv [User Guide](#).

1. Create the virtualenv for Weblate:

```
virtualenv --python=python3 ~/weblate-env
```

2. Activate the virtualenv for Weblate:

```
. ~/weblate-env/bin/activate
```

3. Install Weblate including all dependencies:

```
pip install Weblate
```

4. Install database driver:

```
pip install psycopg2-binary
```

5. Install wanted optional dependencies depending on features you intend to use (some might require additional system libraries, check [Optional dependencies](#)):

```
pip install ruamel.yaml aeidon boto3 zeep chardet tesseract
```

Configuring Weblate

Tip: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **weblate** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. `settings.py` `#####`

3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check [Weblate #####](#) for production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see [#####](#) and [#####](#)):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see [#####](#) [#####](#)):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See [Celery #####](#) [##](#) for more info:

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

8. Start the development server (see [#####](#) for production setup):

```
weblate runserver
```

After installation

Congratulations, your Weblate server is now running and you can start using it.

You can now access Weblate on `http://localhost:8000/`.

Login with admin credentials obtained during installation or register with new users.

You can now run Weblate commands using **weblate** command when Weblate virtualenv is active, see [Management commands](#).

You can stop the test server with Ctrl+C.

Review potential issues with your installation either on `/manage/performance/` URL or using **weblate check --deploy**, see [#####](#).

Adding translation

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See [Project configuration](#) for more details.

All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See [Component configuration](#) for more details.

The important fields here are: Component name, VCS repository address and mask for finding translatable files. Weblate supports a wide range of formats including gettext PO files, Android resource strings, iOS string properties, Java properties or Qt Linguist files, see [#####](#) for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

????????????

1. ??????????????????????????????????????:

Installing on Debian and Ubuntu

Installing on SUSE and openSUSE

Installing on RedHat, Fedora and CentOS

- ## 2. Git [Tutorial](#) Weblate [Tutorial](#) tarball [Tutorial](#):

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

Web <https://weblate.org/>

- ### 3.??? Weblate ??? virtualenv ????

```
. ~/weblate-env/bin/activate
pip install -e weblate-src
```

- 4.weblate/settings_example.py → weblate/settings.py ???????

- [illegible]

6. Weblate : *Weblate* 

- 7.Django :                               

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

???: ???

Installing on OpenShift

With the OpenShift Weblate template you can get your personal Weblate instance up and running in seconds. All of Weblate's dependencies are already included. PostgreSQL is set up as the default database and persistent volume claims are used.

You can find the template at <https://github.com/WeblateOrg/openshift/>.

The following examples assume you have a working OpenShift v3.x environment, with `oc` client tool installed. Please check the OpenShift documentation for instructions.

Web Console

Copy the raw content from `template.yml` and import them into your project, then use the **Create** button in the OpenShift web console to create your application. The web console will prompt you for the values for all of the parameters used by the template.

CLI

To upload the Weblate template to your current project's template library, pass the `template.yml` file with the following command:

```
$ oc create -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
  -n <PROJECT>
```

The template is now available for selection using the web console or the CLI.

?????

The parameters that you can override are listed in the parameters section of the template. You can list them with the CLI by using the following command and specifying the file to be used:

```
$ 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
```

????????

You can also use the CLI to process templates and use the configuration that is generated to create objects immediately.

```
$ 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
```

The Weblate instance should be available after successful migration and deployment at the specified `WEBLATE_SITE_DOMAIN` parameter.

After container setup, you can sign in as *admin* user with password provided in `WEBLATE_ADMIN_PASSWORD`, or a random password generated on first start if that was not set.

To reset *admin* password, restart the container with `WEBLATE_ADMIN_PASSWORD` set to new password in the respective Secret.

????

```
$ 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
```



By processing the template a respective `ConfigMap` will be created and which can be used to customize the Weblate image. The `ConfigMap` is directly mounted as environment variables and triggers a new deployment every time it is changed. For further configuration options, see *Docker environment variables* for full list of environment variables.

Installing on Kubernetes

NOTE: This guide is looking for contributors experienced with Kubernetes to cover the setup in more details.

With the Kubernetes Helm chart you can get your personal Weblate instance up and running in seconds. All of Weblate's dependencies are already included. PostgreSQL is set up as the default database and persistent volume claims are used.

You can find the chart at <https://github.com/WeblateOrg/helm/> and it can be displayed at <https://artifacthub.io/packages/helm/weblate/weblate>.



```
helm repo add weblate https://helm.weblate.org
helm install my-release weblate/weblate
```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:

Docker XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Virtualenv XXXXXXXXXXXXXXXXXXXX:

Installing on Debian and Ubuntu

Installing on SUSE and openSUSE

Installing on RedHat, Fedora and CentOS

Installing on macOS

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Installing on OpenShift

Installing on Kubernetes

XXXXXXXXXX

XXXXXXXXXX XXXXX

Weblate ☐Linux☐FreeBSD☐macOSXXXXXXXXXXXXXXXXXXXX Unix XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Weblate ☐ Windows XX

XXXXXXXXXX

Weblate XXXXXXXXXXXXXXXXXXXXXXXXXXXX:

PostgreSQL XXXXXX XXXXXXXX: *Weblate* XXXXXXXXXXXXX

XXXXXXXXXX XXXXX Redis XXXXXXXX: *Celery* XXXXXXXXXXXXXXXXXXXX XXXX

XXXXXXXXXX SMTP XXXXXXXX: XXXXXXXXXXXXX

Python 3.6

Weblate Python 3.6 requirements.txt

requirements.txt:

<https://www.djangoproject.com/>
<https://docs.celeryproject.org/>
<https://toolkit.translatehouse.org/>
<https://github.com/WeblateOrg/translation-finder>
<https://python-social-auth.readthedocs.io/>
<https://www.django-rest-framework.org/>

Python 3.5

Weblate requirements-optional.txt

<https://www.mercurial-scm.org/>
<https://github.com/viraptor/phply>
<https://github.com/sirfz/tesseract>
<https://github.com/ubernostrum/akismet>
<https://pypi.org/project/ruamel.yaml/>
<https://docs.python-zeep.org/>
<https://pypi.org/project/aeidon/>

Python 3.4

Weblate PostgreSQL MySQL MariaDB Weblate

Python 3.3

requirements.txt:

<https://git-scm.com/>
<https://cairographics.org/> <https://pango.gnome.org/>: Pango Cairo
<https://pypi.org/project/git-review/>
<https://git-scm.com/docs/git-svn>
<https://github.com/tesseract-ocr/tesseract>
<https://github.com/licensee/licensee>

Python 3.2

Python 3.2 requirements.txt
pip Wheels


```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX:

```

```

$ 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]

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:

```

```

$ 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]

```

```

???? ?????????

```

```

Weblate      ????????????????????????????????????????????????? -          DATA_DIR
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Weblat ????????? WSGI ? Celery??? :  ?????? ??? Celery
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX Weblate ?????????????????????????????????????????????????: /var/lib/weblate?

```

```

Weblate ?????????????????????????????????????????????????????????????????????????

```

```

???Management commands ?????????????????????????????????????????????????????????????????????????????????????????????????

```

```

Docker  ???????/app/data  ????????????????????????????????????? weblate  ?????????????????????????????????UID
1000??

```

```

??:

```

```

????????????

```

```

Weblate ?????????????

```

```

PostgreSQL ??????? ??????? Weblate ?????????????????????

```

```

??:

```

```

???????????? ?????????????????????????Migrating from other databases to PostgreSQL

```

PostgreSQL

```

???PostgreSQL ? Django ?????????????????????????????????????????????????????????????????????????

```

```

??:      Weblate  ?????????????????????????????????????????????????????????????????????????????????????????????????
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

```

??:

```

```

PostgreSQL notes

```

PostgreSQL 安装

安装 PostgreSQL 数据库:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser --superuser --pwprompt weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -O weblate weblate
```

安装 PostgreSQL 数据库:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
```

PostgreSQL 配置 Weblate

settings.py PostgreSQL 配置:

```
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": "",
    }
}
```

安装 PostgreSQL 数据库:

```
ALTER ROLE postgres WITH PASSWORD 'password' username
psycopg2.
errors.UndefinedObject: role "weblate@hostname" does not exist
Azure Database for PostgreSQL 安装 PostgreSQL 数据库:
ALTER_ROLE
```

MySQL 和 MariaDB

安装 PostgreSQL 数据库:

安装 PostgreSQL 数据库:

安装 PostgreSQL 数据库:

安装 PostgreSQL 数据库:

安装 PostgreSQL 数据库:

安装 PostgreSQL 数据库:

安装 PostgreSQL 数据库:

SQL `STRICT_TRANS_TABLES`

MySQL 8.x, MariaDB 10.5.x or newer have reasonable default configuration so that no server tweaking should be necessary and all what is needed can be configured on the client side.

8 GB RAM `/etc/my.cnf.d/server.cnf` MySQL
MariaDB

Weblate MySQL/MariaDB `innodb_file_per_table`

```
[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
```

1071: #1071 - Specified key was too long; max key length is 767 bytes
innodb

2006: In case you are getting #2006 - MySQL server has gone away error, configuring `CONN_MAX_AGE` might help.

MySQL/MariaDB Weblate

settings.py MySQL MariaDB:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "127.0.0.1",
        # Set to empty string for default
        "PORT": "3306",
        # In case you wish to use additional
        # connection options
        "OPTIONS": {},
    }
}
```

MySQL MariaDB weblate

```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

??????

??????????

Weblate ???SMTP ?????????????????

??? ??????????????????????????????????????: EMAIL_HOST?EMAIL_HOST_PASSWORD?EMAIL_USE_TLS?EMAIL_USE_SSL
??? EMAIL_PORT? ????????????????????????????????????? Django ?????????????????

???: ??????????????????????????????????????: SMTP AUTH extension not supported by
server??EMAIL_USE_TLS
????????????????????

??:

Not receiving e-mails from Weblate?Configuring outgoing e-mail in Docker container

????? ?????????????????

Weblate ????????????????????? IP ???Spam protection ??? ?????
????????

????????????WEBlate ? WSGI ?????????????????? REMOTE_ADDR ?? IP ?????????????????

????? ??? HTTP ??????????????????
IP ?????????????????? Webrate ??? IP
?? IP_BEHIND_REVERSE_PROXY
???????????????????????????????????? IP_PROXY_HEADER ??? IP_PROXY_OFFSET ?????????????????

??:

Spam protection? ?????? ?????? IP_BEHIND_REVERSE_PROXY? IP_PROXY_HEADER? IP_PROXY_OFFSET? SECURE_PRO

HTTP ??????

Weblate ? VCS ??? settings.py:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

??:

Proxy Environment Variables

?????

??:

???

weblate/settings_example.py? weblate/settings.py ??:
ADMINS

???????? Django ???

??:

ADMINS

ALLOWED_HOSTS

??:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

????????????????????????????:

```
ALLOWED_HOSTS = ["*"]
```

##:

ALLOWED_HOSTS=WEBLATE_ALLOWED_HOSTS

SESSION_ENGINE

#####: **weblate**
clearsessions #####

Redis #####: Redis #####:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

##:

#####SESSION_ENGINE

DATABASES

#####Django #####

##:

Weblate #####DATABASES#####

DEBUG

#####Django #####
ADMINS #####

#####Django #####Weblate #####

##:

DEBUG

DEFAULT_FROM_EMAIL

#####: #####

##:

DEFAULT_FROM_EMAIL

SECRET_KEY

Django Cookie #####Django #####

##:

SECRET_KEY

SERVER_EMAIL

#####: #####

##:

SERVER_EMAIL

#####

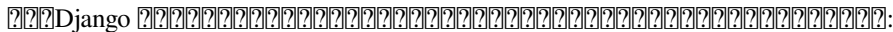
#####weblate migrate#####

#####weblate migrate --noinput #####createadmin#####ad-
min #####

Performance report

##:

##, #####

[illegible]

???? ??????????????????????

□□□□□□□□□□

Django `DEBUG` `DEBUG`:

[illegible]

□□□□

ADMINISTRATIVE INFORMATION

??:

123456789101112

SITE_DOMAIN ???????

??:

150

HTTPS [Tutorial](#)

Enable HTTPS [Tutorial](#) Weblate [Tutorial](#) `ENABLE_HTTPS` [Tutorial](#):

```
ENABLE_HTTPS = True
```

Tip: HSTS [Tutorial](#)SSL/HTTPS [Tutorial](#)

Tip:

`ENABLE_HTTPS` [Tutorial](#) [Tutorial](#)

SECURE_HSTS_SECONDS [Tutorial](#)

SSL [Tutorial](#)HTTP Strict Transport Security [Tutorial](#)settings.py `SECURE_HSTS_SECONDS` [Tutorial](#) 0 [Tutorial](#)

```
SECURE_HSTS_SECONDS = 0
```

[Tutorial](#)django.middleware.security.SecurityMiddleware `HTTP Strict Transport Security` [Tutorial](#)

Tip: [Tutorial](#) HTTP Strict Transport Security [Tutorial](#)

[Tutorial](#) [Tutorial](#)

PostgreSQL [Tutorial](#)Weblate [Tutorial](#) [Tutorial](#)

Tip:

Weblate [Tutorial](#)Migrating from other databases to PostgreSQL[Tutorial](#), [Tutorial](#)

[Tutorial](#)

CACHES [Tutorial](#) Django [Tutorial](#) Redis [Tutorial](#):

```
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",
        },
    },
}
```

Tip: Redis [Tutorial](#)Celery [Tutorial](#): Celery [Tutorial](#) [Tutorial](#)

Tip:

[Tutorial](#)Django's cache framework

Configuration

Django `ALLOWED_HOSTS` Weblate `WEBLATE_ALLOWED_HOSTS` `WEBLATE_ALLOWED_HOSTS`:

```
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,
        },
    },
}
```

##:

`ENABLE_AVATARS` `AVATAR_URL_PREFIX` `Avatars` `WEBLATE_ALLOWED_HOSTS` Django's cache framework

Configuration

Weblate `WEBLATE_ALLOWED_HOSTS` `WEBLATE_ALLOWED_HOSTS` `SERVER_EMAIL` `DEFAULT_FROM_EMAIL` `WEBLATE_ALLOWED_HOSTS`:

```
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"
```

##: Weblate `WEBLATE_ALLOWED_HOSTS` `EMAIL_BACKEND` `django.core.mail.backends.dummy.EmailBackend`

`WEBLATE_ALLOWED_HOSTS` `WEBLATE_ALLOWED_HOSTS` `WEBLATE_ALLOWED_HOSTS`

##:

`WEBLATE_ALLOWED_HOSTS` `EMAIL_BACKEND` `DEFAULT_FROM_EMAIL` `SERVER_EMAIL`

Configuration

Django `ALLOWED_HOSTS` `ALLOWED_HOSTS` `ALLOWED_HOSTS`

HTTP `Invalid HTTP_HOST header: '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS.`

##: Docker `WEBLATE_ALLOWED_HOSTS` `WEBLATE_ALLOWED_HOSTS`

##:

`ALLOWED_HOSTS` `WEBLATE_ALLOWED_HOSTS` `WEBLATE_ALLOWED_HOSTS`

Django 设置

`SECRET_KEY` Django 的 cookie 是一个随机字符串，可以在 `weblate/examples/generate-secret-key` 中找到。

：

`SECRET_KEY`

安装 Weblate

2.1 安装：Weblate 需要 `DATA_DIR` 目录。

Weblate 需要 `SSH` 和 `Git` 支持。Weblate 需要 `SSH` 和 `Git` 支持。

Weblate 需要 `settings.py` 文件。Weblate 需要 `configuration` 文件。

```
os.environ["HOME"] = os.path.join(BASE_DIR, "configuration")
```

： Linux 和 UNIX 系统需要 `/etc/passwd` 文件。Weblate 需要 `WEB` 目录。Weblate 需要 `www-data` 用户。Weblate 需要 `wwwrun` 用户。

：

`weblate`

配置 Weblate

Django 需要 `loaders` 配置。

```
TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [
            os.path.join(BASE_DIR, "templates"),
        ],
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
            "loaders": [
                (
                    "django.template.loaders.cached.Loader",
                    [
                        "django.template.loaders.filesystem.Loader",
                        "django.template.loaders.app_directories.Loader",
                    ],
                ),
            ],
        },
    ],
]
```

：

`django.template.loaders.cached.Loader`

Environment Variables

Environment variables are used to configure the application. The following variables are used in the Celery configuration:

Environment variables:

Environment variables: *Lazy commits* `commit_pending`

Environment variables:

Environment variables: *AUTO_UPDATE*

JSON Environment variables: *dump_memory*

Environment variables: *cleanuptrans*

Environment 3.2: *Environment 3.2* *Celery* *Webplate*
Environment variables: *Celery* *Environment*

Environment Variables

Environment UTF-8 Environment Linux Environment
UTF-8 Environment

Environment/etc/default/locale Environment`LANG="C.UTF-8"`Environment

EnvironmentApache Environment /etc/apache2/envvars Environment:

```
export LANG='en_US.UTF-8'  
export LC_ALL='en_US.UTF-8'
```

Environment Variables

Webplate HTTP Environment SSL Environment

Environment Environment Debian CA /usr/local/share/ca-certificates/ Environment **update-ca-certificates** Environment

Environment Environment Git Environment

Python Environment CA Environment CA Environment
settings.py Environment Debian Environment:

```
import os  
os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Environment Variables

Webplate JavaScript CSS Environment
Webplate Environment

Environmentdjango.conf.settings.COMPRESS_OFFLINE Environmentdjango.conf.settings.COMPRESS_OFFLINE_CONTEXT` Environment:

```
COMPRESS_OFFLINE = True
```

Environment:

```
weblate compress
```

Environment: Docker Environment

Environment:

Common Deployment ScenariosEnvironment

??????

???: In case you are not experienced with services described below, you might want to try *Docker* [????????????](#).

Weblate [????????????????](#):

[??????](#) [????????](#): *Weblate* [????????????](#) ?

[??????](#) [????????](#): [????????????](#) ?

[????????](#) SSL [????????????????????](#) Web [????????](#): [????????????](#) ?

[????????????](#) WSGI [????????](#): *NGINX* [???](#) *uWSGI* [????](#) ?

[??????????](#) [????????????????????](#): *Celery* [????????????????](#) [???](#) ?

??: [????????????????????????????????](#) Celery [???](#) uwsgi [????????????????????????????](#)
[????????????????????????](#)

[??](#) Celery
[??????](#) Wsgi [??????](#) *DATA_DIR* [????????????????](#)

??: [????????](#) WSGI [??????](#) Celery [????????????????????????????????](#) *DATA_DIR*
[??](#)

[????](#) [????????????](#) [???](#) *Celery* [????????????????](#) [???](#) [????????](#)

Web ????????

Weblate [????????](#) Django [????????????????????????](#) Django ? *uWSGI* [???](#) *fcgi* [????????????](#) Web
[????????????](#)

[????????](#) Django [????????](#) Web [????????](#):

```
weblate runserver
```

??: [????????????????????????????????](#) runserver ? Django
[????????????????](#)

???: Django [????????????????????](#) *DEBUG* [????????????????????](#) *NGINX*
[???](#) *uWSGI* [?????](#) *Apache* [?????](#) *Apache* ? *Gunicorn* [????](#) [????????](#) [????????](#)

??????????

[??????](#) 2.4 [???](#): [??????](#) 2.4 [??????](#) Weblate ? Django [??????](#) [????????????????????](#)

Django [????????](#) 1 [????????????????????](#) `weblate collectstatic --noinput`
[????????](#) *STATIC_ROOT* [????????????????](#) *DATA_DIR* [??](#)
`static` [????](#)

[????????](#) Web [????????????????](#):

Weblate [????????????????](#) *STATIC_ROOT* [????](#)

[????????](#): [????????](#)

`/static/favicon.ico` [????????](#)

??:

NGINX [???](#) *uWSGI* [????](#), *Apache* [?????](#) *Apache* ? *Gunicorn* [????](#) [????](#) Django
[????????](#)

Webplate 安装与配置

Webplate 安装时，默认会安装 `weblate.middleware.SecurityMiddleware` 和 `Content-Security-Policy`，并启用 `X-XSS-Protection` 和 `HTTP` 头。

配置：

`CSP_SCRIPT_SRC` `CSP_IMG_SRC` `CSP_CONNECT_SRC` `CSP_STYLE_SRC` `CSP_FONT_SRC`

NGINX 与 uWSGI 配置

Webplate 安装时，默认会安装 `weblate.middleware.SecurityMiddleware` 和 `Content-Security-Policy`，并启用 `X-XSS-Protection` 和 `HTTP` 头。

配置 NGINX 和 uWSGI 的 Webplate 安装。

NGINX 配置 `weblate/examples/weblate.nginx.conf`：

```
# This example assumes Weblate is installed in virtualenv in /home/weblate/
# weblate-env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match
# your setup.
server {
    listen 80;
    server_name weblate;
    # Not used
    root /var/www/html;

    location ~ ^/favicon.ico$ {
        # DATA_DIR/static/favicon.ico
        alias /home/weblate/data/static/favicon.ico;
        expires 30d;
    }

    location /static/ {
        # DATA_DIR/static/
        alias /home/weblate/data/static/;
        expires 30d;
    }

    location /media/ {
        # DATA_DIR/media/
        alias /home/weblate/data/media/;
        expires 30d;
    }

    location / {
        include uwsgi_params;
        # Needed for long running operations in admin interface
        uwsgi_read_timeout 3600;
        # Adjust based to uwsgi configuration:
        uwsgi_pass unix:///run/uwsgi/app/weblate/socket;
        # uwsgi_pass 127.0.0.1:8080;
    }
}
```

uWSGI 配置 `weblate/examples/weblate.uwsgi.ini`：

```
# This example assumes Weblate is installed in virtualenv in /home/weblate/
# weblate-env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match
# your setup.
[uwsgi]
plugins      = python3
master       = true
protocol     = uwsgi
```

(20220202)

(XXXXXXXXXX)

```
socket          = 127.0.0.1:8080
wsgi-file       = /home/weblate/weblate-env/lib/python3.7/site-packages/
↳weblate/wsgi.py

# Add path to Weblate checkout if you did not install
# Weblate by pip
# python-path    = /path/to/weblate

# In case you're using virtualenv uncomment this:
virtualenv      = /home/weblate/weblate-env

# Needed for OAuth/OpenID
buffer-size     = 8192

# Reload when consuming too much of memory
reload-on-rss   = 250

# Increase number of workers for heavily loaded sites
workers         = 8

# Enable threads for Sentry error submission
enable-threads  = true

# Child processes do not need file descriptors
close-on-exec   = true

# Avoid default 0000 umask
umask           = 0022

# Run as weblate user
uid = weblate
gid = weblate

# Enable harakiri mode (kill requests after some time)
# harakiri       = 3600
# harakiri-verbose = true

# Enable uWSGI stats server
# stats          = :1717
# stats-http     = true

# Do not log some errors caused by client disconnects
ignore-sigpipe  = true
ignore-write-errors = true
disable-write-exception = true
```

##:

Django 2 uWSGI XXXXXXXXXXXX

Apache ????

Weblate 2 WSGI XXXXXXXXXXXXprefork MPM XXXXXXXXXXXX

###Weblate 2 WSGI XXXXXXXXXXXXmod_wsgi XXXXXXXXXXXXweblate/examples/apache.conf
#####:

```
#
# VirtualHost for Weblate
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/
↳weblate-env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match.
↳your setup.
#
<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
```

(XXXXXXXXXX)

(XXXXXXXXXX)

```
ServerName weblate.example.org

# DATA_DIR/static/favicon.ico
Alias /favicon.ico /home/weblate/data/static/favicon.ico

# DATA_DIR/static/
Alias /static/ /home/weblate/data/static/
<Directory /home/weblate/data/static/>
    Require all granted
</Directory>

# DATA_DIR/media/
Alias /media/ /home/weblate/data/media/
<Directory /home/weblate/data/media/>
    Require all granted
</Directory>

# Path to your Weblate virtualenv
WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env_
↪user=weblate
WSGIProcessGroup weblate
WSGIApplicationGroup %{GLOBAL}

WSGIScriptAlias / /home/weblate/weblate-env/lib/python3.7/site-
↪packages/weblate/wsgi.py process-group=weblate request-timeout=600
WSGIPassAuthorization On

<Directory /home/weblate/weblate-env/lib/python3.7/site-packages/
↪weblate/>
    <Files wsgi.py>
        Require all granted
    </Files>
</Directory>

</VirtualHost>
```

##: Weblate ## Python3 #####modwsgi ## Python 3 variant #####libapache2-
mod-wsgi-py3 #####

##:

#####Django ## Apache ## mod_wsgi #####

Apache ## Gunicorn

#####Gunicorn ### Apache 2.4 ## Weblate #####weblate/examples/apache.gunicorn.conf
#####:

```
#
# VirtualHost for Weblate using gunicorn on localhost:8000
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/
↪weblate-env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match_
↪your setup.
#
<VirtualHost *:443>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /static/ /home/weblate/data/static/
```

(XXXXXXXXXX)

(XXXXXXXXXX)

```
<Directory /home/weblate/data/static/>
    Require all granted
</Directory>

# DATA_DIR/media/
Alias /media/ /home/weblate/data/media/
<Directory /home/weblate/data/media/>
    Require all granted
</Directory>

SSLEngine on
SSLCertificateFile /etc/apache2/ssl/https_cert.cert
SSLCertificateKeyFile /etc/apache2/ssl/https_key.pem
SSLProxyEngine On

ProxyPass /favicon.ico !
ProxyPass /static/ !
ProxyPass /media/ !

ProxyPass / http://localhost:8000/
ProxyPassReverse / http://localhost:8000/
ProxyPreserveHost On
</VirtualHost>
```

??:

How to use Django with Gunicorn

?????? Weblate ???

????? 1.3 ???.

Weblate ? WSGI ?????????prefork MPM ?????????

Weblate ? /weblate ?????? Apache ?????? ``mod\_wsgi`` ???weblate/examples/apache-path.conf ??????:

```
#
# VirtualHost for Weblate, running under /weblate path
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/
# ↳weblate-env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match
# ↳your setup.
#
<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /weblate/favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /weblate/static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>

    # DATA_DIR/media/
    Alias /weblate/media/ /home/weblate/data/media/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env
    ↳user=weblate
```

(XXXXXXXXXX)

(XXXXXXXXXX)

```
WSGIProcessGroup weblate
WSGIApplicationGroup %{GLOBAL}

WSGIScriptAlias /weblate /home/weblate/weblate-env/lib/python3.7/site-
↪packages/weblate/wsgi.py process-group=weblate request-timeout=600
WSGIPassAuthorization On

<Directory /home/weblate/weblate-env/lib/python3.7/site-packages/
↪weblate/>
    <Files wsgi.py>
        Require all granted
    </Files>
</Directory>

</VirtualHost>
```

weblate/settings.py

```
URL_PREFIX = "/weblate"
```

Celery

3.2

Weblate Celery Redis:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

:

Redis broker configuration in Celery

Eager Weblate:

```
CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True
```

Celery:

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

: Celery WSGI DATA_DIR

Celery

Celery Daemonization systemd Linux

/etc/systemd/system/celery-weblate.service Systemd:

```
[Unit]
Description=Celery Service (Weblate)
After=network.target

[Service]
Type=forking
User=weblate
Group=weblate
```

(XXXXXXXXXX)

(???????????)

```
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 ??????????????:

```
# Name of nodes to start
CELERYD_NODES="celery notify memory backup translate"

# Absolute or relative path to the 'celery' command:
CELERY_BIN="/home/weblate/weblate-env/bin/celery"

# App instance to use
# comment out this line if you don't use an app
CELERY_APP="weblate.utils"

# Extra command-line arguments to the worker,
# increase concurrency if you get weblate.E019
CELERYD_OPTS="--beat:celery --queues:celery=celery --prefetch-
↪multiplier:celery=4 \
  --queues:notify=notify --prefetch-multiplier:notify=10 \
  --queues:memory=memory --prefetch-multiplier:memory=10 \
  --queues:translate=translate --prefetch-multiplier:translate=4 \
  --concurrency:backup=1 --queues:backup=backup --prefetch-
↪multiplier:backup=2"

# Logging configuration
# - %n will be replaced with the first part of the nodename.
# - %I will be replaced with the current child process index
# and is important when using the prefork pool to avoid race conditions.
CELERYD_PID_FILE="/run/celery/weblate-%n.pid"
CELERYD_LOG_FILE="/var/log/celery/weblate-%n%I.log"
CELERYD_LOG_LEVEL="INFO"

# Internal Weblate variable to indicate we're running inside Celery
CELERY_WORKER_RUNNING="1"
```

/etc/logrotate.d/celery ??? Celery ??? logrotate ??????????????????????????????:

```
/var/log/celery/*.log {
    weekly
    missingok
    rotate 12
    compress
    notifempty
}
```

Celery beat [Celery beat](#)

Weblate [Celery beat](#) settings.py [Lazy commits](#)

[Celery beats](#)

Celery [Celery](#)

[celery_queues](#) Celery [Celery](#)

??: [Celery](#) Celery [Celery](#)

??:

[Configuration and defaults](#) [Workers](#) [Guide](#) [Daemonization](#) [Monitoring and Management](#) [celery_queues](#)

Weblate [Weblate](#)

Weblate [Kubernetes](#) [/healthz/ URL](#)

???

Weblate [Weblate](#)

Sentry

Weblate [Sentry](#) [https://sentry.io/](#) settings.py [SENTRY_DSN](#)

SENTRY_DSN = "https://id@your.sentry.example.com/"

Rollbar

Weblate [Rollbar](#) [Rollbar notifier for Python](#)

settings.py

```
# Add rollbar as last middleware:
MIDDLEWARE = [
    # ... other middleware classes ...
    "rollbar.contrib.django.middleware.RollbarNotifierMiddleware",
]

# Configure client access
ROLLBAR = {
    "access_token": "POST_SERVER_ITEM_ACCESS_TOKEN",
    "client_token": "POST_CLIENT_ITEM_ACCESS_TOKEN",
    "environment": "development" if DEBUG else "production",
    "branch": "main",
    "root": "/absolute/path/to/code/root",
}
```

Weblate [Getting started](#)

Weblate [Getting started](#)
Weblate [Getting started](#)

[Getting started](#)

[Getting started](#) [Getting started](#)1 [Getting started](#)
[Getting started](#): `mysqldump`Getting started`:com-`
`mand:`pg_dumpGetting startedDjango Getting started:`

```
# Export current data
weblate dumpdata > /tmp/weblate.dump
# Import dump
weblate loaddata /tmp/weblate.dump
```

VCS [Getting started](#)

`DATA_DIR` [Getting started](#) VCS [Getting started](#) `rsync`
[Getting started](#)

[Getting started](#)

Redis [Getting started](#) Cron [Getting started](#)Weblate [Getting started](#)

Weblate [Getting started](#)

Weblate [Getting started](#):

Docker [Getting started](#)

Installing on OpenShift

Installing on Kubernetes

[Getting started](#) Weblate [Getting started](#)

[Getting started](#): [Getting started](#)Weblate [Getting started](#)

Bitnami Weblate [Getting started](#)

Bitnami [Getting started](#)<<https://bitnami.com/stack/weblate>> [Getting started](#) Weblate
[Getting started](#)<<https://bitnami.com/stack/weblate/README.txt>>
[Getting started](#)

Weblate Cloudron [Getting started](#)

Cloudron [Getting started](#) Web [Getting started](#)Cloudron [Getting started](#) Weblate
[Getting started](#) Cloudron [Getting started](#) Weblate package repo [Getting started](#)



YunoHost Weblate

YunoHost  Weblate YunoHost Weblate
Weblate
:



:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

Weblate

Docker

Docker Docker: Docker

settings.py settings_example.py

Always check [Version specific instructions](#) before upgrade. In case you are skipping some versions, please follow instructions for all versions you are skipping in the upgrade. Sometimes it's better to upgrade to some intermediate version to ensure a smooth migration. Upgrading across multiple releases should work, but is not as well tested as single version upgrades.

⚠: It is recommended to perform a full database backup prior to upgrade so that you can roll back the database in case upgrade fails, see [Weblate](#).

1. Stop wsgi and Celery processes. The upgrade can perform incompatible changes in the database, so it is always safer to avoid old processes running while upgrading.

2. Upgrade Weblate code.

For pip installs it can be achieved by:

```
pip install -U Weblate
```

With Git checkout you need to fetch new source code and update your installation:

```
cd weblate-src
git pull
# Update Weblate inside your virtualenv
. ~/weblate-env/bin/pip install -e .
# Install dependencies directly when not using virtualenv
pip install --upgrade -r requirements.txt
```

3. Upgrade configuration file, refer to settings_example.py or [Version specific instructions](#) for needed steps.

4. Upgrade database structure:

```
weblate migrate --noinput
```

5. Collect updated static files (see and):

```
weblate collectstatic --noinput
```

6. Compress JavaScript and CSS files (optional, see):

```
weblate compress
```

7. Git

```
weblate compilemessages
```

8. Verify that your setup is sane (see also [????????](#)):

```
weblate check --deploy
```

9. Restart celery worker (see *Celery* [????????????????](#) [???](#)).

Version specific instructions

Upgrade from 2.x

If you are upgrading from 2.x release, always first upgrade to 3.0.1 and then continue upgrading in the 3.x series. Upgrades skipping this step are not supported and will break.

??:

[Upgrade from 2.20 to 3.0 in Weblate 3.0 documentation](#)

Upgrade from 3.x

If you are upgrading from 3.x release, always first upgrade to 4.0.4 or 4.1.1 and then continue upgrading in the 4.x series. Upgrades skipping this step are not supported and will break.

??:

[Upgrade from 3.11 to 4.0 in Weblate 4.0 documentation](#)

Upgrade from 4.0 to 4.1

Please follow [????????????????](#) in order to perform update.

Notable configuration or dependencies changes:

There are several changes in `settings_example.py`, most notable middleware changes, please adjust your settings accordingly.

There are new file formats, you might want to include them in case you modified the `WEBLATE_FORMATS`.

There are new quality checks, you might want to include them in case you modified the `CHECK_LIST`.

`DEFAULT_THROTTLE_CLASSES` [????????API](#) [????????????????????](#)

There are some new and updated requirements.

There is a change in `INSTALLED_APPS`.

The *DeepL* machine translation now defaults to v2 API, you might need to adjust `MT_DEEPL_API_VERSION` in case your current DeepL subscription does not support that.

??:

[????????????????](#)

Upgrade from 4.1 to 4.2

Please follow [????????????????](#) in order to perform update.

Notable configuration or dependencies changes:

Upgrade from 3.x releases is not longer supported, please upgrade to 4.0 or 4.1 first.

There are some new and updated requirements.

There are several changes in `settings_example.py`, most notable new middleware and changed application ordering.

The keys for JSON based formats no longer include leading dot. The strings are adjusted during the database migration, but external components might need adjustment in case you rely on keys in exports or API.

The Celery configuration was changed to no longer use `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.

The Weblate domain is now configured in the settings, see `SITE_DOMAIN` (or `WEBLATE_SITE_DOMAIN`). You will have to configure it before running Weblate.

The username and email fields on user database now should be case insensitive unique. It was mistakenly not enforced with PostgreSQL.

??:

????????????????

Upgrade from 4.2 to 4.3

Please follow [????????????????](#) in order to perform update.

Notable configuration or dependencies changes:

There are some changes in quality checks, you might want to include them in case you modified the `CHECK_LIST`.

The source language attribute was moved from project to a component what is exposed in the API. You will need to update *Weblate* [???????](#) in case you are using it.

The database migration to 4.3 might take long depending on number of strings you are translating (expect around one hour of migration time per 100,000 source strings).

There is a change in `INSTALLED_APPS`.

There is a new setting `SESSION_COOKIE_AGE_AUTHENTICATED` which complements `SESSION_COOKIE_AGE`.

In case you were using **hub** or **lab** to integrate with GitHub or GitLab, you will need to reconfigure this, see `GITHUB_CREDENTIALS` and `GITLAB_CREDENTIALS`.

????? 4.3.1 **???**: The Celery configuration was changed to add `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.

????? 4.3.2 **???**: The `post_update` method of addons now takes extra `skip_push` parameter.

??:

????????????????

Upgrade from 4.3 to 4.4

Please follow [????????????????](#) in order to perform update.

Notable configuration or dependencies changes:

There is a change in `INSTALLED_APPS`, `weblate.configuration` has to be added there.

Django 3.1 is now required.

In case you are using MySQL or MariaDB, the minimal required versions have increased, see *MySQL* [?](#) *MariaDB*.

????? 4.4.1 **???**: *Monolingual gettext* now uses both `msgid` and `msgctxt` when present. This will change identification of translation strings in such files breaking links to Weblate extended data such as screenshots or review states. Please make sure you commit pending changes in such files prior upgrading and it is recommended to force loading of affected component using *loadpo*. Increased minimal required version of `translate-toolkit` to address several file format issues.

??:

????????????????

Upgrade from 4.4 to 4.5

Please follow [????????????????](#) in order to perform update.

Notable configuration or dependencies changes:

The migration might take considerable time if you had big glossaries.

Glossaries are now stored as regular components.

The glossary API is removed, use regular translation API to access glossaries.

There is a change in `INSTALLED_APPS` - `weblate.metrics` should be added.

????? 4.5.1 **???**: There is a new dependency on the *pyahocorasick* module.

❗:

🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗

Upgrade from 4.5 to 4.6

Please follow 🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗 in order to perform update.

Notable configuration or dependencies changes:

There are new file formats, you might want to include them in case you modified the `WEBLATE_FORMATS`.

🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗 API 🔗🔗🔗🔗🔗 `Weblate 🔗🔗🔗 URL 🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗`: `POST /api/projects/(string:project)/components/🔗`

There is a change in dependencies and `PASSWORD_HASHERS` to prefer Argon2 for passwords hashing.

❗:

🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗

Upgrading from Python 2 to Python 3

Weblate no longer supports Python older than 3.5. In case you are still running on older version, please perform migration to Python 3 first on existing version and upgrade later. See [Upgrading from Python 2 to Python 3](#) in the [Weblate 3.11.1 documentation](#).

Migrating from other databases to PostgreSQL

If you are running Weblate on other database than PostgreSQL, you should migrate to PostgreSQL as that will be the only supported database backend in the 4.0 release. The following steps will guide you in migrating your data between the databases. Please remember to stop both web and Celery servers prior to the migration, otherwise you might end up with inconsistent data.

PostgreSQL 🔗🔗🔗🔗🔗🔗🔗🔗

🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗 Weblate 🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗 🔗🔗🔗🔗🔗🔗🔗:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser -D -P weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -O weblate weblate
```

Migrating using Django JSON dumps

The simplest approach for migration is to utilize Django JSON dumps. This works well for smaller installations. On bigger sites you might want to use pgloader instead, see [Migrating to PostgreSQL using pgloader](#).

1. Add PostgreSQL as additional database connection to the `settings.py`:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
```

(🔗🔗🔗🔗🔗🔗🔗)

(XXXXXXXXXX)

```

        # Set to empty string for default
        "PORT": "",
        # Additional database options
        "OPTIONS": {
            # In case of using an older MySQL server, which has MyISAM as
            ↪ a default storage
            # 'init_command': 'SET storage_engine=INNODB',
            # Uncomment for MySQL older than 5.7:
            # 'init_command': "SET sql_mode='STRICT_TRANS_TABLES'",
            # If your server supports it, see the Unicode issues above
            "charset": "utf8mb4",
            # Change connection timeout in case you get MySQL gone away
            ↪ error:
            "connect_timeout": 28800,
        },
    },
    "postgresql": {
        # Database engine
        "ENGINE": "django.db.backends.postgresql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
    },
}

```

2.Run migrations and drop any data inserted into the tables:

```

weblate migrate --database=postgresql
weblate sqlflush --database=postgresql | weblate dbshell --
↪ database=postgresql

```

3.Dump legacy database and import to PostgreSQL

```

weblate dumpdata --all --output weblate.json
weblate loaddata weblate.json --database=postgresql

```

4.Adjust `DATABASES` to use just PostgreSQL database as default, remove legacy connection.

Weblate should be now ready to run from the PostgreSQL database.

Migrating to PostgreSQL using pgloader

The `pgloader` is a generic migration tool to migrate data to PostgreSQL. You can use it to migrate Weblate database.

1.Adjust your `settings.py` to use PostgreSQL as a database.

2.Migrate the schema in the PostgreSQL database:

```

weblate migrate
weblate sqlflush | weblate dbshell

```

3.Run the `pgloader` to transfer the data. The following script can be used to migrate the database, but you might want to learn more about `pgloader` to understand what it does and tweak it to match your setup:

```

LOAD DATABASE
FROM      mysql://weblate:password@localhost/weblate
INTO      postgresql://weblate:password@localhost/weblate

WITH include no drop, truncate, create no tables, create no indexes, no
↪ foreign keys, disable triggers, reset sequences, data only

```

(XXXXXXXXXX)

()

```
ALTER SCHEMA 'weblate' RENAME TO 'public';
```

Migrating from Pootle

As Weblate was originally written as replacement from Pootle, it is supported to migrate user accounts from Pootle. You can dump the users from Pootle and import them using *importusers*.

Weblate

BorgBackup ??????????????

□□□□□ 3.9 □□□.

Webplate BorgBackup Borg
                                                             

4.4.1 PostgreSQL & MySQL/MariaDB

14

□□□□□□□□□□ 6 □□

□□□□□□□□□□ 6 □□□

Webplate

Dashboard

Projects

Languages

Checks

Manage / Backups

Backup process triggered

Webplate status

Backups

Translation memory

Performance report

SSH keys

Alerts

Repositories

Users

Appearance

Tools

Billing

Backup service: /tmp/tmpkrnjny6webplate

Backup service credentials

April 18, 2021

Backup repository

/tmp/tmpkrnjny6webplate

Passphrase

ZbTHOQCyWyE085zWc1xR1G(pr*UaPFg\$TDhVMYfbpBBCj)0A2g

The passphrase is used to encrypt the backups and is necessary to restore them.

SSH key

Download private key

The private key is needed to access the remote backup repository.

Deleted the oldest backups

April 18, 2021

Backup performed

April 18, 2021

Repository initialization

April 18, 2021

Turn off

Perform backup

Delete

Activate support package

The support packages include priority e-mail support, or cloud backups of your Webplate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate

Purchase support package

Add backup service

Backup repository URL

Use /path/to/repo for local backups or user@host:/path/to/repo for remote SSH backups.

Add

Powered by Webplate 4.6

About Webplate

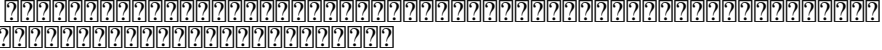
Legal

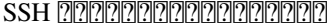
Contact

Documentation

Donate to Webplate

Borg

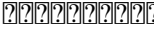
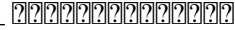
BorgBackup 

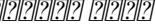
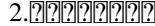
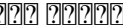
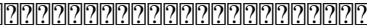
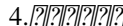
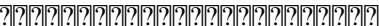
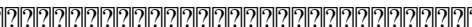
Weblate   SSH 

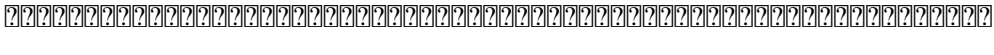
:

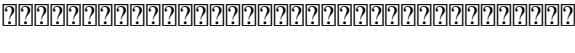
borg init

Weblate

Weblate  backup service at weblate.org _ :

1. <https://weblate.org/support/#backup>  
2. : 
3. Weblate  
4.  
5. Borg : *Borg* 

: 

SSH  BorgBackup

:

Borg General

It is recommended to specify the absolute path for the local backup, for example `/path/to/backup`. The directory has to be writable by the user running Weblate (see  ). If it doesn't exist, Weblate attempts to create it but needs the appropriate permissions to do so.

: When running Weblate in Docker, please ensure the backup location is exposed as a volume from the Weblate container. Otherwise the backups will be discarded by Docker upon restarting the container it is in.

One option is to place backups into an existing volume, for example `/app/data/borgbackup`. This is an existing volume in the container.

You can also add a new container for the backups in the Docker Compose file for example by using `/borgbackup`:

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

The directory where backups will be stored have to be owned by UID 1000, otherwise Weblate won't be able to write the backups there.

Native database backup

The recommended approach is to save a dump of the database using database-native tools such as `pg_dump` or `mysqldump`. It usually performs better than Django backup, and it restores complete tables with all their data.

You can restore this backup in a newer Weblate release, it will perform all the necessary migrations when running in `migrate`. Please consult [Weblate `UPGRADING`](#) on more detailed info on how to upgrade between versions.

Django `dumpdata` `loaddata`

Alternatively, you can back up your database using Django's `dumpdata` command. That way the backup is database agnostic and can be used in case you want to change the database backend.

Prior to restoring the database you need to be running exactly the same Weblate version the backup was made on. This is necessary as the database structure does change between releases and you would end up corrupting the data in some way. After installing the same version, run all database migrations using `migrate`.

Afterwards some entries will already be created in the database and you will have them in the database backup as well. The recommended approach is to delete such entries manually using the management shell (see [Invoking management commands](#)):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

`LANGUAGES`

If you have enough backup space, simply back up the whole `DATA_DIR`. This is a safe bet even if it includes some files you don't want. The following sections describe what you should back up and what you can skip in detail.

Dumped data for backups

Stored in `DATA_DIR/backups`.

Weblate dumps various data here, and you can include these files for more complete backups. The files are updated daily (requires a running Celery beats server, see [Celery `beat` `worker`](#)). Currently, this includes:

Weblate settings as `settings.py` (there is also expanded version in `settings-expanded.py`).

PostgreSQL database backup as `database.sql`.

The database backups are saved as plain text by default, but they can also be compressed or entirely skipped using `DATABASE_BACKUP`.

Version control repositories

Stored in `DATA_DIR/vcs`.

The version control repositories contain a copy of your upstream repositories with Weblate changes. If you have *Push on commit* enabled for all your translation components, all Weblate changes are included upstream. No need to back up the repositories on the Weblate side as they can be cloned again from the upstream location(s) with no data loss.

SSH and GPG keys

Stored in `DATA_DIR/ssh` and `DATA_DIR/home`.

If you are using SSH or GPG keys generated by Weblate, you should back up these locations. Otherwise you will lose the private keys and you will have to regenerate new ones.

GitHub `WEBSITE_URL=https://example.com/accounts/complete/github/`

`WEBSITE_URL=https://example.com/accounts/complete/github/`

`WEBSITE_URL=https://example.com/accounts/complete/github/`

GitHub

Bitbucket

`WEBSITE_URL=https://example.com/accounts/complete/bitbucket/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.bitbucket.BitbucketOAuth",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_BITBUCKET_KEY = "Bitbucket Client ID"
SOCIAL_AUTH_BITBUCKET_SECRET = "Bitbucket Client Secret"
SOCIAL_AUTH_BITBUCKET_VERIFIED_EMAILS_ONLY = True
```

`WEBSITE_URL=https://example.com/accounts/complete/bitbucket/`

`WEBSITE_URL=https://example.com/accounts/complete/bitbucket/`

Bitbucket

Google OAuth 2

Google OAuth 2 `WEBSITE_URL=https://example.com/accounts/complete/google-oauth2/`

`WEBSITE_URL=https://example.com/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"
```

`WEBSITE_URL=https://example.com/accounts/complete/google-oauth2/`

`WEBSITE_URL=https://example.com/accounts/complete/google-oauth2/`

Google

Facebook OAuth 2

OAuth 2 Facebook Webrate:
URL `https://WEBLATE_SERVER/accounts/complete/facebook/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.facebook.FacebookOAuth2",
    "social_core.backends.email.EmailAuth",
    "webrate.accounts.auth.WebrateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_FACEBOOK_KEY = "key"
SOCIAL_AUTH_FACEBOOK_SECRET = "secret"
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
```

: Webrate URL URL
URL

:
Facebook

GitLab OAuth 2

GitLab OAuth 2<<https://gitlab.com/profile/applications>>
URL `https://WEBLATE_SERVER/accounts/complete/gitlab/` `read_user`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.gitlab.GitLabOAuth2",
    "social_core.backends.email.EmailAuth",
    "webrate.accounts.auth.WebrateUserBackend",
)

# 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/'
```

: Webrate URL URL
URL

:
GitLab

Microsoft Azure Active Directory

Weblate `XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX`

`XXXXXXXXXX` URL `XXXXXXXX` `https://WEBLATE_SERVER/accounts/complete/azuread-oauth2/`
`XXXXXXXXXXXXXXXXXX` `https://WEBLATE_SERVER/accounts/complete/azuread-tenant-oauth2/`
`XXX`

```
# 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 = ""
```

`XX:` `XXXX` Weblate `XXXXXXXXXX` URL `XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX`URL
`XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX` `XXXXXXXXXXXX`

`XX:`

Microsoft Azure Active Directory

Slack

Slack OAuth 2 `XXXXXXXXXX`<`https://api.slack.com/apps`> `XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX`

`XXXXXXXXXX` URL `XX``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 = ""
```

`XX:` `XXXX` Weblate `XXXXXXXXXX` URL `XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX`URL
`XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX` `XXXXXXXXXXXX`

`XX:`

Slack

Authentication

```
AUTHENTICATION_BACKENDS = (
    social_core.backends.email.EmailAuth
    weblate.accounts.auth.WeblateUserBackend
    Weblate
)
```

Example: `weblate/accounts/auth/weblateuserbackend.py/admin/`

openSUSE Open ID configuration:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

Settings

Example: `settings.py` `AUTH_PASSWORD_VALIDATORS`:

```
AUTH_PASSWORD_VALIDATORS = (
```

```
    "django.contrib.auth.password_validation.MinimumLength",
```

```
    "django.contrib.auth.password_validation.CommonPasswordCheck",
```

```
    "django.contrib.auth.password_validation.NumericPasswordCheck",
```

```
    "django.contrib.auth.password_validation.UserAttributeSimilarity",
```

```
    "django.contrib.auth.password_validation.CommonPasswordCheck",
```

```
    "django.contrib.auth.password_validation.UserAttributeSimilarity",
```

```
    "django_zxcvbn_password.ZxcvbnPasswordValidator",
)
```

SAML

Example 4.1.1: SAML.

Example: Python Social Auth configuration:

```
Weblate SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
```

```
        "saml": {
            "url": "/accounts/metadata/saml/"
```

```
        "SOCIAL_AUTH_SAML_SP_PUBLIC_CERT": "SOCIAL_AUTH_SAML_SP_PUBLIC_CERT",
        "SOCIAL_AUTH_SAML_SP_PRIVATE_KEY": "SOCIAL_AUTH_SAML_SP_PRIVATE_KEY",
        "SOCIAL_AUTH_SAML_SUPPORT_CONTACT": "SOCIAL_AUTH_SAML_SUPPORT_CONTACT"
```

```
    },
```

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.saml.SAMLAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEEDjCCAvagAwIBAgIBADA ... 8Bbn1+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
```


???? ?????????

??:

```
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 ???

```
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
```

??:

Django Authentication Using LDAP Authentication

CAS ??

CAS ??? django-cas-ng ?????????????????????????????????????

???? 1?CAS ???CAS ?????????????????????????CAS v 1
????????????????????????CAS v 2 ?????????

???? 2?Weblate ?????? CAS ?????????????????????

django-cas-ng ?????????:

```
pip install django-cas-ng
```

????????????????????settings.py ????????? Django ?????????????????????:

```
# 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 = (
```

(?????????)

(XXXXXXXXXX)

```
"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")
```

```
signal django_cas_ng django_cas_ng signal
signal
django.apps.AppConfig.ready()
urls.py
```

```
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()
```

:

Django CAS NG

Django

Django Weblate Weblate user backend

:

LDAP CAS

AUTHENTICATION_BACKENDS INSTALLED_APPS

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (
    # Install authentication app here
)
```

Weblate

3.0 Weblate 3.0 Django Weblate

??????????

If you are not administrating the whole Weblate installation and just have access to manage certain projects (like on [Hosted Weblate](#)), your access control management options are limited to following settings. If you don't need any complex setup, those are sufficient for you.

??????????

??: ?????? Hosted Weblate ? Libre ?????????????????????????????????

You can limit user's access to individual projects by selecting a different *Access control* setting. Available options are:

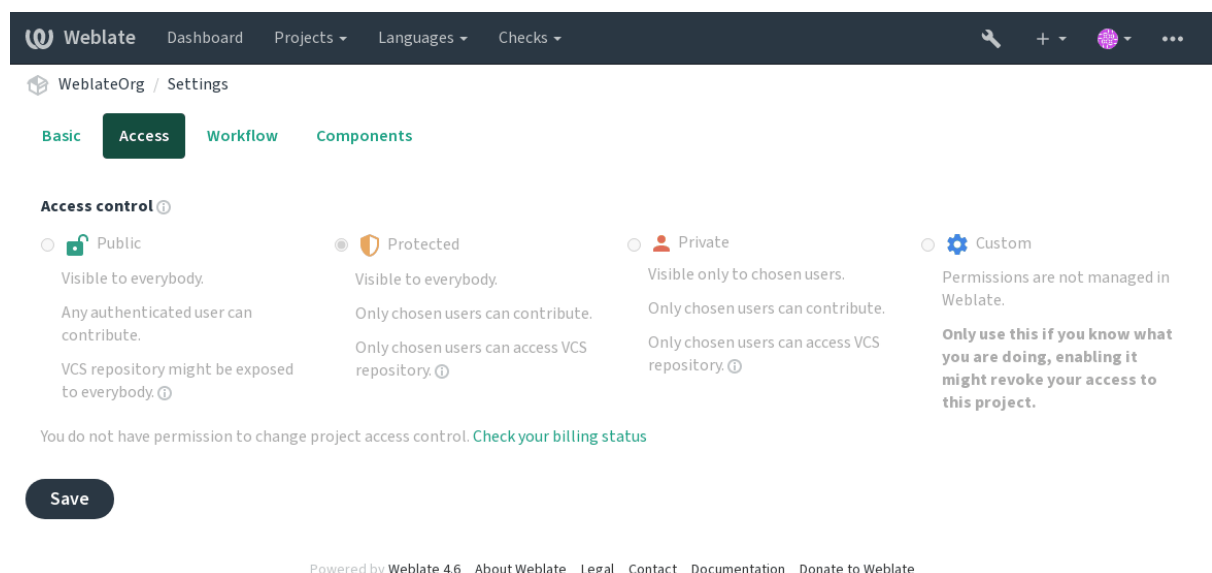
????????????????????????????????

????????????????????????????

????????????????????????????

User management features will be disabled; by default all users are forbidden to performed any actions on the project. You will have to set up all the permissions using [????? ??????](#).

Access control can be changed in the *Access* tab of the configuration (*Manage* ↓ *Settings*) of each respective project.



????????? `DEFAULT_ACCESS_CONTROL` ??????????

??: Even for *Private* projects, some info about your project will be exposed: statistics and language summary for the whole instance will include counts for all projects despite the access control setting. Your project name and other information can't be revealed through this.

??: The actual set of permissions available for users by default in *Public*, *Protected*, and *Private* projects can be redefined by Weblate instance administrator using *custom settings*.

??: [?????](#) ??????????????????Weblate ?????????????????????????????????? special groups ??????????Weblate
??

??:
[????????](#)

[illegible]

For *Public*, *Protected* and *Private* projects:

[illegible]

QUESTION

For *Protected* and *Private* projects only:

Webplate

DashboardProjectsLanguagesChecks

WebplateOrg / Access control

Users

API access

Users

Username	Full name	E-mail	Last login	Administration	Billing	Glossary	Languages	Memory	Screenshots	Sources	Translate	VCS
testuser	Webplate Test	webplate@example.org	25 seconds ago	☐	☐	☐	☐	☐	☐	☐	☒	☐

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 Webplate account name or e-mail address.

Add

Invite new user

E-mail

Username

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

Full name

Invite

These features are available on the *Access control* page, which can be accessed from the project's menu *Manage* ↓ *Users*.

Invitations

Also, besides adding an existing user to the project, it is possible to invite new ones. Any new user will be created immediately, but the account will remain inactive until signing in with a link in the invitation sent via an e-mail. It is not required to have any site-wide privileges in order to do so, access management permission on the project's scope (e.g. a membership in the *Administration* group) would be sufficient.

FAQ: If the invited user missed the validity of the invitation, they can set their password using invited e-mail address in the password reset form as the account is created already.

3.11 FAQ: How to manage invitations?

The same kind of invitations are available site-wide from the *management interface* on the *Users* tab.

Managing permissions

Weblate **manage users** *manage users* **manage users**

By default this prevents Weblate from granting access provided by *Users* and *Viewers default groups* due to these groups' own configuration. This doesn't prevent you from granting permissions to those projects site-wide by altering default groups, creating a new one, or creating additional custom settings for individual component as described in *FAQ: How to manage permissions* below.

One of the main benefits of managing permissions through the Weblate user interface is that you can delegate it to other users without giving them the superuser privilege. In order to do so, add them to the *Administration* group of the project.

FAQ: How to manage permissions?

FAQ: **Hosted Weblate** **Libre** **Hosted Weblate**

Managing permissions through the Django admin interface

The most powerful features of the Weblate's access control system for now are available only through the *Django admin interface*. You can use it to manage permissions of any project. You don't necessarily have to switch it to *Custom access control* to utilize it. However you must have superuser privileges in order to use it.

If you are not interested in details of implementation, and just want to create a simple-enough configuration based on the defaults, or don't have a site-wide access to the whole Weblate installation (like on *Hosted Weblate*), please refer to the *FAQ: How to manage permissions* section.

FAQ: How to manage permissions?

This section contains an overview of some common configurations you may be interested in.

Managing permissions for a whole instance

To manage permissions for a whole instance at once, add users to appropriate *default groups*:

Users (this is done by default by the *automatic group assignment*).

Reviewers (if you are using *review workflow* with dedicated reviewers).

Managers (if you want to delegate most of the management operations to somebody else).

You should keep all projects configured as *Public* (see *FAQ: How to manage permissions*), otherwise the site-wide permissions provided by membership in the *Users* and *Reviewers* groups won't have any effect.

You may also grant some additional permissions of your choice to the default groups. For example, you may want to give a permission to manage screenshots to all the *Users*.

You can define some new custom groups as well. If you want to keep managing your permissions site-wide for these groups, choose an appropriate value for the *Project selection* (e.g. *All projects* or *All public projects*).

????????????????

[illegible]

QUESTION

[illegible]

□□□□□□□

The scope of the permission assigned by the roles in the groups are applied by the following rules:

If the group specifies any *Component list*, all the permissions given to members of that group are granted for all the components in the component lists attached to the group, and an access with no additional permissions is granted for all the projects these components are in. *Components* and *Projects* are ignored.

If the group specifies any *Components*, all the permissions given to the members of that group are granted for all the components attached to the group, and an access with no additional permissions is granted for all the projects these components are in. *Projects* are ignored.

Otherwise, if the group specifies any *Projects*, either by directly listing them or by having *Projects selection* set to a value like *All public projects*, all those permissions are applied to all the projects, which effectively grants the same permissions to access all projects *unrestricted components*.

The restrictions imposed by a group's *Languages* are applied separately, when it's verified if a user has an access to perform certain actions. Namely, it's applied only to actions directly related to the translation process itself like reviewing, saving translations, adding suggestions, etc.

[illegible]

?

Let's say there is a project `foo` with the components: `foo/bar` and `foo/baz` and the following group:

Table4Group *Spanish Admin-Reviewers*

```

[?] [?] [?]      [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] foo/bar
[?] [?]      [?] [?] [?] [?]

```

Members of that group will have following permissions (assuming the default role settings):

General (browsing) access to the whole project `foo` including both components in it: `foo/bar` and `foo/baz`.

Review strings in `foo/bar` Spanish translation (not elsewhere).

Manage VCS for the whole `foo/bar` repository e.g. commit pending changes made by translators for all languages.

????????????????

Django admin interface

The most common use-case for the feature is to assign all new users to some default group. In order to do so, you will probably want to keep the default value (`^.*$`) in the regular expression field. Another use-case for this option might be to give some additional privileges to employees of your company by default. Assuming all of them use corporate e-mail addresses on your domain, this can be accomplished with an expression like `^.*@mycompany.com`.

[illegible]

FAQ: As for now, there is no way to bulk-add already existing users to some group via the user interface. For that, you may resort to using the [REST API](#).

????????????????

[illegible]

?????? [Administration, Billing]

[??????](#)
[\[Administration, Edit source, Power user, Review strings, Translate\]](#)

???????? [Administration]

???????? [Administration, Manage glossary, Power user]

???????? [Administration, Manage glossary, Power user]

[??????](#)
[\[Administration, Edit source, Power user, Review strings, Translate\]](#)

???????? [Administration, Manage translation memory]

???????????? [Administration]

🔍🔍🔍🔍🔍🔍🔍🔍🔍🔍 [Administration, Manage screenshots]

???????? [Administration, Manage screenshots]

???????? [Administration]

⌘⌘⌘⌘⌘⌘⌘⌘⌘⌘ [Administration, Edit source, Power user, Review strings, Translate]

?????? [Administration, Review strings]

 [Administration, Edit source, Power user]

⛔⛔⛔⛔⛔ [Administration, Edit source, Power user, Review strings, Translate]

⚙️ [Administration, Edit source, Power user, Review strings, Translate]

⌘⌘⌘⌘⌘⌘⌘ [Administration, Manage languages]

???????????? [Administration, Manage languages]

[Administration, Edit source, Power user, Review strings, Translate]

???????????? [Administration, Access repository, Power user, Manage repository]

[Administration, Manage repository]
[Administration, Manage repository]
upstream ##### [Administration, Access repository, Power user, Manage repository]
[Administration, Manage repository]

#####

?: #####
Weblate #####

#####

setupgroups #####

#####: *ANONYMOUS_USER_NAME*)

#####: *Add suggestion* *Access repository*

#####*automatic group assignment* #####
#####: none

#####*automatic group assignment* #####
#####: *Power user*
#####: #####
#####: *Review strings*

#####: *Administration*

?: ##### Weblate #####

Configuration

If you want to use your Weblate installation in a less public manner, i.e. allow new users on an invitational basis only, it can be done by configuring Weblate in such a way that only known users have an access to it. In order to do so, you can set `REGISTRATION_OPEN` to `False` to prevent registrations of any new users, and set `REQUIRE_LOGIN` to `/.*` to require logging-in to access all the site pages. This is basically the way to lock your Weblate installation.

Tip: You can use built-in *invitations* to add new users.

Translation organization

Weblate organizes translatable VCS content of project/components into a tree-like structure.

The bottom level object is *Project configuration*, which should hold all translations belonging together (for example translation of an application in several versions and/or accompanying documentation).

On the level above, *Component configuration*, which is actually the component to translate, you define the VCS repository to use, and the mask of files to translate.

Above *Component configuration* there are individual translations, handled automatically by Weblate as translation files (which match *File mask* defined in *Component configuration*) appear in the VCS repository.

Weblate supports a wide range of translation formats (both bilingual and monolingual ones) supported by Translate Toolkit, see [Translations formats](#).

Tip: You can share cloned VCS repositories using *Weblate URL*. Using this feature is highly recommended when you have many components sharing the same VCS. It improves performance and decreases required disk space.

Adding translation projects and components

3.2 Django: An interface for adding projects and components is included, and you no longer have to use *Django*.

3.4: The process of adding components is now multi staged, with automated discovery of most parameters.

Based on your permissions, new translation projects and components can be created. It is always permitted for users with the *Add new projects* permission, and if your instance uses billing (e.g. like <https://hosted.weblate.org/> see [Billing](#)), you can also create those based on your plans allowance from the user account that manages billing.

You can view your current billing plan on a separate page:

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	
Languages limit	Used 0	
Last invoice	2021-04-17 - 2021-04-19	
Projects limit	Used 0 of 1	
Projects	No projects currently assigned! <button>Add new translation project</button>	
Terminate billing plan		

Invoices		
Invoice period	Invoice amount	Download invoice
04/17/2021 - 04/19/2021	19.0 EUR	Not available

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

The project creation can be initiated from there, or using the menu in the navigation bar, filling in basic info about the translation project to complete addition of it:

Dashboard
Projects ▾
Languages ▾
Checks ▾

Create project

Add new 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.

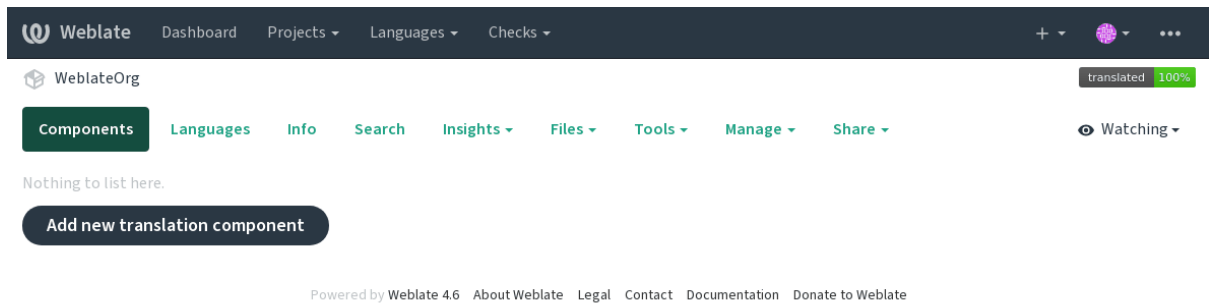
Billing ⓘ

Weblate Test (Basic plan)

Save

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

After creating the project, you are taken directly to the project page:



Creating a new translation component can be initiated via a single click there. The process of creating a component is multi-staged and automatically detects most translation parameters. There are several approaches to creating component:

- Creates component from remote version control repository.

- Creates additional component to existing one by choosing different files.

- Creates additional component to existing one, just for different branch.

- Upload translation files to Weblate in case you do not have version control or do not want to integrate it with Weblate. You can later update the content using the web interface or [Weblate's REST API](#).

- Upload single document and translate that.

- Create blank translation project and add strings manually.

Once you have existing translation components, you can also easily add new ones for additional files or branches using same repository.

First you need to fill in name and repository location:

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 ⓘ

Language names

Display name

URL slug ⓘ

language-names

Name used in URLs and filenames.

☐ Use as a glossary

Project ⓘ

WeblateOrg

Source language ⓘ

English

Language used for source strings in all components

Version control system ⓘ

Git

Version control system to use to access your repository with translations.

Source code repository ⓘ

https://github.com/WeblateOrg/demo.git

URL of a repository, use weblate://project/component for sharing with other component.

Repository branch ⓘ

Repository branch to translate

Continue

Powered by Weblate 4.6

About Weblate

Legal

Contact

Documentation

Donate to Weblate

On the next page, you are presented with a list of discovered translatable resources:

Weblate

Dashboard

Projects

Languages

Checks

Create component

Add new translation component ⓘ

Choose translation files to import ⓘ

☐ Specify configuration manually

☐ File format Android String Resource , Filemask app/src/main/res/values-*/strings.xml

☐ File format gettext PO file , Filemask weblate/langdata/locale/*/LC_MESSAGES/django.po

☐ File format gettext PO file , Filemask weblate/locale/*/LC_MESSAGES/django.po

☐ File format gettext PO file , Filemask weblate/locale/*/LC_MESSAGES/djangojs.po

Continue

Powered by Weblate 4.6

About Weblate

Legal

Contact

Documentation

Donate to Weblate

As a last step, you review the translation component info and fill in optional details:

193

Webplate

Dashboard

Projects ▾

Languages ▾

Checks ▾

+

▾

...

Create component

Add new translation component

Project ⓘ

WebplateOrg

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/WebplateOrg/demo.git

URL of a repository, use weplate://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/WebplateOrg/demo/blob/{{branch}}/{{filename}}#L{{line}}

Link to repository browser, use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders.

File format ⓘ

gettext PO file

Filemask ⓘ

weplate/langdata/locale/*/LC_MESSAGES/django.po

Path of files to translate relative to repository root, use * instead of language code, for example: po/* or locale/*/LC_MESSAGES/django.po.

Monolingual base language file ⓘ

Filename of translation base file, containing all strings and their source; it is recommended for monolingual translation formats.

☒ Edit base file

Whether users will be able to edit the base file for monolingual translations.

Intermediate language file ⓘ

Filename of intermediate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.

Template for new translations ⓘ

weplate/langdata/locale/django.pot

Filename of file used for creating new translations. For gettext choose .pot file.

Translation license ⓘ

GNU General Public License v3.0 or later

Adding new translation ⓘ

Create new language file

How to handle requests for creating new translations.

Language code style ⓘ

Default based on the file format

Customize language code used to generate the filename for translations created by Webplate.

Language filter ⓘ

^(cs|he|hu)\$

Regular expression used to filter translation files when scanning for filemask.

Source language ⓘ

English

Language used for source strings in all components

☐ Use as a glossary

You will be able to edit more options in the component settings after creating it.

Save

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

194

Django **Project configuration** **Component configuration**

[illegible]

/devel/integration

□ □ □ □ □ □ □

URL

?????? Web ???

This is a required parameter unless turned off by `WEBSITE_REQUIRED`.

URL to more site with more detailed instructions for translators.

Whether Weblate should manage the Language-Team header (this is a *GNU gettext* only feature right now).

Whether to use shared translation memory, see [\[10\]](#) for more details.

123456789101112

Whether to contribute to shared translation memory, see [???????](#) for more details.

?????? *DEFAULT_SHARED_TM* ??????????

Configure per project access control, see [??????????????](#) for more details.

Default value can be changed by `DEFAULT_ACCESS_CONTROL`.

??????

Enable review workflow for translations, see [??????](#).

??????????

Enable review workflow for source strings, see [??????](#).

??:

report-source? [??????](#)

??????????

Whether unauthenticated [??????](#) are to be used for this repository.

??:

[??????????](#)? [????????????????????](#)? *Bilingual and monolingual formats*? [????????](#)

??????

Define language codes mapping when importing translations into Weblate. Use this when language codes are inconsistent in your repositories and you want to get a consistent view in Weblate or in case you want to use non-standard naming of your translation files.

The typical use case might be mapping American English to English: `en_US:en`

Multiple mappings to be separated by comma: `en_GB:en,en_US:en`

Using non standard code: `ia_FOO:ia`

???: The language codes are mapped when matching the translation files and the matches are case sensitive, so make sure you use the source language codes in same form as used in the filenames.

??:

[??????????](#)

Component configuration

A component is a grouping of something for translation. You enter a VCS repository location and file mask for which files you want translated, and Weblate automatically fetches from this VCS, and finds all matching translatable files.

??:

/devel/integration

You can find some examples of typical configurations in the [????????????](#).

???: It is recommended to keep translation components to a reasonable size - split the translation by anything that makes sense in your case (individual apps or addons, book chapters or websites).

Weblate easily handles translations with 10000s of strings, but it is harder to split work and coordinate among translators with such large translation components.

Should the language definition for a translation be missing, an empty definition is created and named as "cs_CZ (generated)". You should adjust the definition and report this back to the Weblate authors, so that the missing languages can be included in next release.

The component contains all important parameters for working with the VCS, and for getting translations out of it:

????????

Verbose component name, used to display the component name.

????????????

Component name suitable for URLs.

Component project

Project configuration where the component belongs.

????????????

VCS to use, see [????????????](#) for details.

??:

Pushing changes from Weblate

????????????

VCS repository used to pull changes.

??:

See [??????????](#) for more details on specifying URLs.

???: This can either be a real VCS URL or `weblate://project/component` indicating that the repository should be shared with another component. See [Weblate ??? URL](#) for more details.

???????? URL

Repository URL used for pushing. This setting is used only for *Git* and *Mercurial* and push support is turned off for these when this is empty.

??:

See [??????????](#) for more details on how to specify a repository URL and *Pushing changes from Weblate* for more details on pushing changes from Weblate.

????? ??????

URL of repository browser used to display source files (location of used messages). When empty, no such links will be generated. You can use [????????????????](#).

For example on GitHub, use something like: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

In case your paths are relative to different folder, you might want to strip leading directory by `parentdir` filter (see [????????????](#)): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

WEBLATE_EXPORT_URL URL

URL where changes made by Weblate are exported. This is important when [Web `weblate`](#) is not used, or when there is a need to manually merge changes. You can use [Git exporter](#) to automate this for Git repositories.

WEBLATE_VCS VCS

Which branch to checkout from the VCS, and where to look for translations.

WEBLATE_PUSH push

Branch for pushing changes, leave empty to use [weblate](#).

WE: This is currently only supported for Git, GitLab and GitHub, it is ignored for other VCS integrations.

WE:

Pushing changes from Weblate

File mask

Mask of files to translate, including path. It should include one "*" replacing language code (see [weblate](#) for info on how this is processed). In case your repository contains more than one translation file (e.g. more gettext domains), you need to create a component for each of them.

WE: `po/*.*.po locale/*/LC_MESSAGES/django.po`

In case your filename contains special characters such as `[`, `]`, these need to be escaped as `[ ]` or `[ ]`.

WE:

Bilingual and monolingual formats **WE:** *What does mean "There are more files for the single language (en)"?*

WE_TRANSLATE_BASE_FILE

Base file containing string definitions for [weblate](#).

WE:

Bilingual and monolingual formats **WE:** *What does mean "There are more files for the single language (en)"?*

WE_ALLOW_EDITING

Whether to allow editing the base file for [weblate](#).

WE_TRANSLATE_INTERMEDIATE_FILE

Intermediate language file for [weblate](#). In most cases this is a translation file provided by developers and is used when creating actual source strings.

When set, the source strings are based on this file, but all other languages are based on [weblate](#). In case the string is not translated into the source language, translating to other languages is prohibited. This provides [weblate](#).

WE:

Bilingual and monolingual formats **WE:** *What does mean "There are more files for the single language (en)"?*

`POFILE`

Base file used to generate new translations, e.g. `.pot` file with `gettext`.

POFILE: In many monolingual formats Weblate starts with blank file by default. Use this in case you want to have all strings present with empty value when creating new translation.

POFILE:

adding-translation?Bilingual and monolingual formats?What does mean "There are more files for the single language (en)"?

`POFILE`

Translation file format, see also `POFILE`.

`REPORTING_EMAIL`

Email address used for reporting upstream bugs. This address will also receive notification about any source string comments made in Weblate.

`PROPAGATION`

You can turn off propagation of translations to this component from other components within same project. This really depends on what you are translating, sometimes it's desirable to have make use of a translation more than once. It's usually a good idea to turn this off for monolingual translations, unless you are using the same IDs across the whole project.

Default value can be changed by `DEFAULT_TRANSLATION_PROPAGATION`.

`REVIEW`

Whether translation suggestions are accepted for this component.

`REVIEW`

Turns on vote casting for suggestions, see `REVIEW`.

`REVIEW`

Automatically accept voted suggestions, see `REVIEW`.

`REVIEW`

`REVIEW` Weblate `REVIEW`: `REVIEW`

`REVIEW`

List of checks which can not be ignored, see `REVIEW`.

REVIEW: Enforcing the check does not automatically enable it, you still should enabled it using `REVIEW` in `REVIEW` or *Additional info on source strings*.

??????

License of the translation (does not need to be the same as the source code license).

??????

????????????????????????????????????

??????

How to handle requests for creation of new languages. Available options:

User can select desired language and the project maintainers will receive a notification about this. It is up to them to add (or not) the language to the repository.

User is presented a link to page which describes process of starting new translations. Use this in case more formal process is desired (for example forming a team of people before starting actual translation).

User can select language and Weblate automatically creates the file for it and translation can begin.

There will be no option for user to start new translation.

???: The project admins can add new translations even if it is disabled here when it is possible (either [????????](#) or the file format supports starting from an empty file).

??:

adding-translation [????????](#)

??????

????? 4.5 ???.

Configures whether users in Weblate will be allowed to add new strings and remove existing ones. Adjust this to match your localization workflow - how the new strings are supposed to be introduced.

For bilingual formats, the strings are typically extracted from the source code (for example by using **xgettext**) and adding new strings in Weblate should be disabled (they would be discarded next time you update the translation files). In Weblate you can manage strings for every translation and it does not enforce the strings in all translations to be consistent.

For monolingual formats, the strings are managed only on source language and are automatically added or removed in the translations. The strings appear in the translation files once they are translated.

??:

Bilingual and monolingual formats [adding-new-strings](#) `POST` `/api/translations/(string:project)/(string:component)/(string:language)/units/`

????? ?????

Weblate [????????????????????????????????](#)

??:

[????????](#) [??????](#) [????????](#)

merge_style

You can configure how updates from the upstream repository are handled. This might not be supported for some VCSs. See [Merge or rebase](#) for more details.

Default value can be changed by `DEFAULT_MERGE_STYLE`.

Commit, add, delete, merge and addon messages

Message used when committing a translation, see [commit_message](#).

`DEFAULT_ADD_MESSAGE` `DEFAULT_ADDON_MESSAGE` `DEFAULT_COMMIT_MESSAGE` `DEFAULT_DELETE_MESSAGE`

push_upstream

Whether committed changes should be automatically pushed to the upstream repository. When enabled, the push is initiated once Weblate commits changes to its underlying repository (see [Lazy commits](#)). To actually enable pushing *Repository push URL* has to be configured as well.

commit_pending_hours

Sets how old (in hours) changes have to be before they are committed by background task or the `commit_pending` management command. All changes in a component are committed once there is at least one change older than this period.

Default value can be changed by `COMMIT_PENDING_HOURS`.

Warning: There are other situations where pending changes might be committed, see [Lazy commits](#).

lock_on_first_failure

Locks the component (and linked components, see [Weblate URL](#)) upon the first failed push or merge into its upstream repository, or pull from it. This avoids adding another conflicts, which would have to be resolved manually.

The component will be automatically unlocked once there are no repository errors left.

source_language

Language used for source strings. Change this if you are translating from something else than English.

Warning: In case you are translating bilingual files from English, but want to be able to do fixes in the English translation as well, choose *English (Developer)* as a source language to avoid conflict between the name of the source language and the existing translation.

For monolingual translations, you can use intermediate translation in this case, see [intermediate_translation](#).

filter_files

`Webblate` `Webblate`

Warning: You need to list language codes as they appear in the filename.

Some examples of filtering:

Filter description	<code>filter_files</code>
Selected languages only	<code>^(cs de es)\$</code>
Exclude languages	<code>^(?!(it fr)\$) .+\$</code>
Filter two letter codes only	<code>^[.]+\$</code>
Exclude non language files	<code>^(?!(blank)\$) .+\$</code>
Include all files (default)	<code>^[^.] +\$</code>

Translations

Translations: variants

Q: Weblate translations

A:

Does Weblate support other VCSes than Git and Mercurial?

FAQ

Translations

FAQ

Translations

Restricting access at a component, or component-list level takes over access permission to a component, regardless of project-level permissions. You will have to grant access to it explicitly. This can be done through granting access to a new user group and putting users in it, or using the default *custom* or *private* access control groups.

Translations `DEFAULT_RESTRICTED_COMPONENT`

Q: —

FAQ

Translations

Q:

FAQ

Translations 4.5

Translations

The glossary will be accessible in all projects defined by

It is recommended to enable on glossaries in order to allow adding new words to them.

Q:

FAQ

Display color for a glossary used when showing word matches.

????????????

Weblate ??? The Django Template Language ?????????????????????

????????:

???? ??????????: *Component configuration*

??????

????????

????????

????????????????

????????????????:

????

??

????????

????????????

????????

????????????

?? URL

????????

????????????????????????????????????

??????

????????????

????????????

????????????

????????????

????????????

????????????

????????????????????????????????

??

????????????????????????????????????:

??????

??????

??????parentdir ??????????????????????????????: {{filename|parentdir}}

????????????:

```
{{ component|title }}
```

?????:

```
{% if stats.translated_percent > 80 %}Well translated!{% endif %}
```

????????:

```
{% replace component "-" " " %}
```

????????:

```
{% replace component|capfirst "-" " " %}
```

????????:

```
Directory of a file: {{ filename|dirname }}
File without extension: {{ filename|stripext }}
File in parent dir: {{ filename|parentdir }}
It can be used multiple times: {{ filename|parentdir|parentdir }}
```

... Django

VCS Weblate

Weblate

Celery: *Celery*

— Weblate

CPU Celery

CHECK_LIST

gettext Android
import_project *import_json*

1

import_project *import_json*

Management commands

□□□□□□□□

Webplate ISO 639-1

□□□□□ □□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□

[illegible]

QUESTION

????????????????

????????????????

□□□□□□□□

```

???????????????? -- cs_CZ  ???? cs  ????

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX1 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX xx_XX
XXXXxx_XX (generated) XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX: XXXXXX XXXXXXXXXXXXXXXXXXXX:
Webplate XXX XXXXXXXXXXXXXXX Webplate XXXXXXXXXXXXXXX

```

[illegible]

??:

QUESTION

□ □ □ □ □ □ □

????????????????/languages/ **URL**????????

[illegible]

□ □ □ □ □ □ □ □ □

```
? 600 ???????? Weblate ??????????????????????????????????Weblate ????????????????????????????????
```

```
weblate      migrate ??????????????:    ????????????????   ?????????????????????Weblate
```

```
????????????????????????????????
```

```

##### UPDATE LANGUAGES #####setuptools#####Webplate
#####

```

??:

The language definitions are in the [weblate-language-data](#) repository.

????????????????

[illegible]

??:

QUESTION

□□□□□

[illegible]

?????

Webdate ISO 639-1 2 ISO 639-2 ISO 639-3 BCP 47

:

?

???

Webdate

???????

?

?????

?

???????

Gettext

:

GNU gettext utilities: Plural forms Language Plural Rules by the Unicode Consortium

?????????

2.18 2.18

Webdate

string resources : GNU Android gettext JSON files

Component configuration Webdate

?

?:

POSIX

gettext pt_BR

POSIX cs_CZ

Web pt-BR

BCP cs-CZ

Android pt-rBR

Java — BCP

?

: Webdate

:

?

Web ??????????

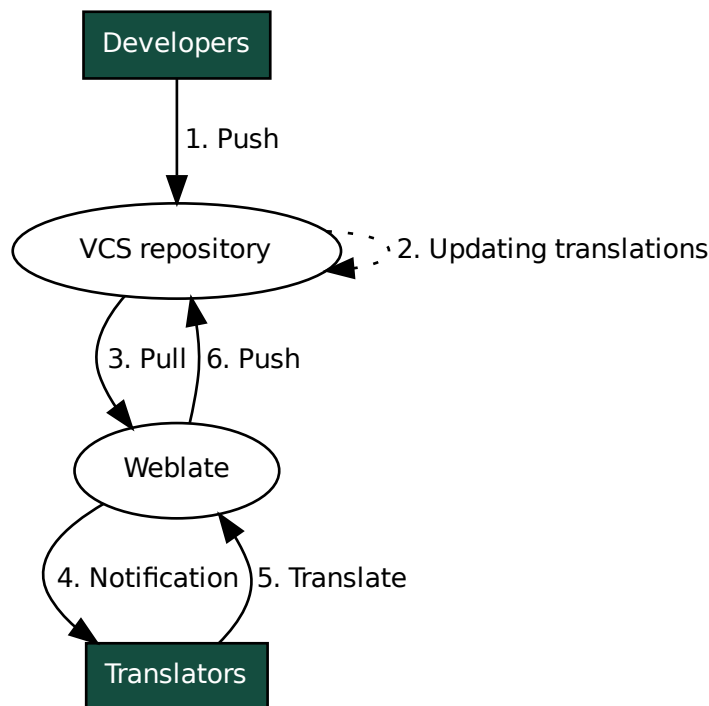
There is infrastructure in place so that your translation closely follows development. This way translators can work on translations the entire time, instead of working through huge amount of new text just prior to release.

??:

/devel/integration describes basic ways to integrate your development with Weblate.

This is the process:

1. Developers make changes and push them to the VCS repository.
2. Optionally the translation files are updated (this depends on the file format, see *Why does Weblate still show old translation strings when I've updated the template?*).
3. Weblate pulls changes from the VCS repository, see *Updating repositories*.
4. Once Weblate detects changes in translations, translators are notified based on their subscription settings.
5. Translators submit translations using the Weblate web interface, or upload offline changes.
6. Once the translators are finished, Weblate commits the changes to the local repository (see *Lazy commits*) and pushes them back if it has permissions to do so (see *Pushing changes from Weblate*).



Automatically receiving changes from GitHub

Weblate comes with native support for GitHub.

If you are using Hosted Weblate, the recommended approach is to install the [Weblate app](#), that way you will get the correct setup without having to set much up. It can also be used for pushing changes back.

To receive notifications on every push to a GitHub repository, add the Weblate Webhook in the repository settings (*Webhooks*) as shown on the image below:

The screenshot shows the GitHub repository settings page for 'WeblateOrg / hello'. The 'Settings' tab is selected, and the 'Webhooks' section is active. The 'Add webhook' form is displayed with the following fields and options:

- Payload URL:** `https://hosted.weblate.org/hooks/github/`
- Content type:** `application/x-www-form-urlencoded`
- Secret:** (empty text box)
- SSL verification:** A checkbox labeled 'By default, we verify SSL certificates when delivering payloads.' with a red button to 'Disable SSL verification'.
- Which events would you like to trigger this webhook?:**
 - ☒ Just the push event.
 - ☐ Send me everything.
 - ☐ Let me select individual events.
- Active:** A checked checkbox with the text 'We will deliver event details when this hook is triggered.'
- Add webhook:** A green button.

For the payload URL, append `/hooks/github/` to your Weblate URL, for example for the Hosted Weblate service, this is `https://hosted.weblate.org/hooks/github/`.

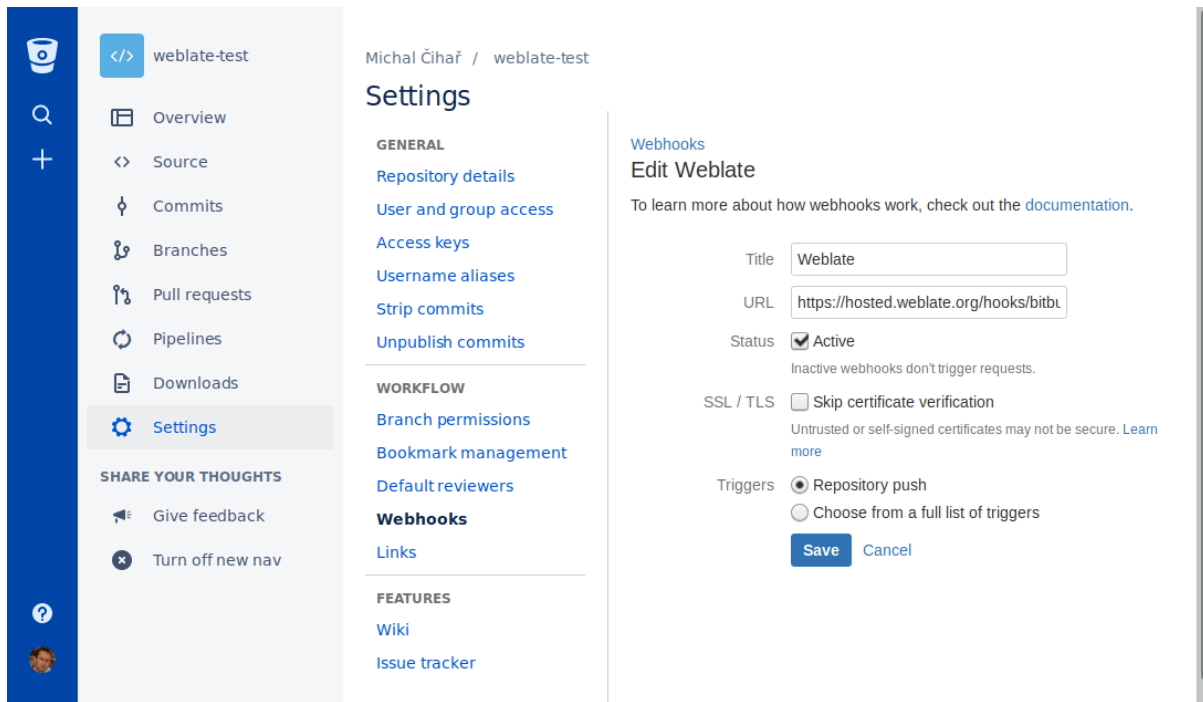
You can leave other values at default settings (Weblate can handle both content types and consumes just the *push* event).

❓:

`POST /hooks/github/Hosted Weblate`

Automatically receiving changes from Bitbucket

Weblate has support for Bitbucket webhooks, add a webhook which triggers upon repository push, with destination to `/hooks/bitbucket/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/bitbucket/`).



??:

`POST /hooks/bitbucket/Hosted Weblate`

Automatically receiving changes from GitLab

Weblate has support for GitLab hooks, add a project webhook with destination to `/hooks/gitlab/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/gitlab/`).

??:

`POST /hooks/gitlab/Hosted Weblate`

Pagure

3.3

Weblate Pagure Webhook URL WEB: `https://hosted.weblate.org/hooks/pagure/Project options` Activate Web-hooks

The screenshot shows the Fedora Pagure web interface for the 'nijel-test' project. The sidebar on the left contains navigation links: Project Settings, Project Details, Default Branch, Private Web Hook Key, API Keys, Project Options (selected), Public Notifications, Users & Groups, Deploy Keys, Hooks, Priorities, Roadmap, Close Status, Custom Issue Fields, Reports, Tags, Quick Replies, Regenerate Repos, Give Project, and Delete Project. The main content area is titled 'Project Options' and contains the following settings:

- ☐ Activate always merge
- ☐ Activate disable non fast-forward merges
- ☐ Activate Enforce signed-off commits in pull-request
- ☒ Activate fedmsg notifications
- ☒ Activate Issue tracker
- ☐ Activate Issue tracker read only
- ☐ Activate Issues default to private
- Activate Minimum score to merge pull-request:
- ☐ Activate notify on commit flag
- ☐ Activate notify on pull-request flag
- ☐ Activate Only assignee can merge pull-request
- ☐ Activate open metadata access to all
- ☐ Activate project documentation
- ☐ Activate pull request access only
- ☒ Activate pull requests
- ☒ Activate stomp notifications

Below the settings, there is a section for 'Activate Web-hooks' with a text input field containing 'https://hosted.weblate.org/hooks/pagure/'. There are buttons for 'Update' and 'Test web-hook'. At the bottom, there is a 'Learn more about' section with links to 'Flags', 'Tracker read-only', 'Pull-request access only', 'Roadmap on Issue page', and 'fedmsg notifications'.

???:
 POST /hooks/pagure/?Hosted Weblate ????????????????

Automatically receiving changes from Azure Repos

????? 3.8 ???.

Weblate has support for Azure Repos web hooks, add a webhook for *Code pushed* event with destination to /hooks/azure/ URL on your Weblate installation (for example <https://hosted.weblate.org/hooks/azure/>). This can be done in *Service hooks* under *Project settings*.

???:
 Web hooks in Azure DevOps manual ?POST /hooks/azure/?Hosted Weblate ????????????????

Automatically receiving changes from Gitea Repos

3.9

Weblate has support for Gitea webhooks, add a *Gitea Webhook* for *Push events* event with destination to `/hooks/gitea/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/gitea/`). This can be done in *Webhooks* under repository *Settings*.

🔗:

Webhooks in Gitea manual `POST /hooks/gitea/` Hosted Weblate

Automatically receiving changes from Gitee Repos

3.9

Weblate has support for Gitee webhooks, add a *WebHook* for *Push* event with destination to `/hooks/gitee/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/gitee/`). This can be done in *WebHooks* under repository *Management*.

🔗:

Webhooks in Gitee manual `POST /hooks/gitee/` Hosted Weblate

Automatically updating repositories nightly

Weblate automatically fetches remote repositories nightly to improve performance when merging changes later. You can optionally turn this into doing nightly merges as well, by enabling `AUTO_UPDATE`.

Pushing changes from Weblate

Each translation component can have a push URL set up (see `URL`), and in that case Weblate will be able to push change to the remote repository. Weblate can be also be configured to automatically push changes on every commit (this is default, see `Component configuration`). If you do not want changes to be pushed automatically, you can do that manually under *Repository maintenance* or using API via `wlc push`.

The push options differ based on the `used`, more details are found in that chapter.

Weblate `GitHub` `GitLab` `Pagure` `Gerrit`
`Component configuration` `GitHub` `GitLab` `Gerrit`
`Pagure`

Overall, following options are available with Git, GitHub and GitLab:

Desired setup	URL	push
No push	<code>Git</code>	<code>empty</code>
Push directly	<code>Git</code>	<code>SSH URL</code>
<code>push</code>	<code>Git</code>	<code>SSH URL</code>
GitHub pull request from fork	<code>GitHub</code>	<code>empty</code>
GitHub pull request from branch	<code>GitHub</code>	<code>SSH URL¹</code>
GitLab merge request from fork	<code>GitLab</code>	<code>empty</code>
GitLab merge request from branch	<code>GitLab</code>	<code>SSH URL^{p. 53, 1}</code>
<code>Pagure</code>	<code>Pagure</code>	<code>empty</code>
<code>Pagure</code>	<code>Pagure</code>	<code>SSH URL^{p. 53, 1}</code>

🔗: You can also enable automatic pushing of changes after Weblate commits, this can be done in `.`

🔗:

See `for setting up SSH keys`, and *Lazy commits* for info about when Weblate decides to commit changes.

Can be empty in case `supports pushing`.

Protected branches

If you are using Weblate on protected branch, you can configure it to use pull requests and perform actual review on the translations (what might be problematic for languages you do not know). An alternative approach is to waive this limitation for the Weblate push user.

For example on GitHub this can be done in the repository configuration:

☒ **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



Merge or rebase

By default, Weblate merges the upstream repository into its own. This is the safest way in case you also access the underlying repository by other means. In case you don't need this, you can enable rebasing of changes on upstream, which will produce a history with fewer merge commits.

??: Rebasing can cause you trouble in case of complicated merges, so carefully consider whether or not you want to enable them.

Interacting with others

Weblate makes it easy to interact with others using its API.

??:

Weblate's REST API

Consistency check

The `check` check fires whenever the strings are different. You can utilize this to review such differences manually and choose the right translation.

□□□□

Automatic translation based on different components can be way to synchronize the translations across components. You can either trigger it manually (see [\[1\]](#)) or make it run automatically on repository update using `addon` (see [\[2\]](#)).

□□□□□□□□

Component configuration

??????

URL ##### Component configuration #####

Libre
#####

□□□□□□□

[illegible]

Weblate
Dashboard
Projects
Languages
Checks

WeblateOrg / Language names

translated 95%

Contribution to this translation requires you to agree with a contributor agreement.

View contributor agreement

Translations

Info
Alerts
Search
Insights
Files
Tools
Manage
Share

Watching

Language	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
Czech						
Hebrew						
Hungarian	81%	4	5			
English						

Start new translation

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

????????????????????HTML ?????????????????

[illegible]

□□□□□

□□□□□□

Everyone can add suggestions by default, to be accepted by signed in users. Suggestion voting can be used to make use of a string when more than one signed-in user agrees, by setting up the *Component configuration* configuration with *Suggestion voting* to turn on voting, and *Autoaccept suggestions* to set a threshold for accepted suggestions (this includes a vote from the user making the suggestion if it is cast).

22: Once automatic acceptance is set up, normal users lose the privilege to directly save translations or accept suggestions. This can be overridden with the *Edit string when suggestions are enforced* [permission](#).

You can combine these with *access control* into one of the following setups:

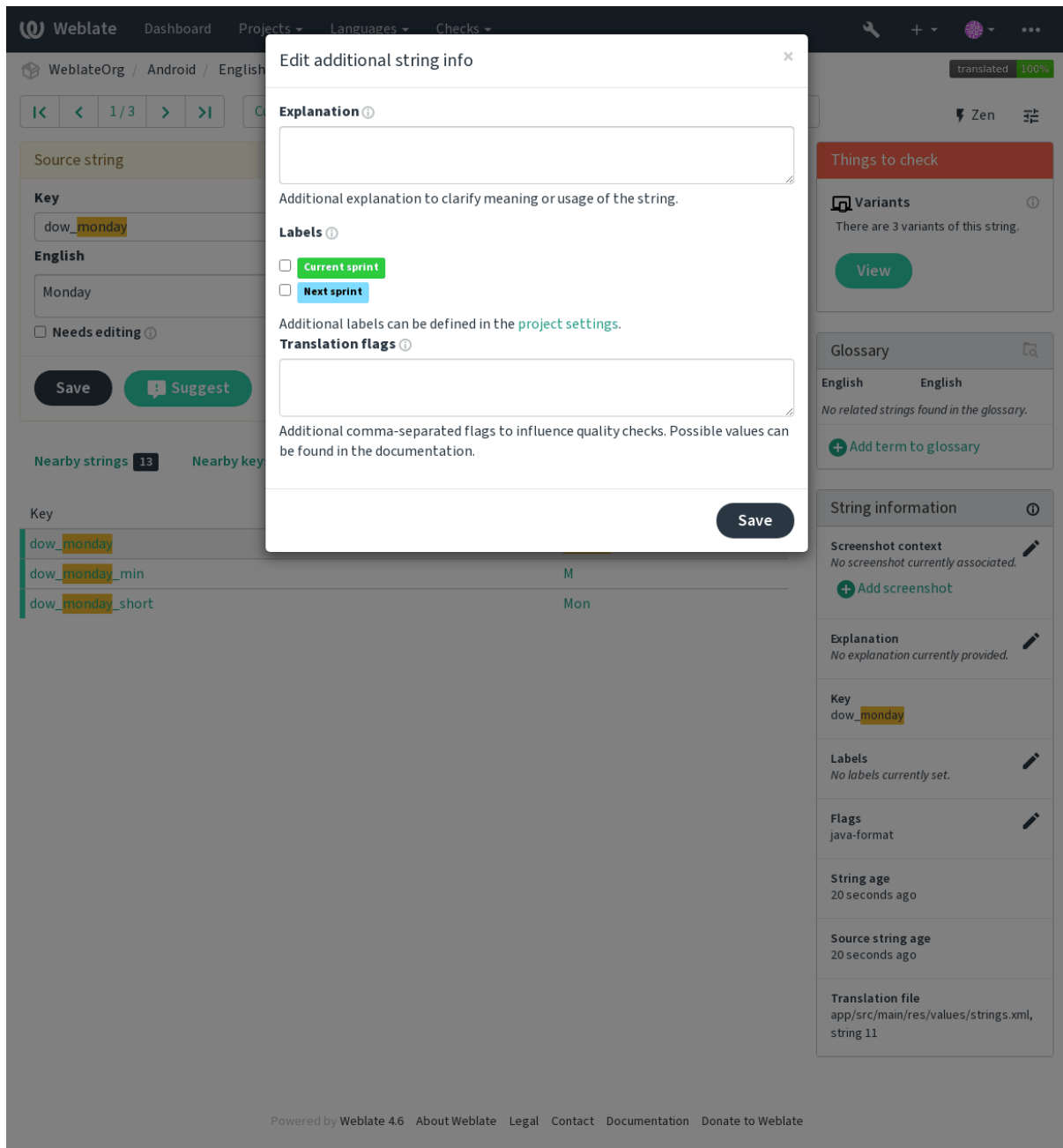
Users suggest and vote for suggestions and a limited group controls what is accepted. - Turn on voting. - Turn off automatic acceptance. - Don't let users save translations.

Users suggest and vote for suggestions with automatic acceptance once the defined number of them agree. - Turn on voting. - Set the desired number of votes for automatic acceptance.

Optional voting for suggestions. (Can optionally be used by users when they are unsure about a translation by making multiple suggestions.) - Only turn on voting.

Additional info on source strings

Enhance the translation process by adding additional info to the strings including explanations, string priorities, check flags and visual context. Some of that info may be extracted from the translation files and some may be added by editing the additional string info:



Access this directly from the translation interface by clicking the "Edit" icon next to *Screenshot context* or *Flags*.

WebateDashboardProjectsLanguagesChecks

WebateOrg / Django / Czech / Translate

translated96%

<<11 / 26>>

All strings

Position and priority

Zen

Translation

Explanation

Help text for automatic translation tool

English

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

Czech

Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.

Needs editing

169 / 1570 · 157

Save

Suggest

Skip

Nearby strings26

Comments

Automatic suggestions

Other languages4

History

Context	English	Czech
	Files	Soubory
	Automatic translation	Automatický překlad
	Add new translation string	Add new translation string
	Translation status	Stav překladu
	%(count)s word	%(count)s slovo
	Other components	Další součásti
	Translation file	Soubor s překladem
	Download	Stáhnout
	Browse all translation changes	Procházet všechny změny v překladu.
	Automatic translation takes existing translations in this project and applies them to the current component. It can be used to push translations to a different branch, to fix inconsistent translations or to translate a new component using translation memory.	Automatický překlad použije stávající překlady v projektu na tuto součást. Může být užitečný pro sloučení překladů z jiné větve, opravu nekonzistentních překladů nebo překlad nové součásti pomocí překladové paměti.
	Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.	Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.
	You can add new translation string here, it will automatically appear in all translations.	Zde můžete přidat nový řetězec k překladu, automaticky se objeví ve všech jazycích.
	The uploaded file will be merged with the current translation. In case you want to overwrite already translated strings, don't forget to enable it.	Nahráný soubor bude sloučen se stávajícími překlady. Pokud chcete přepsat již přeložené řetězce, nezapomeňte to povolit.
	The uploaded file will be merged with the current translation.	Nahráný soubor bude sloučen se stávajícími překlady.
	The fulltext search might not work properly as the fulltext index for this translation is not yet up to date.	Fulltextové vyhledávání nemusí fungovat správně, protože fulltextový index pro tento překlad ještě není plně zpracován.
	Review	Kontrola
	Review translations touched by other users.	Zkontrolovat překlady od ostatních uživatelů.
	Start review	Začít kontrolu
	Percent	Procenta
	Total	Celkem
	Failing check	Neúspěšných kontrol
	Last activity	Poslední aktivita
	Last change	Poslední změna
	Last author	Poslední autor
Question for a mathematics-based CAPTCHA, the %s is an arithmetic problem	What is %s?	Kolik to je?
	The string uses three dots (...) instead of an ellipsis character (...)	

Glossary

EnglishCzech

machinestrojovýweblateorg

translationpřekladweblateorg

projectprojektweblateorg

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

Help text for automatic translation tool

Labels

No labels currently set.

Flags

No flags currently set.

Source string location

weblate/templates/translation.html:212

String age

3 seconds ago

Source string age

3 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGES/django.po, string 11

Powered by Webate 4.6About WebateLegalContactDocumentationDonate to Webate

218

Strings prioritization

????? 2.0 ???.

String priority can be changed to offer higher priority strings for translation earlier by using the `priority` flag.

???: This can be used to order the flow of translation in a logical manner.

??:

????

??????

????? 2.4 ???.

????? 3.3 ???: Previously called *Quality checks flags*, it no longer configures only checks.

The default set of translation flags is determined by the translation *Component configuration* and the translation file. However, you might want to use it to customize this per source string.

??:

????

??

????? 4.1 ???: In previous versions this has been called *Extra context*.

Use the explanation to clarify scope or usage of the translation. You can use Markdown to include links and other markup.

Visual context for strings

????? 2.9 ???.

You can upload a screenshot showing a given source string in use within your program. This helps translators understand where it is used, and how it should be translated.

The uploaded screenshot is shown in the translation context sidebar:

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

translated 96%

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

[11 / 26](#)

[All strings](#)

[Position and priority](#)

Translation

Explanation

Help text for automatic translation tool

English

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

Czech

Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.

Needs editing

169/1570 · 157

Save

Suggest

Skip

Nearby strings 26

Comments

Automatic suggestions

Other languages 4

History

Translation memory

Search

Translation	Source	Origin	Similarity		
Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.	Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.	Weblate Translation Memory (Project: weblateorg/django) Weblate Translation Memory (Shared: weblateorg/django) Weblate (WeblateOrg/Django)	100 %	Copy	Copy and save

Glossary

English	Czech	
machine translation	strojový překlad	WeblateOrg
project	projekt	WeblateOrg

Add term to glossary

String information

Screenshot context

Add screenshot

Explanation

Help text for automatic translation tool

Labels

No labels currently set.

Flags

No flags currently set.

Source string location

weblate/templates/translation.html:212

String age

14 seconds ago

Source string age

14 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGES/django.po, string 11

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

In addition to *Additional info on source strings*, screenshots have a separate management interface under the *Tools* menu. Upload screenshots, assign them to source strings manually, or use optical character recognition to do so.

Once a screenshot is uploaded, this interface handles management and source string association:

Webate

Dashboard

Projects

Languages

Checks

+

WebateOrg / Django / Screenshots / Automatic translation

Screenshot has been uploaded, you can now assign it to source strings.

Assigned source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			
Screenshot is shown to add visual context for all listed source strings.			

Assign source strings

English	Location	Assigned screenshots	Actions
Loading results...			

Source string search

Search

Automatically recognize

Image

Source string

Hello, world!🇺🇸

One
Orangutan has %d banana.🇺🇸

Other
Orangutan has %d bananas.🇺🇸

Try Webate at <http://demo.webate.org/>!🇺🇸

Thank you for using Webate.

Screenshot is shown to add visual context for all listed source strings.

Edit screenshot

Screenshot name

Automatic translation

Image

Currently: screenshots/screenshot.png

Change:

Choose File

No file chosen

Upload JPEG or PNG images up to 2000x2000 pixels.

Save

Screenshot details

Created	now
Uploaded by	testuser
Language	English

Delete screenshot

Deleting screenshot will remove it from all associated source strings.

Delete

?????

????????????

```

AUTOFIX_LIST
;
foo bar:

```

```

#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#

from django.utils.translation import gettext_lazy as _

from weblate.trans.autofixes.base import AutoFix

class ReplaceFooWithBar(AutoFix):
    """Replace foo with bar."""

    name = _("Foobar")

    def fix_single_target(self, target, source, unit):
        if "foo" in target:
            return target.replace("foo", "bar"), True
        return target, False

```

```

AUTOFIX_LIST
Python
:

```

????????????

Weblate : Additional info on source strings Component configuration
:
:

```

placeholders:"special:value":"other value", regex:.*

```

```

:
reStructuredText 
Markdown
UNIX DOS \n \r\n
URL
HTML: HTML
Weblate:
100
N:

```

[illegible]

Font group			
Name	default-font		
Default font	Source Sans Pro Bold		
Japanese	language override	Droid Sans Fallback Regular	Remove
Korean	language override	Droid Sans Fallback Regular	Remove
Delete			

Add language override

Language

▼

Font

▼

Save

Edit font group

Font group name

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font

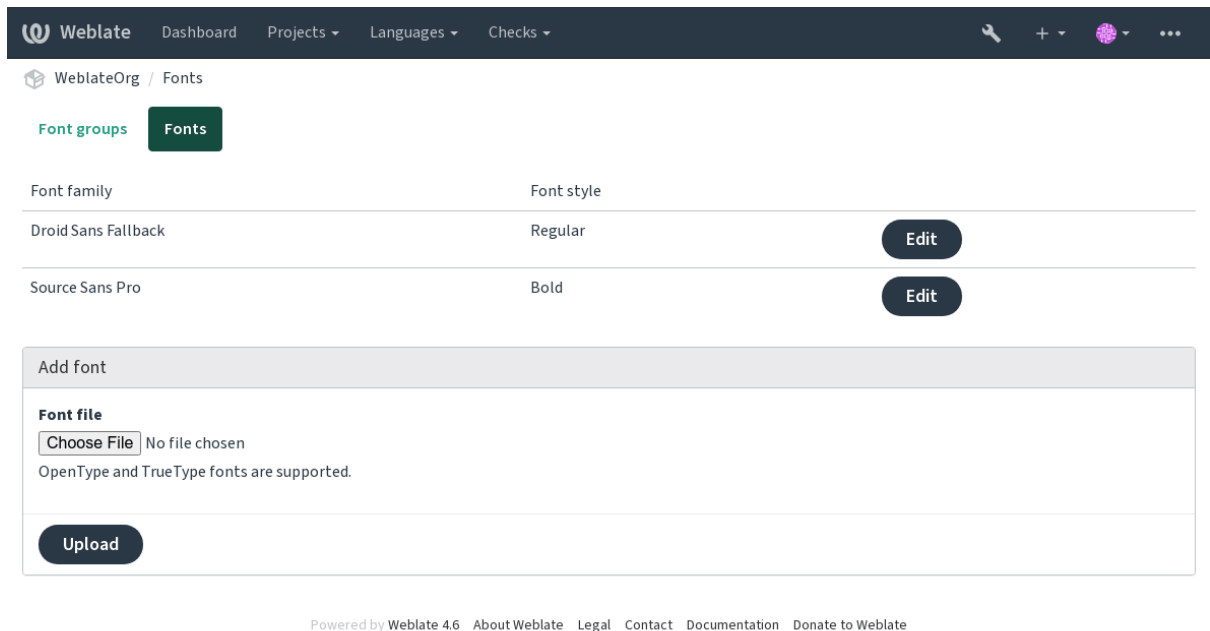
Default font is used unless per language override matches.

Save

[illegible]

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

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



:

:
CHECK_LIST

- 1.weblate.checks.Check
- 2.
- 3.check check_single

CHECK_LIST Python

foo

foo

```
#
# Copyright © 2012 – 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
```

(XXXXXXXXXX)

```
"""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
```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX 2 XXXXXXXXXXXXXXXXXXXXXXXXXXXX

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""Quality check example for Czech plurals."""

from django.utils.translation import gettext_lazy as _
from weblate.checks.base import TargetCheck

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
```

(XXXXXXXXXX)

DeepL

2022 2.20

DeepL is paid service providing good machine translation for a few languages. You need to purchase *DeepL API* subscription or you can use legacy *DeepL Pro (classic)* plan.

Turn on this service by adding `weblate.machinery.deepl.DeepLTranslation` to `MT_SERVICES` and set `MT_DEEPL_KEY`.

???: In case you have subscription for CAT tools, you are supposed to use "v1 API" instead of default "v2" used by Weblate (it is not really an API version in this case). You can toggle this by `MT_DEEPL_API_VERSION`.

??:

`MT_DEEPL_KEY` `MT_DEEPL_API_VERSION` [DeepL website](#) [DeepL pricing](#) [DeepL API documentation](#)

Glosbe

Free dictionary and translation memory for almost every living language.

The API is gratis to use, but subject to the used data source license. There is a limit of calls that may be done from one IP in a set period of time, to prevent abuse.

Turn on this service by adding `weblate.machinery.glosbe.GlosbeTranslation` to `MT_SERVICES`.

??:

[Glosbe website](#)

Google Translate

Google

This service uses the Google Translation API, and you need to obtain an API key and turn on billing in the Google API console.

To turn on this service, add `weblate.machinery.google.GoogleTranslation` to `MT_SERVICES` and set `MT_GOOGLE_KEY`.

??:

`MT_GOOGLE_KEY` [Google translate documentation](#)

Google Translate API V3 (Advanced)

Google Cloud

This service differs from the former one in how it authenticates. To enable service, add `weblate.machinery.googlev3.GoogleV3Translation` to `MT_SERVICES` and set

`MT_GOOGLE_CREDENTIALS`

`MT_GOOGLE_PROJECT`

If `location` fails, you may also need to specify `MT_GOOGLE_LOCATION`.

??:

`MT_GOOGLE_CREDENTIALS` `MT_GOOGLE_PROJECT` `MT_GOOGLE_LOCATION` [Google translate documentation](#)

Microsoft Cognitive Services Translator

2.10

Cognitive Services Azure portal

Weblate implements Translator API V3.

To enable this service, add `weblate.machinery.microsoft.MicrosoftCognitiveTranslation` to `MT_SERVICES` and set `MT_MICROSOFT_COGNITIVE_KEY`.

Translator Text API V2

The key you use with Translator API V2 can be used with API 3.

Translator Text API V3

Azure `MT_MICROSOFT_REGION` Azure `MT_MICROSOFT_REGION`

:

`MT_MICROSOFT_COGNITIVE_KEY` `MT_MICROSOFT_REGION` Cognitive Services - Text Translation API
Microsoft Azure Portal

Microsoft Terminology Service

2.19

The Microsoft Terminology Service API Web UI

`weblate.machinery.microsoftterminology.MicrosoftTerminologyService` `MT_SERVICES`

:

Microsoft Terminology Service API

ModernMT

4.2

Turn this service on by adding `weblate.machinery.modernmt.ModernMTTranslation` to `MT_SERVICES` and configure `MT_MODERNMT_KEY`.

:

ModernMT API, `MT_MODERNMT_KEY`, `MT_MODERNMT_URL`

MyMemory

Huge translation memory with machine translation.

Free, anonymous usage is currently limited to 100 requests/day, or to 1000 requests/day when you provide a contact e-mail address in `MT_MYMEMORY_EMAIL`. You can also ask them for more.

Turn on this service by adding `weblate.machinery.mymemory.MyMemoryTranslation` to `MT_SERVICES` and set `MT_MYMEMORY_EMAIL`.

:

`MT_MYMEMORY_EMAIL` `MT_MYMEMORY_USER` `MT_MYMEMORY_KEY` MyMemory website

NetEase Sight API machine translation

3.3

Netease

This service uses an API, and you need to obtain key and secret from NetEase.

Turn on this service by adding `weblate.machinery.youdao.NeteaseSightTranslation` to `MT_SERVICES` and set `MT_NETEASE_KEY` and `MT_NETEASE_SECRET`.

🔗:

`MT_NETEASE_KEY` `MT_NETEASE_SECRET` Netease Sight Translation Platform

tmserver

You can run your own translation memory server by using the one bundled with Translate-toolkit and let Weblate talk to it. You can also use it with an amaGama server, which is an enhanced version of tmserver.

1. First you will want to import some data to the translation memory:

2. Turn on this service by adding `weblate.machinery.tmserver.TMServerTranslation` to `MT_SERVICES`.

```
build_tmdb -d /var/lib/tm/db -s en -t cs locale/cs/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t de locale/de/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t fr locale/fr/LC_MESSAGES/django.po
```

3. Start tmserver to listen to your requests:

```
tmserver -d /var/lib/tm/db
```

4. Configure Weblate to talk to it:

```
MT_TMSERVER = "http://localhost:8888/tmserver/"
```

🔗:

`MT_TMSERVER` tmserver Installing amaGama Amagama Amagama Translation Memory

Yandex Translate

Yandex

This service uses a Translation API, and you need to obtain an API key from Yandex.

Turn on this service by adding `weblate.machinery.yandex.YandexTranslation` to `MT_SERVICES`, and set `MT_YANDEX_KEY`.

🔗:

`MT_YANDEX_KEY` Yandex Translate API Powered by Yandex.Translate

Youdao Zhiyun API machine translation

3.2

Youdao

This service uses an API, and you need to obtain an ID and an API key from Youdao.

Turn on this service by adding `weblate.machinery.youdao.YoudaoTranslation` to `MT_SERVICES` and set `MT_YOUDAO_ID` and `MT_YOUDAO_SECRET`.

🔗:

`MT_YOUDAO_ID` `MT_YOUDAO_SECRET` Youdao Zhiyun Natural Language Translation Service

Weblate

Weblate can be the source of machine translations as well. It is based on the Woosh fulltext engine, and provides both exact and inexact matches.

Turn on these services by adding `weblate.machinery.weblatetm.WeblateTranslation` to `MT_SERVICES`.

Weblate Translation Memory

2.20

The `weblate.memory.machine.WeblateMemory` can be used as a source for machine translation suggestions as well.

Turn on these services by adding `weblate.memory.machine.WeblateMemory` to the `MT_SERVICES`. This service is turned on by default.

SAP Translation Hub

SAP

You need to have a SAP account (and enabled the SAP Translation Hub in the SAP Cloud Platform) to use this service.

Turn on this service by adding `weblate.machinery.saptranslationhub.SAPTranslationHub` to `MT_SERVICES` and set the appropriate access to either sandbox or the productive API.

Tip: To access the Sandbox API, you need to set `MT_SAP_BASE_URL` and `MT_SAP_SANDBOX_APIKEY`.

To access the productive API, you need to set `MT_SAP_BASE_URL`, `MT_SAP_USERNAME` and `MT_SAP_PASSWORD`.

Tip:

`MT_SAP_BASE_URL``MT_SAP_SANDBOX_APIKEY``MT_SAP_USERNAME``MT_SAP_PASSWORD``MT_SAP_USE_MT`
SAP Translation Hub API

Custom machine translation

You can also implement your own machine translation services using a few lines of Python code. This example implements machine translation in a fixed list of languages using dictionary Python module:

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""Machine translation example."""

import dictionary

from weblate.machinery.base import MachineTranslation
```

(XXXXXXXXXXXX)

```
class SampleTranslation(MachineTranslation):
    """Sample machine translation interface."""

    name = "Sample"

    def download_languages(self):
        """Return list of languages your machine translation supports."""
        return {"cs"}

    def download_translations(
        self,
        source,
        language,
        text: str,
        unit,
        user,
        search: bool,
        threshold: int = 75,
    ):
        """Return tuple with translations."""
        for t in dictionary.translate(text):
            yield {"text": t, "quality": 100, "service": self.name, "source
↪": text}
```

You can list own class in `MT_SERVICES` and Weblate will start using that.

????

XXXXX 2.19 XXX.

XX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

??

↓

XXXXX

Installed addons ⓘ	
There are no addons currently installed.	
Available addons ⓘ	
 Automatic translation ⓘ Automatically translates strings using machine translation or other components. Install	
 Add missing languages ⓘ Ensures a consistent set of languages is used for all components within a project. project wide Install	
 Component discovery ⓘ Automatically adds or removes project components based on file changes in the version control system. repository wide Install	
 Bulk edit ⓘ Bulk edit flags, labels, or states of strings. Install	
 Statistics generator ⓘ Generates a file containing detailed info about the translation status. Install	
 Pseudolocale generation ⓘ Generates a translation by adding prefix and suffix to source strings automatically. Install	
 Contributors in comment ⓘ Updates the comment part of the PO file header to include contributor names and years of contributions. Install	
 Customize gettext output ⓘ Allows customization of gettext output behavior, for example line wrapping. Install	
 Generate MO files ⓘ Automatically generates a MO file for every changed PO file. Install	
 Update PO files to match POT (msgmerge) ⓘ Updates all PO files (as configured by "Filemask") to match the POT file (as configured by "Template for new translations") using msgmerge. Install	
 Squash Git commits ⓘ Squash Git commits prior to pushing changes. repository wide Install	
 Stale comment removal ⓘ Set a timeframe for removal of comments. project wide Install	
 Stale suggestion removal ⓘ Set a timeframe for removal of suggestions. project wide Install	
Some addons will ask for additional configuration during installation.	

XXXXXXXXXX

XXXXX

XXXXXX 3.9 XXX.

XX

XXXXX:

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Once in a month for every component, this can be configured using *BACKGROUND_TASKS*.

XX:

XXXXXX Keeping translations same across components

JavaScript XXXX CDN

XXXXXX 4.2 XXX.

JavaScript XXX HTML XXX

XXX HTML XXXXXXXXXXXXXXXJavaScript XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

HTML XXX URL XXXXXXXXXXXXXXXXXXXXXXXweblate-cdn XXXXXXXXXXXXXXX

XX:

cdn-addon-configweblate-cdncdn-addon-extractcdn-addon-html

XXXXXXXXXX

XXXXXX 4.4 XXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XX: XXX

XX:

Does Weblate update translation files besides translations?

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XX

XX:

Does Weblate update translation files besides translations?

XXXXXXXXXXXXXXXXXXXX

XX

XXXXXXXXXXXX24 XXXXX 1 XXXXXXXXXXXXXXX Weblate XXXXXXXXXXXXXXX

XX

XXX: XXXXXXXXXXXXXXX XXXXX XXXXXXXXXXXXXXX

????????

??

VCS ?????????????????? *import_project* ??????????????????1 ?? VCS
????????????????????

??
????????????????

?? ???:

W Weblate

DashboardProjectsLanguagesChecks

⚙️ + 🌐 ⋮

WebateOrg / Language names / Addons / Component discovery

Configure addon

Please review and confirm the matched components.

Component	Matched files
Following components would be created	
Djangojs	weblate/locale/cs/LC_MESSAGES/djangojs.po (cs) weblate/locale/hu/LC_MESSAGES/djangojs.po (hu) weblate/locale/he/LC_MESSAGES/djangojs.po (he)
Django	weblate/locale/cs/LC_MESSAGES/django.po (cs) weblate/locale/hu/LC_MESSAGES/django.po (hu) weblate/locale/he/LC_MESSAGES/django.po (he)

☐ I confirm the above matches look correct

Regular expression to match translation files against

weblate/locale/(?P<language>[^\.]*)(?P<component>[^\.]*).po

File format

gettext PO file

Customize the component name

{{ component|title }}

Define the monolingual base filename

Leave empty for bilingual translation files.

Define the base file for new translations

weblate/locale/{{ component }}.pot

Filename of file used for creating new translations. For gettext choose .pot file.

Language filter

^(cs|he|hu)\$

Regular expression to filter translation files against when scanning for filemask.

☒ Clone addons from the main component to the newly created ones

☐ Remove components for inexistant files

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

Regular expression	Example matched files	Description
(?P<language>[^\.]*)(?P<component>[^\.]*).po	cs/application.po cs/website.po de/application.po de/website.po	One folder per language containing translation files for components.
locale/(?P<language>[^\.]*)/LC_MESSAGES/(?P<component>[^\.]*).po	locale/cs/LC_MESSAGES/application.po locale/cs/LC_MESSAGES/website.po locale/de/LC_MESSAGES/application.po locale/de/LC_MESSAGES/website.po	Usual structure for storing gettext PO files.
src/locale/(?P<component>[^\.]*).(?P<language>[^\.]*).po	src/locale/application.cs.po src/locale/website.cs.po src/locale/application.de.po src/locale/website.de.po	Using both component and language name within filename.
locale/(?P<language>[^\.]*)(?P<component>[^\.]*)(?P=language).po	locale/cs/application/cs.po locale/cs/website/cs.po locale/de/application/de.po locale/de/website/de.po	Using language in both path and filename.
res/values-(?P<language>[^\.]*)/strings-(?P<component>[^\.]*).xml	res/values-cs/strings-about.xml res/values-cs/strings-help.xml res/values-de/strings-about.xml res/values-de/strings-help.xml	Android resource strings, split into several files.

You can use Django template markup in both component name and the monolingual base filename, for example:

{{ component }}

Component filename match

{{ component|title }}

Component filename with upper case first letter

Save

Tip: Component discovery addon uses [Weblate `weblate://` URL](#). It's a convenient way to share VCS setup between multiple components. Linked components use the local repository of the main component set up by filling `weblate: /project/main-component` into the `weblate: /project/main-component` field (in *Manage* ↓ *Settings* ↓ *Version control system*) of each respective component. This saves time with configuration and system resources too.

Tip:

[Weblate `weblate://` URL](#)

Tip:

Tip: 3.11 **Tip:**

[Weblate `weblate://` URL](#)

NOT has:label [Weblate `weblate://` URL](#)

Tip:

Table5Label new strings automatically

[Weblate `weblate://` URL](#) NOT has:label
[Weblate `weblate://` URL](#)

Table6[Weblate `weblate://` URL](#) [Weblate `weblate://` URL](#) [Weblate `weblate://` URL](#)

[Weblate `weblate://` URL](#) language:en AND key:changelogs/
[Weblate `weblate://` URL](#)read-only

Tip:

[Weblate `weblate://` URL](#)labels

[Weblate `weblate://` URL](#) "Weblate" [Weblate `weblate://` URL](#)

Tip: 3.1 **Tip:**

VCS [Weblate `weblate://` URL](#) "Weblate" [Weblate `weblate://` URL](#)

Tip: You might also want to tighten the [Weblate `weblate://` URL](#) check by adding `strict-same` flag to [Weblate `weblate://` URL](#).

Tip:

Translation states

[Weblate `weblate://` URL](#) "Weblate" [Weblate `weblate://` URL](#)

Tip: VCS [Weblate `weblate://` URL](#) "Weblate" [Weblate `weblate://` URL](#)

Tip:

Translation states

"???"

VCS ##### Weblate ??? "???" #####
??:

Translation states

#####

#####

Django #####

#####:

locale/{{ language_code }}.json

```
{
  "language": "{{ language_code }}",
  "strings": "{{ stats.all }}",
  "translated": "{{ stats.translated }}",
  "last_changed": "{{ stats.last_changed }}",
  "last_author": "{{ stats.last_author }}"
}
```

??:

#####

#####

#####

#####

Finding strings whose localized counterparts might not fit the layout is also possible.

??? : ##### Weblate ##### en_XA ? ar_XB #####

#####

PO #####

The PO file header will look like this:

```
# Michal Čihař <michal@cihar.com>, 2012, 2018, 2019, 2020.
# Pavel Borecki <pavel@example.com>, 2018, 2019.
# Filip Hron <filip@example.com>, 2018, 2019.
# anonymous <noreply@weblate.org>, 2019.
```

configure ##### ALL_LINGUAS #####

#####configure?configure.in##### configure.ac ##### ALL_LINGUAS
#####

gettext `XXXXXXXXXX`

```
XXXXXXXXXX gettext XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXX:
77 XXXXXXXXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
```

??: XXXXXXXXgettext `77 XXXXXXXXXXXXXXXXXXXX--no-wrap XXXXXXXXXXXXXXXXXXXXXXXXXXXX`

LINGUAS `XXXXXXXXXX`

```
XXXXXXXXXXXXXXXXXX LINGUAS XXXXXXXXXXXXXXXX
```

MO `XXXXXXXXXX`

```
XXXXXX PO XXXXXXXXXXXXXXXX MO XXXXXXXXXXXXXXXX
MO XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX XXXXXXXX
```

POT `XXXXXXXXXX` PO `XXXXXXXXXX`msgmerge`?`

```
msgmerge XXXXX POT XXXXX File mask XXXXXXXXXXXXXXXXXXXX PO XXXXX XXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXX
```

```
XXXXXXXXXX upstream XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX msgmerge XXXXX XXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

??:

Does Weblate update translation files besides translations?

Git `XXXXXXXXXX`

```
XXXXXXXXXXXXXXXXXXGit XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXX Git XXXXXXXXXXXXXXXXXXXXXXXX:
XXXXXX 3.4 XXX.
XXXXXXXXXXXX 1 XXXXXX
XXX
XXXXXX
XXXXXX 3.5 XXX.
XXXXXX
```

```
XXXXXXXXXX XXXXXXXXXXXXXXXXXXXXPer author XXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXX 4.1 XXX.
XXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX Co-authored-by: ... XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Co-authored-by: XXXXXXXXXXXXXXX
```

JSON [\[?\]](#)

`#####JSON #####`

Java [\[?\]](#) [\[?\]](#)

Java [\[?\]](#) [\[?\]](#)

[\[?\]](#)

3.7 [\[?\]](#).

`#####`

`#####`

[\[?\]](#)

3.7 [\[?\]](#).

`#####`

`#####`: [\[?\]](#)

RESX [\[?\]](#)

3.9 [\[?\]](#).

`#####`

[?]: `#####` [\[?\]](#) `#####`

[?]:

Does Weblate update translation files besides translations?

YAML [\[?\]](#)

3.10.2 [\[?\]](#).

`#####YAML #####`

[\[?\]](#)

`WEBLATE_ADDONS` `#####`

[\[?\]](#)

`#####` `weblate.addons.base.BaseAddon` `#####`

[?]:

[\[?\]](#)

????????????????

?? Weblate ???

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""Example pre commit script."""
```

```
from django.utils.translation import gettext_lazy as _

from weblate.addons.events import EVENT_PRE_COMMIT
from weblate.addons.scripts import BaseScriptAddon

class ExamplePreAddon(BaseScriptAddon):
    # Event used to trigger the script
    events = (EVENT_PRE_COMMIT,)
    # Name of the addon, has to be unique
    name = "weblate.example.pre"
    # Verbose name and long description
    verbose = _("Execute script before commit")
    description = _("This addon executes a script.")

    # Script to execute
    script = "/bin/true"
    # File to add in commit (for pre commit event)
    # does not have to be set
    add_file = "po/{{ language_code }}.po"
```

????????????? ?????????????????????????????????????? ??????????????

???????????????????????????????????? VCS ???

????????????????????:

WL_VCS

????????????????????????

WL_REPO

Upstream ??????? URL?

WL_PATH

VCS ??????????????????

WL_BRANCH

?????? 2.11 ???.

??

WL_FILEMASK

????????????????????????

WL_TEMPLATE

??

WL_NEW_BASE

?????? 2.14 ???.

244

Automatically, by translating strings using `gettext`, or `gettext` addon.

For installation tips, see [Weblate Translation Memory](#), which is turned on by default.

Translation memory scopes

3.2 In earlier versions translation memory could be only loaded from a file corresponding to the current imported translation memory scope.

The translation memory scopes are there to allow both privacy and sharing of translations, to suit the desired behavior.

Imported translation memory

Importing arbitrary translation memory data using the `import_memory` command makes memory content available to all users and projects.

Per user translation memory

Stores all user translations automatically in the personal translation memory of each respective user.

Per project translation memory

All translations within a project are automatically stored in a project translation memory only available for this project.

Shared translation memory

All translation within projects with shared translation memory turned on are stored in a shared translation memory available to all projects.

Please consider carefully whether to turn this feature on for shared Weblate installations, as it can have severe implications:

The translations can be used by anybody else.

This might lead to disclosing secret information.

Managing translation memory

User interface

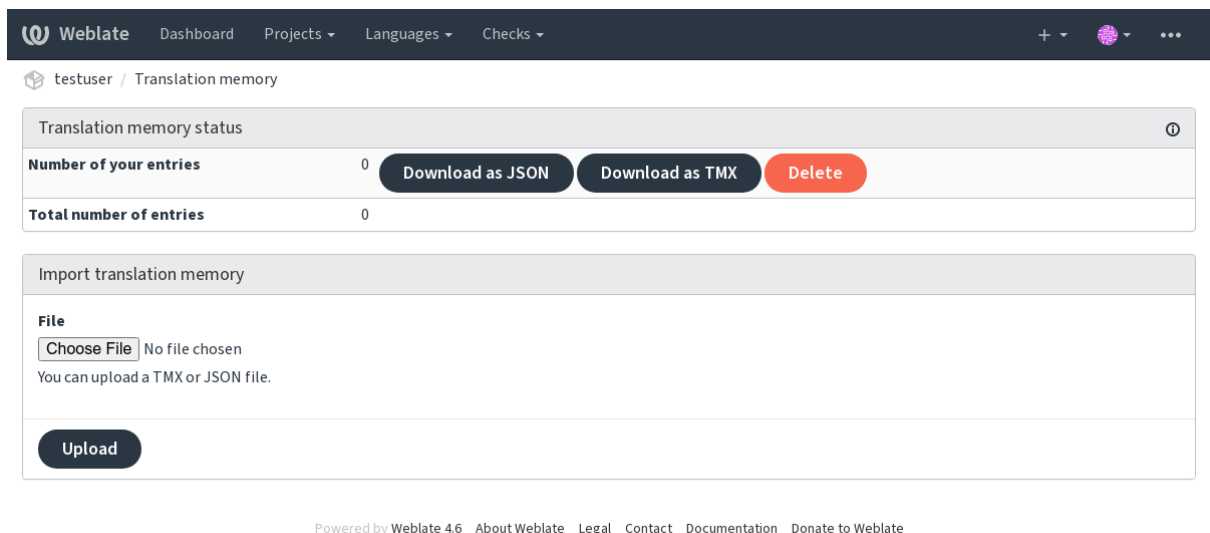
3.2.

In the basic user interface you can manage per user and per project translation memories. It can be used to download, wipe or import translation memory.

3.2: Translation memory in JSON can be imported into Weblate, TMX is provided for interoperability with other tools.

3.2:

Weblate `3.2`



????

There are several management commands to manipulate the translation memory content. These operate on the translation memory as whole, unfiltered by scopes (unless requested by parameters):

Exports the memory into JSON

Imports TMX or JSON files into the translation memory

??

settings.py #####Django

?: #####WeblateWSGI ##### Celery #####
mod_wsgi #####Apache #####

?:

Django #####Django's documentation #####

AKISMET_API_KEY

Weblate ? Akismet #####API #####akismet.com
#####

ANONYMOUS_USER_NAME

#####

?:

#####

AUDITLOG_EXPIRY

Default 3.6 days.
Default Webate days
Default 180 days

AUTH_LOCK_ATTEMPTS

Default 2.14 days.
Default
Default
Default days
Default days
Default 10 days
Default:
Default,

AUTO_UPDATE

Default 3.2 days.
Default 3.11 days: on/off Default
Default

Default: Webate Default Default

Default: Default on/off Default

Default:
Default
Default
Default

Default: Default Celery Default Default

AVATAR_URL_PREFIX

Default URL Default: \${AVATAR_URL_PREFIX}/avatar/\${MAIL_HASH}?
\${PARAMS}Default:
AVATAR_URL_PREFIX = 'https://www.gravatar.com/'
AVATAR_URL_PREFIX = 'https://www.libravatar.org/'
Default:
DefaultENABLE_AVATARSDefaultAvatars

AUTH_TOKEN_VALID

2.14.

1728002

AUTH_PASSWORD_DAYS

2.15.

: Weblate 2.15

180

AUTOFIX_LIST

: autofixer Python

:

......

safe-html HTML: [HTML](#)

:

```
AUTOFIX_LIST = (  
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",  
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",  
)
```

:

BACKGROUND_TASKS

4.5.2.

Defines how often lengthy maintenance tasks should be triggered for a component.

Right now this controls:

[recalculation](#)

[recalculation](#)

:

`weekly`

`daily`

`never`

: Increasing the frequency is not recommended when Weblate contains thousands of components.

BASE_DIR

DATA_DIR

Webplate

BASIC_LANGUAGES

QUESTION 4.4 (3 points)

[illegible][illegible]

Weblate

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

??: ??? Weblate ???

?:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

??:

?????

CSP_SCRIPT_SRC?CSP_IMG_SRC?CSP_CONNECT_SRC?CSP_STYLE_SRC?CSP_FONT_SRC

Weblate  Content-Security-Policy 

MatomoGoogle Sentry...

□ □

?:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

??:

???? ???? ???? ???? ???? CSP?

CHECK_LIST

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

🔗: check 🌐🌐🌐🌐🌐🌐🌐🌐 Python 🌐🌐🌐🌐🌐🌐🌐🌐🌐🌐🌐🌐🌐🌐

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

????????????????:

```
CHECK_LIST = ()
```

????????????????:

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```


DATABASE_BACKUP

3.1

Plain or compressed backup

"plain"

"compressed"

"none"

Example:

Weblate

DEFAULT_ACCESS_CONTROL

3.3

Access control

ACL

ACL

ACL

ACL

ACL Weblate

Example:

ACL

DEFAULT_AUTO_WATCH

4.5

Configures whether *Automatically watch projects on contribution* should be turned on for new users. Defaults to `True`.

Example:

True

DEFAULT_RESTRICTED_COMPONENT

4.1

Restricted component

Example:

True

DEFAULT_ADD_MESSAGE?DEFAULT_ADDON_MESSAGE?DEFAULT_COMMIT_MESSAGE?DEFAULT_DELETE_MESSAGE?

Component configuration

Example:

Component configuration

DEFAULT_ADDONS

XX

##: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

?:

```
DEFAULT_ADDONS = {
    # Addon with no parameters
    "weblate.flags.target_edit": {},
    # Addon with parameters
    "weblate.autotranslate.autotranslate": {
        "mode": "suggest",
        "filter_type": "todo",
        "auto_source": "mt",
        "component": "",
        "engines": ["weblate-translation-memory"],
        "threshold": "80",
    },
}
```

##:

install_addon?WEBLATE_ADDONS

DEFAULT_COMMITER_EMAIL

XXXXXX 2.4 XXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXX noreply@weblate.org XXX

##:

DEFAULT_COMMITER_NAME

DEFAULT_COMMITER_NAME

XXXXXX 2.4 XXX.

XXXXXXXXXXXXXXXXXXXX Weblate XXX

##:

DEFAULT_COMMITER_EMAIL

DEFAULT_LANGUAGE

XXXXXX 4.3.2 XXX.

[XXXXXX](#) XXXXXXXXXXXXXXXXXXXXXXX

XXXXXX en XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

##:

[XXXXXX?XXXXXX](#)

DEFAULT_MERGE_STYLE

3.4

rebase -

merge

:

Component configuration

DEFAULT_SHARED_TM

3.2

Configures default value of and .

DEFAULT_TRANSLATION_PROPAGATION

2.5

True

:

Component configuration

DEFAULT_PULL_MESSAGE

'Update from Weblate'

ENABLE_AVATARS

Gravatar -

:

AVATAR_URL_PREFIX

ENABLE_HOOKS

:

ENABLE_HTTPS

Weblate HTTPS HTTP URL

HTTPS Django - Cookie HSTS
HTTPS URL

HTTPS Django SSL
X-Forwarded-Proto For-
warded Django SSL SE-
CURE_PROXY_SSL_HEADER

:

SESSION_COOKIE_SECURE CSRF_COOKIE_SECURE SECURE_SSL_REDIRECT SECURE_PROXY_SSL_HEADER

ENABLE_SHARING

Webplate 4.5.2 2022. 2022 on/off 2022

GET_HELP_URL

Webplate 4.5.2 2022.

Webplate 2022 URL

GITLAB_CREDENTIALS

Webplate 4.3 2022.

GitLab 2022

2022: Webplate 2022 GitLab 2022 GitLab 2022 *GITLAB_USERNAME* 2022 *GITLAB_TOKEN* 2022

```
GITLAB_CREDENTIALS = {
  "gitlab.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitlab.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

GITLAB_USERNAME

GitLab 2022 2022

2022:

GITLAB_CREDENTIALS 2022 *GitLab*

GITLAB_TOKEN

Webplate 4.3 2022.

2022 API 2022 GitLab 2022 2022

2022:

GITLAB_CREDENTIALS 2022 *GitLab* 2022 *GitLab: Personal access token*

GITHUB_CREDENTIALS

Webplate 4.3 2022.

GitHub 2022

2022: Webplate 2022 GitHub 2022 GitHub 2022 *GITHUB_USERNAME* 2022 *GITHUB_TOKEN* 2022

```
GITHUB_CREDENTIALS = {
  "api.github.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
}
```

(20222022)

()

```

"github.example.com": {
  "username": "weblate",
  "token": "another-api-token",
},
}

```

GITHUB_USERNAME

GitHub

??:

`GITHUB_CREDENTIALS?` [GitHub](#)

GITHUB_TOKEN

□□□□□ 4.3 □□□.

GitHub API

??:

GITHUB_CREDENTIALS?GitHub?Creating a personal access token

GOOGLE_ANALYTICS_ID

Google 翻译 ID 11111 Weblate 111111111111

HIDE_REPO_CREDENTIALS

Web [REDACTED] URL [REDACTED] Weblate [REDACTED]
[REDACTED] URL [REDACTED]

```
??? https://user:password@git.example.com/repo.git ???? https://git.example.com/repo.git` ?????VCS ?????????????????????
```

??: ?????????????????

HIDE_VERSION

4.3.1 4.3.1

[illegible]

??: ??????????

IP_BEHIND_REVERSE_PROXY

2.14 $\frac{1}{2}$.

Weblate                             

True ?????Weblate ? IP_PROXY_HEADER IP ?

```
??      ???? ??????????????????????????????   ?????????????????????????????????
???????????????????????????? IP ?????????????????
```

??:

```

00000000 00000000 00000000 00000000 IP PROXY HEADER00000000 IP PROXY OFFSET

```

IP_PROXY_HEADER

Figure 2.14: [?].

IP_BEHIND_REVERSE_PROXY [REDACTED] Weblate [REDACTED] IP [REDACTED]

```
##### HTTP_X_FORWARDED_FOR #####
```

??:

```

0000 000000000000?00000?SECURE_PROXY_SSL_HEADER?IP_BEHIND_REVERSE_PROXY?IP_PROXY_OFFSET

```

IP_PROXY_OFFSET

Ex. 2.14. $\square$

```

00000000 IP 00000000 IP_PROXY_HEADER 000000000000000000000000

```

[illegible]

000000 0 000

??:

```

[REDACTED]SECURE_PROXY_SSL_HEADER,IP_BEHIND_REVERSE_PROXY[REDACTED]IP_PROXY_HEADER

```

LEGAL_URL

3.5

?:

```
LEGAL_URL = "https://weblate.org/terms/"
```

LICENSE EXTRA

????????????????

[?]:

```
LICENSE_EXTRA = [
    (
        "AGPL-3.0",
        "GNU Affero General Public License v3.0",
        "https://www.gnu.org/licenses/agpl-3.0-standalone.html",
    ),
]
```

LICENSE_FILTER

4.3:

:

:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

:

```
LICENSE_FILTER = set()
```

:

alerts

LICENSE_REQUIRED

Component configuration

:

LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

* 10

: False 10,000

: True

LOCALIZE_CDN_URL LOCALIZE_CDN_PATH

JavaScript CDN LOCALIZE_CDN_URL CDN
 URL LOCALIZE_CDN_PATH LOCALIZE_CDN_URL Weblate

: Hosted Weblate https://weblate-cdn.com/

:

JavaScript CDN

LOGIN_REQUIRED_URLS

URL Weblate

:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)  
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [  
    "rest_framework.permissions.IsAuthenticated"  
]
```

???: `????????????API ?????????????????????`

?:

`REQUIRE_LOGIN`

LOGIN_REQUIRED_URLS_EXCEPTIONS

`LOGIN_REQUIRED_URLS` `????????????????????????????????????????????????????????????`

`????????????????`:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (  
    r"/accounts/(.*)$", # Required for sign in  
    r"/static/(.*)$", # Required for development mode  
    r"/widgets/(.*)$", # Allowing public access to widgets  
    r"/data/(.*)$", # Allowing public access to data exports  
    r"/hooks/(.*)$", # Allowing public access to notification hooks  
    r"/api/(.*)$", # Allowing access to API  
    r"/js/i18n/$", # JavaScript localization  
)
```

MATOMO_SITE_ID

`???? Matomo?? Piwik???????? ID?`

?: `????Matomo ?? ?????????????????????`

?:

`MATOMO_URL`

MATOMO_URL

Weblate `????????????????????` Matomo`????` Piwik`????????????????` URL`????????????????????????????` `<https://matomo.org/>` `????????????`

???: `????Matomo ?? ?????????????????????`

`?`:

```
MATOMO_SITE_ID = 1  
MATOMO_URL = "https://example.matomo.cloud/"
```

?:

`MATOMO_SITE_ID`

MT_SERVICES

`???? 3.0` `???:` `????????????????????????????????????????` `MACHINE_TRANSLATION_SERVICES` `??` `MT_SERVICES` `????????`

`????????????????????????????`

?: `???????? API ?????????????????????????????????????????` `????` `????????????????????`

```
MT_SERVICES = (
    "weblate.machinery.apertium.ApertiumAPYTranslation",
    "weblate.machinery.deepl.DeepLTranslation",
    "weblate.machinery.glosbe.GlosbeTranslation",
    "weblate.machinery.google.GoogleTranslation",
    "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    "weblate.machinery.mymemory.MyMemoryTranslation",
    "weblate.machinery.tmserver.AmagamaTranslation",
    "weblate.machinery.tmserver.TMServerTranslation",
    "weblate.machinery.yandex.YandexTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.machinery.saptranslationhub.SAPTranslationHub",
    "weblate.memory.machine.WeblateMemory",
)
```

??:

????? ?????

MT_APERTIUM_APY

Apertium-APy *?????* URL <https://wiki.apertium.org/wiki/Apertium-apy>

??:

Apertium?????

MT_AWS_ACCESS_KEY_ID

Amazon *????????* ID

??:

AWS?????

MT_AWS_SECRET_ACCESS_KEY

???????? API *????*

??:

AWS?????

MT_AWS_REGION

Amazon *????????????????*

??:

AWS?????

MT_Baidu_ID

Baidu Zhiyun API *????????* ID <https://api.fanyi.baidu.com/api/trans/product/index> *????*

??:

Baidu API machine translation?????

MT_Baidu_SECRET

Baidu Zhiyun API <https://api.fanyi.baidu.com/api/trans/product/index>

Baidu API machine translation

MT_DEEPL_API_VERSION

4.1.1

DeepL API

CAT

API

Webate DeepL CAT v1 API v2 CAT Webate v1

DeepL

MT_DEEPL_KEY

DeepL API <https://www.deepl.com/pro.html>

DeepL

MT_GOOGLE_KEY

Google Translate API v2 API register at <https://cloud.google.com/translate/docs>

Google Translate

MT_GOOGLE_CREDENTIALS

Google API v3 JSON OS <https://cloud.google.com/docs/authentication/getting-started>

MT_GOOGLE_PROJECT

Google API v3 ID <https://cloud.google.com/appengine/docs/standard/nodejs/building-app/creating-project>

MT_GOOGLE_LOCATION

API v3 Google <https://cloud.google.com/appengine/docs/locations> global

<https://cloud.google.com/appengine/docs/locations>

Google Translate API V3 (Advanced)

MT_MYMEMORY_USER

MyMemory ID `MT_MYMEMORY_KEY`

MyMemory: API key generator

MT_NETEASE_KEY

NetEase Sight API App key `https://sight.youdao.com/`

NetEase Sight API machine translation

MT_NETEASE_SECRET

NetEase Sight API App secret `https://sight.youdao.com/`

NetEase Sight API machine translation

MT_TMSERVER

tmserver URL

tmserver

MT_YANDEX_KEY

Yandex Translate API API key `https://yandex.com/dev/translate/`

Yandex Translate

MT_YOUDAO_ID

Youdao Zhiyun API ID `https://ai.youdao.com/product-fanyi-text.s`

Youdao Zhiyun API machine translation

MT_YOUDAO_SECRET

Youdao Zhiyun API Client secret `https://ai.youdao.com/product-fanyi-text.s`

Youdao Zhiyun API machine translation

PAGURE_USERNAME

4.3.2 .

Pagure

:

`PAGURE_CREDENTIALS` `Pagure`

PAGURE_TOKEN

4.3.2 .

API Pagure

:

`PAGURE_CREDENTIALS` `Pagure` `Pagure API`

RATELIMIT_ATTEMPTS

3.2 .

5

:

`RATELIMIT_WINDOW` `RATELIMIT_LOCKOUT`

RATELIMIT_WINDOW

3.2 .

300 25

:

`RATELIMIT_ATTEMPTS` `RATELIMIT_LOCKOUT`

RATELIMIT_LOCKOUT

3.2 .

600 10

:

`RATELIMIT_ATTEMPTS` `RATELIMIT_WINDOW`

REGISTRATION_ALLOW_BACKENDS

4.1 .

`REGISTRATION_OPEN` Weblate

:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

###: ##### URL #####

##:

REGISTRATION_OPEN###

REGISTRATION_CAPTCHA

CAPTCHA ##### True ## False #####
True #####

CAPTCHA

#####

#####

#####

#####

REGISTRATION_EMAIL_MATCH

2.17 ##.

#####

.*#####

#####

REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org\$"

REGISTRATION_OPEN

True ##### False

Python Social Auth #####:setting: REGISTRATION_ALLOW_BACKENDS'#####

##: *LDAP* ##### #####
#####

##:

REGISTRATION_ALLOW_BACKENDS#REGISTRATION_EMAIL_MATCH###

REPOSITORY_ALERT_THRESHOLD

4.0.2 ##.

25

##:

alerts

REQUIRE_LOGIN

□□□□□ 4.1 □□□.

?????	<i>LOGIN_REQUIRED_URLS</i>	??????REST	??????????????	API
????????????????????????????????				

```
[[??: ??? [???] ?????????? Docker [???] WEBLATE_REQUIRE_LOGIN [?????]
```

SENTRY_DSN

□□□□□ 3.9 □□□.

Sentry DSN

??:

Sentry ? Django ??

SESSION_COOKIE_AGE_AUTHENTICATED

Figure 4.3: Example.

[illegible]

??:

SESSION_COOKIE_AGE

SIMPLIFY_LANGUAGES

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX f r _ F R XXXX f r XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

SITE_DOMAIN

??? ??????????????????: ??????????RSS ??????????????????

Weblate

?:

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

[illegible]

####: Docker ##### WEBLATE ALLOWED HOSTS #####

??:

`[[[?]]]HTTPS [[?]] WEBLATE SITE DOMAIN[?]ENABLE HTTPS`

SITE_TITLE

Web [?]

SPECIAL_CHARS

1 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 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 104

```
SPECIAL_CHARS = ("\t", "\n", "...")
```

SINGLE_PROJECT

QUESTION 3.8 [3].

```

True We-
blate 1

```

3.11: $\frac{1}{2} \left(\frac{1}{2} + \frac{1}{2} \right) = \frac{1}{2}$

?:

```
SINGLE_PROJECT = "test"
```

STATUS_URL

Weblate URL

SUGGESTION_CLEANUP_DAYS

□□□□□ 3.2.1 □□□.

None

UPDATE_LANGUAGES

4.3.2 4.3.2

[illegible]

setuplang

??:

??????????

URL_PREFIX

Webplate Webplate

```
##:                WSGI  WSGIScriptAlias
#####
```

???: ??? / ?????????

?:

```
URL_PREFIX = "/translations"
```

```
???: ????? Django ??????????urls.py ??????????
```


Webplate **Webplate**

??:

[[[[]]]]DEFAULT ADDONS

WEBLATE EXPORTERS

4.2. $\mathbb{R}^n$

????????????????????

??:

??????????

WEBLATE FORMATS

3.0

□□□□□□□□□□□□□□□□

??: ?????????????????????

??:

??????????

WEBLATE GPG IDENTITY

3.1 3.1.

Weblate [?] Git [????????????????] ID[????]:

```
WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"
```

Webplate GPG `DATA_DIR` home/.gnupg

??:

Signing Git commits with GnuPG

WEBSITE_REQUIRED

Defines whether `Web [??]` has to be specified when creating a project. Turned on by default as that suits public server setups.

???

```
Webdate [REDACTED] webdate/settings example.py [REDACTED]
```

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
```

(XXXXXXXXXX)

```

# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#

import os
import platform
from logging.handlers import SysLogHandler

#
# Django settings for Weblate project.
#

DEBUG = True

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

DATABASES = {
    "default": {
        # Use "postgresql" or "mysql".
        "ENGINE": "django.db.backends.postgresql",
        # Database name.
        "NAME": "weblate",
        # Database user.
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for
↪authentication.
        "ALTER_ROLE": "weblate",
        # Database password.
        "PASSWORD": "",
        # Set to empty string for localhost.
        "HOST": "127.0.0.1",
        # Set to empty string for default.
        "PORT": "",
        # Customizations for databases.
        "OPTIONS": {
            # In case of using an older MySQL server,
            # which has MyISAM as a default storage
            # "init_command": "SET storage_engine=INNODB",
            # Uncomment for MySQL older than 5.7:
            # "init_command": "SET sql_mode='STRICT_TRANS_TABLES'",
            # Set emoji capable charset for MySQL:
            # "charset": "utf8mb4",
            # Change connection timeout in case you get MySQL gone away
↪error:
            # "connect_timeout": 28800,
        },
    }
}

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Data directory
DATA_DIR = os.path.join(BASE_DIR, "data")

# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List\_of\_tz\_zones\_by\_name
# although not all choices may be available on all operating systems.

```

(XXXXXXXXXX)

(XXXXXXXXXX)

```
# In a Windows environment this must be set to your system time zone.
TIME_ZONE = "UTC"

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

LANGUAGES = (
    ("ar", "العربية"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be@latin", "Biełaruskaja"),
    ("bg", "Български"),
    ("br", "Brezhoneg"),
    ("ca", "Català"),
    ("cs", "Čeština"),
    ("da", "Dansk"),
    ("de", "Deutsch"),
    ("en", "English"),
    ("el", "Ελληνικά"),
    ("en-gb", "English (United Kingdom)"),
    ("es", "Español"),
    ("fi", "Suomi"),
    ("fr", "Français"),
    ("gl", "Galego"),
    ("he", "עברית"),
    ("hu", "Magyar"),
    ("hr", "Hrvatski"),
    ("id", "Indonesia"),
    ("is", "Íslenska"),
    ("it", "Italiano"),
    ("ja", "日本語"),
    ("kab", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "한국어"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),
    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "简体中文"),
    ("zh-hant", "繁體中文"),
)

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
```

(XXXXXXXXXX)

(XXXXXXXXXX)

```
# Type of automatic primary key, introduced in Django 3.2
DEFAULT_AUTO_FIELD = "django.db.models.AutoField"

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded
↪files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(DATA_DIR, "static")

# URL prefix for static files.
STATIC_URL = f"{URL_PREFIX}/static/"

# Additional locations of static files
STATICFILES_DIRS = (
    # Put strings here, like "/home/html/static" or "C:/www/django/static".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    "django.contrib.staticfiles.finders.FileSystemFinder",
    "django.contrib.staticfiles.finders.AppDirectoriesFinder",
    "compressor.finders.CompressorFinder",
)

# Make this unique, and don't share it with anybody.
# You can generate it using weblate/examples/generate-secret-key
SECRET_KEY = ""

_TEMPLATE_LOADERS = [
    "django.template.loaders.filesystem.Loader",
    "django.template.loaders.app_directories.Loader",
]
if not DEBUG:
    _TEMPLATE_LOADERS = [("django.template.loaders.cached.Loader", _
↪TEMPLATE_LOADERS)]
TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
            "loaders": _TEMPLATE_LOADERS,
        },
    },
]
```

(XXXXXXXXXX)

```

# GitHub username and token for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None
GITHUB_TOKEN = None

# GitLab username and token for sending merge requests.
# Please see the documentation for more details.
GITLAB_USERNAME = None
GITLAB_TOKEN = None

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth",
    # "social_core.backends.suse.OpenSUSEOpenId",
    # "social_core.backends.ubuntu.UbuntuOpenId",
    # "social_core.backends.fedora.FedoraOpenId",
    # "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]

SOCIAL_AUTH_BITBUCKET_KEY = ""
SOCIAL_AUTH_BITBUCKET_SECRET = ""
SOCIAL_AUTH_BITBUCKET_VERIFIED_EMAILS_ONLY = True

SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}

SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# 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",

```

(XXXXXXXXXX)

```
        "weblate.accounts.pipeline.password_reset",
    )
    SOCIAL_AUTH_DISCONNECT_PIPELINE = (
        "social_core.pipeline.disconnect.allowed_to_disconnect",
        "social_core.pipeline.disconnect.get_entries",
        "social_core.pipeline.disconnect.revoke_tokens",
        "weblate.accounts.pipeline.cycle_session",
        "weblate.accounts.pipeline.adjust_primary_mail",
        "weblate.accounts.pipeline.notify_disconnect",
        "social_core.pipeline.disconnect.disconnect",
        "weblate.accounts.pipeline.cleanup_next",
    )

    # Custom authentication strategy
    SOCIAL_AUTH_STRATEGY = "weblate.accounts.strategy.WeblateStrategy"

    # Raise exceptions so that we can handle them later
    SOCIAL_AUTH_RAISE_EXCEPTIONS = True

    SOCIAL_AUTH_EMAIL_VALIDATION_FUNCTION = "weblate.accounts.pipeline.send_
    ↪validation"
    SOCIAL_AUTH_EMAIL_VALIDATION_URL = f"{URL_PREFIX}/accounts/email-sent/"
    SOCIAL_AUTH_LOGIN_ERROR_URL = f"{URL_PREFIX}/accounts/login/"
    SOCIAL_AUTH_EMAIL_FORM_URL = f"{URL_PREFIX}/accounts/email/"
    SOCIAL_AUTH_NEW_ASSOCIATION_REDIRECT_URL = f"{URL_PREFIX}/accounts/profile/
    ↪#account"
    SOCIAL_AUTH_PROTECTED_USER_FIELDS = ("email",)
    SOCIAL_AUTH_SLUGIFY_USERNAMES = True
    SOCIAL_AUTH_SLUGIFY_FUNCTION = "weblate.accounts.pipeline.slugify_username"

    # Password validation configuration
    AUTH_PASSWORD_VALIDATORS = [
        {
            "NAME": "django.contrib.auth.password_validation.
    ↪UserAttributeSimilarityValidator" # noqa: E501, pylint: disable=line-
    ↪too-long
        },
        {
            "NAME": "django.contrib.auth.password_validation.
    ↪MinimumLengthValidator",
            "OPTIONS": {"min_length": 10},
        },
        {
            "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",
    ]
```

(XXXXXXXXXX)

```

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [
    "weblate.middleware.RedirectMiddleware",
    "weblate.middleware.ProxyMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "weblate.accounts.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
    "social_django.middleware.SocialAuthExceptionMiddleware",
    "weblate.accounts.middleware.RequireLoginMiddleware",
    "weblate.api.middleware.ThrottlingMiddleware",
    "weblate.middleware.SecurityMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.formats",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",
    "weblate.screenshots",
    "weblate.fonts",
    "weblate.accounts",
    "weblate.configuration",
    "weblate.utils",
    "weblate.vcs",
    "weblate.wladmin",
    "weblate.metrics",
    "weblate",
    # Optional: Git exporter
    "weblate.gitexport",
    # Standard Django modules
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "django.contrib.admin.apps.SimpleAdminConfig",
    "django.contrib.admindocs",
    "django.contrib.sitemaps",
    "django.contrib.humanize",
    # Third party Django modules
    "social_django",
    "crispy_forms",
    "compressor",
    "rest_framework",
    "rest_framework.authtoken",
    "django_filters",
]

```

```

# Custom exception reporter to include some details
DEFAULT_EXCEPTION_REPORTER_FILTER = "weblate.trans.debug.
↪WeblateExceptionReporterFilter"

# Default logging of Weblate messages
# - to syslog in production (if available)
# - otherwise to console
# - you can also choose "logfile" to log into separate file
#   after configuring it below

# Detect if we can connect to syslog
HAVE_SYSLOG = False
if platform.system() != "Windows":
    try:
        handler = SysLogHandler(address="/dev/log", facility=SysLogHandler.
↪LOG_LOCAL2)
        handler.close()
        HAVE_SYSLOG = True
    except OSError:
        HAVE_SYSLOG = False

if DEBUG or not HAVE_SYSLOG:
    DEFAULT_LOG = "console"
else:
    DEFAULT_LOG = "syslog"
DEFAULT_LOGLEVEL = "DEBUG" if DEBUG else "INFO"

# A sample logging configuration. The only tangible logging
# performed by this configuration is to send an email to
# the site admins on every HTTP 500 error when DEBUG=False.
# See http://docs.djangoproject.com/en/stable/topics/logging for
# more details on how to customize your logging configuration.
LOGGING = {
    "version": 1,
    "disable_existing_loggers": True,
    "filters": {"require_debug_false": {"()": "django.utils.log.
↪RequireDebugFalse"}},
    "formatters": {
        "syslog": {"format": "weblate[%(process)d]: %(levelname)s
↪%(message)s"},
        "simple": {"format": "[%asctime)s: %(levelname)s/%(process)s]
↪%(message)s"},
        "logfile": {"format": "%(asctime)s %(levelname)s %(message)s"},
        "django.server": {
            "()": "django.utils.log.ServerFormatter",
            "format": "[%server_time)s] %(message)s",
        },
    },
    "handlers": {
        "mail_admins": {
            "level": "ERROR",
            "filters": ["require_debug_false"],
            "class": "django.utils.log.AdminEmailHandler",
            "include_html": True,
        },
        "console": {
            "level": "DEBUG",
            "class": "logging.StreamHandler",
            "formatter": "simple",
        },
        "django.server": {
            "level": "INFO",
            "class": "logging.StreamHandler",
            "formatter": "django.server",
        },
        "syslog": {

```

```

        "level": "DEBUG",
        "class": "logging.handlers.SysLogHandler",
        "formatter": "syslog",
        "address": "/dev/log",
        "facility": SysLogHandler.LOG_LOCAL2,
    },
    # Logging to a file
    # "logfile": {
    #     "level": "DEBUG",
    #     "class": "logging.handlers.RotatingFileHandler",
    #     "filename": "/var/log/weblate/weblate.log",
    #     "maxBytes": 100000,
    #     "backupCount": 3,
    #     "formatter": "logfile",
    # },
},
"loggers": {
    "django.request": {
        "handlers": ["mail_admins", DEFAULT_LOG],
        "level": "ERROR",
        "propagate": True,
    },
    "django.server": {
        "handlers": ["django.server"],
        "level": "INFO",
        "propagate": False,
    },
    # Logging database queries
    # "django.db.backends": {
    #     "handlers": [DEFAULT_LOG],
    #     "level": "DEBUG",
    # },
    "weblate": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Logging VCS operations
    "weblate.vcs": {"handlers": [DEFAULT_LOG], "level": DEFAULT_
↪LOGLEVEL},
    # Python Social Auth
    "social": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Django Authentication Using LDAP
    "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_
↪LOGLEVEL},
    # SAML IdP
    "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_
↪LOGLEVEL},
    },
}

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = (
    #     "weblate.machinery.apertium.ApertiumAPYTranslation",
    #     "weblate.machinery.baidu.BaiduTranslation",
    #     "weblate.machinery.deepl.DeepLTranslation",
    #     "weblate.machinery.glosbe.GlosbeTranslation",
    #     "weblate.machinery.google.GoogleTranslation",
    #     "weblate.machinery.googlev3.GoogleV3Translation",
    #     "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    #     "weblate.machinery.microsoftterminology.
↪MicrosoftTerminologyService",
    #     "weblate.machinery.modernmt.ModernMTTranslation",
    #     "weblate.machinery.mymemory.MyMemoryTranslation",
    #     "weblate.machinery.netease.NeteaseSightTranslation",
    #     "weblate.machinery.tmserver.AmagamaTranslation",
    #     "weblate.machinery.tmserver.TMServerTranslation",

```

```

# "weblate.machinery.yandex.YandexTranslation",
# "weblate.machinery.saptranslationhub.SAPTranslationHub",
# "weblate.machinery.youdao.YoudaoTranslation",
"weblate.machinery.weblatetm.WeblateTranslation",
"weblate.memory.machine.WeblateMemory",
)

# Machine translation API keys

# URL of the Apertium APy server
MT_APERTIUM_APY = None

# DeepL API key
MT_DEEPL_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

# ModernMT
MT_MODERNMT_KEY = None

# MyMemory identification email, see
# https://mymemory.translated.net/doc/spec.php
MT_MYMEMORY_EMAIL = None

# Optional MyMemory credentials to access private translation memory
MT_MYMEMORY_USER = None
MT_MYMEMORY_KEY = None

# Google API key for Google Translate API v2
MT_GOOGLE_KEY = None

# Google Translate API3 credentials and project id
MT_GOOGLE_CREDENTIALS = None
MT_GOOGLE_PROJECT = None

# Baidu app key and secret
MT_BAIDU_ID = None
MT_BAIDU_SECRET = None

# Youdao Zhiyun app key and secret
MT_YOUDAO_ID = None
MT_YOUDAO_SECRET = None

# Netease Sight (Jianwai) app key and secret
MT_NETEASE_KEY = None
MT_NETEASE_SECRET = None

# API key for Yandex Translate API
MT_YANDEX_KEY = None

# tmserver URL
MT_TMSERVER = None

# SAP Translation Hub
MT_SAP_BASE_URL = None
MT_SAP_SANDBOX_APIKEY = None
MT_SAP_USERNAME = None
MT_SAP_PASSWORD = None
MT_SAP_USE_MT = True

# Title of site to use
SITE_TITLE = "Weblate"

# Site domain

```

```

SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

# Use HTTPS when creating redirect URLs for social authentication, see
# documentation for more details:
# https://python-social-auth-docs.readthedocs.io/en/latest/configuration/
# settings.html#processing-redirects-and-urlopen
SOCIAL_AUTH_REDIRECT_IS_HTTPS = ENABLE_HTTPS

# Make CSRF cookie HttpOnly, see documentation for more details:
# https://docs.djangoproject.com/en/1.11/ref/settings/#csrf-cookie-httponly
CSRF_COOKIE_HTTPONLY = True
CSRF_COOKIE_SECURE = ENABLE_HTTPS
# Store CSRF token in session
CSRF_USE_SESSIONS = True
# Customize CSRF failure view
CSRF_FAILURE_VIEW = "weblate.trans.views.error.csrf_failure"
SESSION_COOKIE_SECURE = ENABLE_HTTPS
SESSION_COOKIE_HTTPONLY = True
# SSL redirect
SECURE_SSL_REDIRECT = ENABLE_HTTPS
# Sent referrrrer only for same origin links
SECURE_REFERRER_POLICY = "same-origin"
# SSL redirect URL exemption list
SECURE_REDIRECT_EXEMPT = (r"healthz/$",) # Allowing HTTP access to health_
# check
# Session cookie age (in seconds)
SESSION_COOKIE_AGE = 1000
SESSION_COOKIE_AGE_AUTHENTICATED = 1209600
SESSION_COOKIE_SAMESITE = "Lax"
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 50000000

# Apply session coookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings

```

```

IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow
↪ longer
# translations (up to 10.000 characters)
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True

# Use simple language codes for default language/country combinations
SIMPLIFY_LANGUAGES = True

# Render forms using bootstrap
CRISPY_TEMPLATE_PACK = "bootstrap3"

# List of quality checks
# CHECK_LIST = (
#     "weblate.checks.same.SameCheck",
#     "weblate.checks.chars.BeginNewlineCheck",
#     "weblate.checks.chars.EndNewlineCheck",
#     "weblate.checks.chars.BeginSpaceCheck",
#     "weblate.checks.chars.EndSpaceCheck",
#     "weblate.checks.chars.DoubleSpaceCheck",
#     "weblate.checks.chars.EndStopCheck",
#     "weblate.checks.chars.EndColonCheck",
#     "weblate.checks.chars.EndQuestionCheck",
#     "weblate.checks.chars.EndExclamationCheck",
#     "weblate.checks.chars.EndEllipsisCheck",
#     "weblate.checks.chars.EndSemicolonCheck",
#     "weblate.checks.chars.MaxLengthCheck",
#     "weblate.checks.chars.KashidaCheck",
#     "weblate.checks.chars.PunctuationSpacingCheck",
#     "weblate.checks.format.PythonFormatCheck",
#     "weblate.checks.format.PythonBraceFormatCheck",
#     "weblate.checks.format.PHPFormatCheck",
#     "weblate.checks.format.CFormatCheck",
#     "weblate.checks.format.PperlFormatCheck",
#     "weblate.checks.format.JavaScriptFormatCheck",
#     "weblate.checks.format.LuaFormatCheck",
#     "weblate.checks.format.SchemeFormatCheck",
#     "weblate.checks.format.CSharpFormatCheck",
#     "weblate.checks.format.JavaFormatCheck",
#     "weblate.checks.format.JavaMessageFormatCheck",
#     "weblate.checks.format.PercentPlaceholdersCheck",
#     "weblate.checks.format.VueFormattingCheck",
#     "weblate.checks.format.I18NextInterpolationCheck",
#     "weblate.checks.format.ESTemplateLiteralsCheck",
#     "weblate.checks.angularjs.AngularJSInterpolationCheck",
#     "weblate.checks.qt.QtFormatCheck",
#     "weblate.checks.qt.QtPluralCheck",
#     "weblate.checks.ruby.RubyFormatCheck",
#     "weblate.checks.consistency.PluralsCheck",
#     "weblate.checks.consistency.SamePluralsCheck",
#     "weblate.checks.consistency.ConsistencyCheck",
#     "weblate.checks.consistency.TranslatedCheck",
#     "weblate.checks.chars.EscapedNewlineCountingCheck",
#     "weblate.checks.chars.NewLineCountCheck",

```

```

# "weblate.checks.markup.BBCodeCheck",
# "weblate.checks.chars.ZeroWidthSpaceCheck",
# "weblate.checks.render.MaxSizeCheck",
# "weblate.checks.markup.XMLValidityCheck",
# "weblate.checks.markup.XMLTagsCheck",
# "weblate.checks.markup.MarkdownRefLinkCheck",
# "weblate.checks.markup.MarkdownLinkCheck",
# "weblate.checks.markup.MarkdownSyntaxCheck",
# "weblate.checks.markup.URLCheck",
# "weblate.checks.markup.SafeHTMLCheck",
# "weblate.checks.placeholders.PlaceholderCheck",
# "weblate.checks.placeholders.RegexCheck",
# "weblate.checks.duplicate.DuplicateCheck",
# "weblate.checks.source.OptionalPluralCheck",
# "weblate.checks.source.EllipsisCheck",
# "weblate.checks.source.MultipleFailingCheck",
# "weblate.checks.source.LongUntranslatedCheck",
# "weblate.checks.format.MultipleUnnamedFormatsCheck",
# "weblate.checks.glossary.GlossaryCheck",
# )

# List of automatic fixups
# AUTOFIX_LIST = (
#     "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
#     "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
#     "weblate.trans.autofixes.chars.RemoveZeroSpace",
#     "weblate.trans.autofixes.chars.RemoveControlChars",
# )

# List of enabled addons
# WEBLATE_ADDONS = (
#     "weblate.addons.gettext.GenerateMoAddon",
#     "weblate.addons.gettext.UpdateLinguasAddon",
#     "weblate.addons.gettext.UpdateConfigureAddon",
#     "weblate.addons.gettext.MsgmergeAddon",
#     "weblate.addons.gettext.GettextCustomizeAddon",
#     "weblate.addons.gettext.GettextAuthorComments",
#     "weblate.addons.cleanup.CleanupAddon",
#     "weblate.addons.cleanup.RemoveBlankAddon",
#     "weblate.addons.consistency.LangaugeConsistencyAddon",
#     "weblate.addons.discovery.DiscoveryAddon",
#     "weblate.addons.autotranslate.AutoTranslateAddon",
#     "weblate.addons.flags.SourceEditAddon",
#     "weblate.addons.flags.TargetEditAddon",
#     "weblate.addons.flags.SameEditAddon",
#     "weblate.addons.flags.BulkEditAddon",
#     "weblate.addons.generate.GenerateFileAddon",
#     "weblate.addons.generate.PseudolocaleAddon",
#     "weblate.addons.json.JSONCustomizeAddon",
#     "weblate.addons.properties.PropertiesSortAddon",
#     "weblate.addons.git.GitSquashAddon",
#     "weblate.addons.removal.RemoveComments",
#     "weblate.addons.removal.RemoveSuggestions",
#     "weblate.addons.resx.ResxUpdateAddon",
#     "weblate.addons.yaml.YAMLCustomizeAddon",
#     "weblate.addons.cdn.CDNJSAddon",
# )

# E-mail address that error messages come from.
SERVER_EMAIL = "noreply@example.com"

# Default email address to use for various automated correspondence from
# the site managers. Used for registration emails.
DEFAULT_FROM_EMAIL = "noreply@example.com"

# List of URLs your site is supposed to serve
ALLOWED_HOSTS = ["*"]

```

```

# Configuration for caching
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/1",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # "LOCATION": "unix:///var/run/redis/redis.sock?db=1",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
            # If you set password here, adjust CELERY_BROKER_URL as well
            "PASSWORD": None,
            "CONNECTION_POOL_KWARGS": {},
        },
        "KEY_PREFIX": "weblate",
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "TIMEOUT": 86400,
        "OPTIONS": {"MAX_ENTRIES": 1000},
    },
}

# Store sessions in cache
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
# Store messages in session
MESSAGE_STORAGE = "django.contrib.messages.storage.session.SessionStorage"

# REST framework settings for API
REST_FRAMEWORK = {
    # Use Django's standard `django.contrib.auth` permissions,
    # or allow read-only access for unauthenticated users.
    "DEFAULT_PERMISSION_CLASSES": [
        # Require authentication for login required sites
        "rest_framework.permissions.IsAuthenticated"
        if REQUIRE_LOGIN
        else "rest_framework.permissions.IsAuthenticatedOrReadOnly"
    ],
    "DEFAULT_AUTHENTICATION_CLASSES": (
        "rest_framework.authentication.TokenAuthentication",
        "weblate.api.authentication.BearerAuthentication",
        "rest_framework.authentication.SessionAuthentication",
    ),
    "DEFAULT_THROTTLE_CLASSES": (
        "weblate.api.throttling.UserRateThrottle",
        "weblate.api.throttling.AnonRateThrottle",
    ),
    "DEFAULT_THROTTLE_RATES": {"anon": "100/day", "user": "5000/hour"},
    "DEFAULT_PAGINATION_CLASS": ("rest_framework.pagination.
↪PageNumberPagination"),
    "PAGE_SIZE": 20,
    "VIEW_DESCRIPTION_FUNCTION": "weblate.api.views.get_view_description",
    "UNAUTHENTICATED_USER": "weblate.auth.models.get_anonymous",
}

# Fonts CDN URL
FONTS_CDN_URL = None

# Django compressor offline mode
COMPRESS_OFFLINE = False
COMPRESS_OFFLINE_CONTEXT = [
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_
↪BIDI": True},
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_
↪BIDI": False},

```

```

]

# Require login for all URLs
if REQUIRE_LOGIN:
    LOGIN_REQUIRED_URLS = (r"/(.*)$",)

# In such case you will want to include some of the exceptions
# LOGIN_REQUIRED_URLS_EXCEPTIONS = (
#     rf"{URL_PREFIX}/accounts/(.*)$", # Required for login
#     rf"{URL_PREFIX}/admin/login/(.*)$", # Required for admin login
#     rf"{URL_PREFIX}/static/(.*)$", # Required for development mode
#     rf"{URL_PREFIX}/widgets/(.*)$", # Allowing public access to widgets
#     rf"{URL_PREFIX}/data/(.*)$", # Allowing public access to data exports
#     rf"{URL_PREFIX}/hooks/(.*)$", # Allowing public access to_
↪notification hooks
#     rf"{URL_PREFIX}/healthz/$", # Allowing public access to health check
#     rf"{URL_PREFIX}/api/(.*)$", # Allowing access to API
#     rf"{URL_PREFIX}/js/i18n/$", # JavaScript localization
#     rf"{URL_PREFIX}/contact/$", # Optional for contact form
#     rf"{URL_PREFIX}/legal/(.*)$", # Optional for legal app
# )

# Silence some of the Django system checks
SILENCED_SYSTEM_CHECKS = [
    # We have modified django.contrib.auth.middleware.
↪AuthenticationMiddleware
    # as weblate.accounts.middleware.AuthenticationMiddleware
    "admin.E408"
]

# Celery worker configuration for testing
# CELERY_TASK_ALWAYS_EAGER = True
# CELERY_BROKER_URL = "memory://"
# CELERY_TASK_EAGER_PROPAGATES = True
# Celery worker configuration for production
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

# Celery settings, it is not recommended to change these
CELERY_WORKER_MAX_MEMORY_PER_CHILD = 200000
CELERY_BEAT_SCHEDULE_FILENAME = os.path.join(DATA_DIR, "celery", "beat-
↪schedule")
CELERY_TASK_ROUTES = {
    "weblate.trans.tasks.auto_translate*": {"queue": "translate"},
    "weblate.accounts.tasks.notify*": {"queue": "notify"},
    "weblate.accounts.tasks.send_mails": {"queue": "notify"},
    "weblate.utils.tasks.settings_backup": {"queue": "backup"},
    "weblate.utils.tasks.database_backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup_service": {"queue": "backup"},
    "weblate.memory.tasks.*": {"queue": "memory"},
}

# Enable plain database backups
DATABASE_BACKUP = "plain"

# Enable auto updating
AUTO_UPDATE = False

# PGP commits signing
WEBLATE_GPG_IDENTITY = None

# Third party services integration
MATOMO_SITE_ID = None
MATOMO_URL = None
GOOGLE_ANALYTICS_ID = None

```

(XXXXXXXXXXXX)

```
SENTRY_DSN = None
SENTRY_ENVIRONMENT = None
AKISMET_API_KEY = None
```

Management commands

!!!: Running management commands under a different user than the one running your webserver can result in files getting wrong permissions, please check [Permissions](#) [Permissions](#) for more details.

You will find basic management commands (available as `./manage.py` in the Django sources, or as an extended set in a script called **weblate** installable atop Weblate).

Invoking management commands

As mentioned before, invocation depends on how you installed Weblate.

If using virtualenv for Weblate, you can either specify the full path to **weblate**, or activate the virtualenv prior to invoking it:

```
# Direct invocation
~/weblate-env/bin/weblate

# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate
```

If you are using source code directly (either from a tarball or Git checkout), the management script is `./manage.py` available in the Weblate sources. To run it:

```
python ./manage.py list_versions
```

If you've installed Weblate using the pip or pip3 installer, or by using the `./setup.py` script, the **weblate** is installed to your path (or virtualenv path), from where you can use it to control Weblate:

```
weblate list_versions
```

For the Docker image, the script is installed like above, and you can run it using **docker exec**:

```
docker exec --user weblate <container> weblate list_versions
```

For **docker-compose** the process is similar, you just have to use **docker-compose exec**:

```
docker-compose exec --user weblate weblate weblate list_versions
```

In case you need to pass it a file, you can temporary add a volume:

```
docker-compose exec --user weblate /tmp:/tmp weblate weblate importusers /
↪tmp/users.json
```

!!!:

[Docker](#) [Installing on Debian and Ubuntu](#) [Installing on SUSE and openSUSE](#) [Installing on RedHat, Fedora and CentOS](#)

add_suggestions

weblate add_suggestions <project> <component> <language> <file>

?????? 2.5 ???.

Imports a translation from the file to use as a suggestion for the given translation. It skips duplicated translations; only different ones are added.

--author USER@EXAMPLE.COM

E-mail of author for the suggestions. This user has to exist prior to importing (you can create one in the admin interface if needed).

?:

```
weblate --author michal@cihar.com add_suggestions weblate application cs /  
↪tmp/suggestions-cs.po
```

auto_translate

weblate auto_translate <project> <component> <language>

?????? 2.5 ???.

?????? 4.6 ???: ??????????????????????

Performs automatic translation based on other component translations.

--source PROJECT/COMPONENT

Specifies the component to use as source available for translation. If not specified all components in the project are used.

--user USERNAME

Specify username listed as author of the translations. "Anonymous user" is used if not specified.

--overwrite

Whether to overwrite existing translations.

--inconsistent

Whether to overwrite existing translations that are inconsistent (see [??????](#)).

--add

Automatically add language if a given translation does not exist.

--mt MT

Use machine translation instead of other components as machine translations.

--threshold THRESHOLD

Similarity threshold for machine translation, defaults to 80.

--mode MODE

Specify translation mode, default is "translate" but "fuzzy" or "suggest" can be used.

?:

```
weblate auto_translate --user nijel --inconsistent --source weblate/  
↪application weblate website cs
```

?:

????

celery_queues

weblate celery_queues

?????? 3.7 ???.

Displays length of Celery task queues.

checkgit

weblate checkgit <project|project/component>

Prints current state of the back-end Git repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

commitgit

weblate commitgit <project|project/component>

Commits any possible pending changes to the back-end Git repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

commit_pending

weblate commit_pending <project|project/component>

Commits pending changes older than a given age.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

--age HOURS

Age in hours for committing. If not specified the value configured in *Component configuration* is used.

??: This is automatically performed in the background by Weblate, so there no real need to invoke this manually, besides forcing an earlier commit than specified by *Component configuration*.

??:

?????? ???? COMMIT_PENDING_HOURS

cleanuptrans

weblate cleanuptrans

Cleans up orphaned checks and translation suggestions. There is normally no need to run this manually, as the cleanups happen automatically in the background.

??:

?????? ?????

createadmin

weblate createadmin

Creates an `admin` account with a random password, unless it is specified.

--password PASSWORD

Provides a password on the command-line, to not generate a random one.

--no-password

Do not set password, this can be useful with `--update`.

--username USERNAME

Use the given name instead of `admin`.

--email USER@EXAMPLE.COM

Specify the admin e-mail address.

--name

Specify the admin name (visible).

--update

Update the existing user (you can use this to change passwords).

???? 2.9 ??: Added parameters `--username`, `--email`, `--name` and `--update`.

dump_memory

weblate dump_memory

2.20.2022.

Export a JSON file containing Weblate Translation Memory content.

??:

20220201 Weblate 2.20.2022

dumpuserdata

weblate dumpuserdata <file.json>

Dumps userdata to a file for later use by *importuserdata*

???: This comes in handy when migrating or merging Weblate instances.

import_demo

weblate import_demo

20220201 4.1 2022.

Creates a demo project with components based on <<https://github.com/WeblateOrg/demo>>.

This can be useful when developing Weblate.

import_json

weblate import_json <json-file>

20220201 2.7 2022.

Batch import of components based on JSON data.

The imported JSON file structure pretty much corresponds to the component object (see *GET /api/components/(string:project)/(string:component)/*). You have to include the name and filemask fields.

--project PROJECT

Specifies where the components will be imported from.

--main-component COMPONENT

Use the given VCS repository from this component for all of them.

--ignore

Skip (already) imported components.

--update

Update (already) imported components.

20220201 2.9 2022: The parameters **--ignore** and **--update** are there to deal with already imported components.

Example of JSON file:

```
[
  {
    "slug": "po",
    "name": "Gettext PO",
    "file_format": "po",
    "filemask": "po/*.po",
    "new_lang": "none"
  },
  {
    "name": "Android",
    "filemask": "android/values-*/strings.xml",
    "template": "android/values/strings.xml",
    "repo": "weblate://test/test",
```

(202202010202)

```

    "file_format": "aresource"
  }
]

```

??:

import_memory

import_memory

weblate import_memory <file>

XXXXXXXX 2.20 ???.

Imports a TMX or JSON file into the Weblate translation memory.

--language-map LANGMAP

Allows mapping languages in the TMX to the Weblate translation memory. The language codes are mapped after normalization usually done by Weblate.

--language-map en_US:en will for example import all en_US strings as en ones.

????TMX ?????????? Weblate ?????????????????????????????

??:

??????? Weblate ??????????

import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

???? 3.0 ???: The import_project command is now based on the ?????????? add-on, leading to some changes in behavior and what parameters are accepted.

Batch imports components into project based on filemask.

<project> names an existing project, into which the components are to be imported.

The <gitrepo> defines the Git repository URL to use, and <branch> signifies the Git branch. To import additional translation components from an existing Weblate component, use a *weblate://<project>/<component>* URL for the <gitrepo>.

The <filemask> defines file discovery for the repository. It can be either be made simple using wildcards, or it can use the full power of regular expressions.

The simple matching uses **** for component name and *** for language, for example: ***/*.*po*

The regular expression has to contain groups named *component* and *language*. For example: *(?P<language>[^\]*) / (?P<component>[^-/]*) \.po*

The import matches existing components based on files and adds the ones that do not exist. It does not change already existing ones.

--name-template TEMPLATE

Customize the name of a component using Django template syntax.

For example: Documentation: `{{ component }}`

--base-file-template TEMPLATE

Customize the base file for monolingual translations.

For example: `{{ component }}/res/values/string.xml`

--new-base-template TEMPLATE

Customize the base file for addition of new translations.

For example: `{{ component }}/ts/en.ts`

--file-format FORMAT

You can also specify the file format to use (see ??????????), the default is auto-detection.

--language-regex REGEX

You can specify language filtering (see *Component configuration*) with this parameter. It has to be a valid regular expression.

--main-component

You can specify which component will be chosen as the main one—the one actually containing the VCS repository.

--license NAME

Specify the overall, project or component translation license.

--license-url URL

Specify the URL where the translation license is to be found.

--vcs NAME

In case you need to specify which version control system to use, you can do it here. The default version control is Git.

To give you some examples, let's try importing two projects.

First The Debian Handbook translations, where each language has separate a folder with the translations of each chapter:

```
weblate import_project \
  debian-handbook \
  git://anonscm.debian.org/debian-handbook/debian-handbook.git \
  squeeze/master \
  '*/**.po'
```

Then the Tanaguru tool, where the file format needs be specified, along with the base file template, and how all components and translations are located in single folder:

```
weblate import_project \
  --file-format=properties \
  --base-file-template=web-app/tgol-web-app/src/main/resources/i18n/%s-
↪I18N.properties \
  tanaguru \
  https://github.com/Tanaguru/Tanaguru \
  master \
  web-app/tgol-web-app/src/main/resources/i18n/**-I18N_*.properties
```

More complex example of parsing of filenames to get the correct component and language out of a filename like src/security/Numerous_security_holes_in_0.10.1.de.po:

```
weblate import_project \
  tails \
  git://git.tails.boum.org/tails master \
  'wiki/src/security/(?P<component>.*).\.(?P<language>[^.]*).\po$'
```

Filtering only translations in a chosen language:

```
./manage import_project \
  --language-regex '^(cs|sk)$' \
  weblate \
  https://github.com/WeblateOrg/weblate.git \
  'weblate/locale/*/LC_MESSAGES/**/*.po'
```

Importing Sphinx documentation split to multiple files:

```
$ weblate import_project --name-template 'Documentation: %s' \
  --file-format po \
  project https://github.com/project/docs.git master \
  'docs/locale/*/LC_MESSAGES/**/*.po'
```

Importing Sphinx documentation split to multiple files and directories:

```
$ weblate import_project --name-template 'Directory 1: %s' \
  --file-format po \
  project https://github.com/project/docs.git master \
  'docs/locale/*/LC_MESSAGES/dir1/**/*.po'
$ weblate import_project --name-template 'Directory 2: %s' \
  --file-format po \
  project https://github.com/project/docs.git master \
  'docs/locale/*/LC_MESSAGES/dir2/**/*.po'
```

☐☐:

More detailed examples can be found in the starting chapter, alternatively you might want to use *import_json*.

importuserdata

weblate importuserdata <file.json>

Imports user data from a file created by *dumpuserdata*

importusers

weblate importusers --check <file.json>

Imports users from JSON dump of the Django auth_users database.

--check

With this option it will just check whether a given file can be imported and report possible conflicts arising from usernames or e-mails.

You can dump users from the existing Django installation using:

```
weblate dumpdata auth.User > users.json
```

install_addon

weblate 3.2

weblate install_addon --addon ADDON <project|project/component>

Installs an addon to a set of components.

--addon ADDON

Name of the addon to install. For example `weblate.gettext.customize`.

--configuration CONFIG

JSON encoded configuration of an addon.

--update

Update the existing addon configuration.

You can either define which project or component to install the addon in (for example `weblate/application`), or use `--all` to include all existing components.

To install *gettext* for all components:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width
↪": -1}' --update --all
```

weblate

weblate

list_languages

weblate list_languages <locale>

Lists supported languages in MediaWiki markup - language codes, English names and localized names.

This is used to generate `<https://wiki.110n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF>`.

list_translators

weblate list_translators <project|project/component>

Lists translators by contributed language for the given project:

```
[French]
Jean Dupont <jean.dupont@example.com>
[English]
John Doe <jd@example.com>
```

--language-code

List names by language code instead of language name.

You can either define which project or component to use (for example `weblate/application`), or use `--all` to list translators from all existing components.

list_versions

weblate list_versions

Lists all Weblate dependencies and their versions.

loadpo

weblate loadpo <project|project/component>

Reloads translations from disk (for example in case you have done some updates in the VCS repository).

--force

Force update, even if the files should be up-to-date.

--lang LANGUAGE

Limit processing to a single language.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

??: You seldom need to invoke this, Weblate will automatically load changed files for every VCS update. This is needed in case you manually changed an underlying Weblate VCS repository or in some special cases following an upgrade.

lock_translation

weblate lock_translation <project|project/component>

Prevents further translation of a component.

???: Useful in case you want to do some maintenance on the underlying repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

??:

`unlock_translation`

move_language

weblate move_language source target

????? 3.0 ???.

Allows you to merge language content. This is useful when updating to a new version which contains aliases for previously unknown languages that have been created with the (*generated*) suffix. It moves all content from the *source* language to the *target* one.

:

```
weblate move_language cze cs
```

After moving the content, you should check whether there is anything left (this is subject to race conditions when somebody updates the repository meanwhile) and remove the (*generated*) language.

pushgit

weblate pushgit <project|project/component>

Pushes committed changes to the upstream VCS repository.

--force-commit

Force commits any pending changes, prior to pushing.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

??: Weblate pushes changes automatically if `????????????????` in *Component configuration* is turned on, which is the default.

unlock_translation

weblate unlock_translation <project|project/component>

Unlocks a given component, making it available for translation.

???: Useful in case you want to do some maintenance on the underlying repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

??:

`lock_translation`

setupgroups

weblate setupgroups

Configures default groups and optionally assigns all users to that default group.

--no-privs-update

Turns off automatic updating of existing groups (only adds new ones).

--no-projects-update

Prevents automatic updates of groups for existing projects. This allows adding newly added groups to existing projects, see `????????????????`.

??:

`?????`

setuplang

weblate setuplang

Updates list of defined languages in Weblate.

--no-update

Turns off automatic updates of existing languages (only adds new ones).

updatechecks

weblate updatechecks <project|project/component>

Updates all checks for all strings.

???: Useful for upgrades which do major changes to checks.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

updategit

weblate updategit <project|project/component>

Fetches remote VCS repositories and updates the internal cache.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

??: Usually it is better to configure hooks in the repository to trigger [????](#), instead of regular polling by [updategit](#).

????

???? 4.0 ????: ???
??
??
??
Web ???
?????????????????Manage ?????? Post announcement ??????????????:

Webate

Dashboard

Projects

Languages

Checks

WebateOrg

translated 90%

Translations will be used only if they reach 60%.

ComponentsLanguagesInfoSearchInsightsFilesToolsManageShare

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 Webate 4.6

About Webate

Legal

Contact

Documentation

Donate to Webate

XXXXXXXXXXXXXXXXXXXXXXXXXXXX:

Message:

Translations will be used only if they reach 60%.

Project:

WebplateOrg

Info (light blue) ▼

Today |

☒ Notify users

SAVE

[illegible]

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

Choose all ⓘ

Chosen components ⓘ

Remove all

Hold down "Control", or "Command" on a Mac, to select more than one.

AUTOMATIC COMPONENT LIST ASSIGNMENTS

PROJECT REGULAR EXPRESSION ⓘ	COMPONENT REGULAR EXPRESSION ⓘ	DELETE? ⓘ
^.*\$	^.*\$	✕

+ Add another Automatic component list assignment

Save and add another

Save and continue editing

SAVE

Optional Weblate modules

Several optional modules are available for your setup.

Git exporter

2.10.0.

Provides you read-only access to the underlying Git repository using HTTP(S).

??

1. Add `weblate.gitexport` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.gitexport",)
```

2. Export existing repositories by migrating your database after installation:

```
weblate migrate
```

Usage

The module automatically hooks into Weblate and sets the exported repository URL in the *Component configuration*. The repositories are accessible under the `/git/` part of the Weblate URL, for example `https://example.org/git/weblate/main/`.

Repositories for publicly available projects can be cloned without authentication:

```
git clone 'https://example.org/git/weblate/main/'
```

Access to browse the repositories with restricted access (with *Private access control* or when `REQUIRE_LOGIN` is enabled) requires an API token which can be obtained in your *user profile*:

```
git clone 'https://user:KEY@example.org/git/weblate/main/'
```

???: By default members or *Users* group and anonymous user have access to the repositories for public projects via *Access repository* and *Power user* roles.

??

???? 2.4 ???.

This is used on [Hosted Weblate](#) to define billing plans, track invoices and usage limits.

??

1. Add `weblate.billing` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.billing",)
```

2. Run the database migration to optionally install additional database structures for the module:

```
weblate migrate
```

Usage

After installation you can control billing in the admin interface. Users with billing enabled will get new *Billing* tab in their *profile*.

The billing module additionally allows project admins to create new projects and components without being superusers (see *Adding translation projects and components*). This is possible when following conditions are met:

The billing is in its configured limits (any overusage results in blocking of project/component creation) and paid (if its price is non zero)

The user is admin of existing project with billing or user is owner of billing (the latter is necessary when creating new billing for users to be able to import new projects).

Upon project creation user is able to choose which billing should be charged for the project in case he has access to more of them.

????????

???? 2.15 ???.

This is used on [Hosted Weblate](#) to provide required legal documents. It comes provided with blank documents, and you are expected to fill out the following templates in the documents:

Terms of service document

Privacy policy document

Short overview of the terms of service and privacy policy

??: Legal documents for the Hosted Weblate service are available in this Git repository <<https://github.com/WeblateOrg/wllegal/tree/main/wllegal/templates/legal/documents>>.

Most likely these will not be directly usable to you, but might come in handy as a starting point if adjusted to meet your needs.

??

1. Add `weblate.legal` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.legal",)

# Optional:

# Social auth pipeline to confirm TOS upon registration/subsequent sign in
SOCIAL_AUTH_PIPELINE += ("weblate.legal.pipeline.tos_confirm",)

# Middleware to enforce TOS confirmation of signed in users
MIDDLEWARE += [
    "weblate.legal.middleware.RequireTOSMiddleware",
]
```

2.Run the database migration to optionally install additional database structures for the module:

```
weblate migrate
```

3.Edit the legal documents in the `weblate/legal/templates/legal/` folder to match your service.

Usage

After installation and editing, the legal documents are shown in the Weblate UI.

Avatars

Avatars are downloaded and cached server-side to reduce information leaks to the sites serving them by default. The built-in support for fetching avatars from e-mails addresses configured for it can be turned off using `ENABLE_AVATARS`.

Weblate currently supports:

Gravatar

Libravatar

??:

```
?????????????AVATAR_URL_PREFIX?ENABLE_AVATARS
```

Spam protection

You can protect against spamming by users by using the [Akismet](#) service.

- 1.Install the `akismet` Python module (this is already included in the official Docker image).
- 2.Obtain the Akismet API key.
- 3.Store it as `AKISMET_API_KEY` or `WEBLATE_AKISMET_API_KEY` in Docker.

Following content is sent to Akismet for checking:

```
????????????????????
```

```
????????????????????
```

?: This (among other things) relies on IP address of the client, please see `???` `?????????????` for properly configuring that.

?:

```
??? ?????.AKISMET_API_KEY?WEBLATE_AKISMET_API_KEY
```


`AUTH_LOCK_ATTEMPTS`

The settings can be also applied in the Docker container by adding `WEBLATE_` prefix to the setting name, for example `RATELIMIT_ATTEMPTS` becomes `WEBLATE_RATELIMIT_ATTEMPTS`.

API `API`

:

`API`

Fedora Messaging

Fedora Messaging is AMQP-based publisher for all changes happening in Weblate. You can hook additional services on changes happening in Weblate using this.

The Fedora Messaging integration is available as a separate Python module `weblate-fedora-messaging`. Please see <https://github.com/WeblateOrg/fedora_messaging/> for setup instructions.

Customizing Weblate

Extend and customize using Django and Python. Contribute your changes upstream so that everybody can benefit. This reduces your maintenance costs; code in Weblate is taken care of when changing internal interfaces or refactoring the code.

Neither internal interfaces nor templates are considered a stable API. Please review your own customizations for every upgrade, the interfaces or their semantics might change without notice.

:

`Weblate`

Creating a Python module

If you are not familiar with Python, you might want to look into [Python For Beginners](#), explaining the basics and pointing to further tutorials.

To write some custom Python code (called a module), a place to store it is needed, either in the system path (usually something like `/usr/lib/python3.7/site-packages/`) or in the Weblate directory, which is also added to the interpreter search path.

Better yet, turn your customization into a proper Python package:

1. Create a folder for your package (we will use `weblate_customization`).
2. Within it, create a `setup.py` file to describe the package:

```
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. Create a folder for the Python module (also called `weblate_customization`) for the customization code.
4. Within it, create a `__init__.py` file to make sure Python can import the module.
5. This package can now be installed using `pip install -e`. More info to be found in “[Editable](#)” [Installs](#).
6. Once installed, the module can be used in the Weblate configuration (for example `weblate_customization.checks.FooCheck`).

Your module structure should look like this:

```

weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py

```

You can find an example of customizing Weblate at <https://github.com/WeblateOrg/customize-example>, it covers all the topics described below.

Changing the logo

1. [Django](#) [Creating a Python module](#):

Branding appears in the following files:

`/weblate.svg` Logo shown in the navigation bar.

`logo-*.png` Web icons depending on screen resolution and web-browser.

`favicon.ico` Web icon used by legacy browsers.

`weblate-*.png` Avatars for bots or anonymous users. Some web-browsers use these as shortcut icons.

`mail-logo.png` Used in notifications e-mails.

2. Add it to `INSTALLED_APPS`:

```

INSTALLED_APPS = (
    # Add your customization as first
    "weblate_customization",
    # Weblate apps are here...
)

```

3. Run `weblate collectstatic --noinput`, to collect static files served to clients.

⚠:

[JavaScript](#) [CSS](#) [Translations](#)

[Translations](#)

[Translations](#) [Translations](#) [Translations](#) [Translations](#) Weblate [Translations](#):

1. Weblate [Python](#) [Creating a Python module](#):

2. Python [Translations](#) `WEBLATE_ADDONS` `CHECK_LIST` `AUTOFIX_LIST`:

```

# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)

# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)

# Addons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)

```

⚠:

[Translations](#) [Translations](#) [Translations](#) [Translations](#)

□□□□

[illegible]

Webplate

Dashboard

Projects

Languages

Checks

Manage

Tools

Billing

Webplate status

Backups

Translation memory

Performance report

SSH keys

Alerts

Repositories

Users

Appearance

Webplate support status

Webplate version

4.6 — 16021d9c710da36e157149d0f49a4b3547593f9c

Support status

Community support

Purchase support package

Donate to Webplate

Activate support package

The support packages include priority e-mail support, or cloud backups of your Webplate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate

Purchase support package

Powered by Webplate 4.6

About Webplate

Legal

Contact

Documentation

Donate to Webplate

Weblate :

🔗🔗🔗🔗🔗🔗🔗🔗: *Weblate* 🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗🔗

🔗🔗🔗🔗🔗🔗🔗🔗: *Weblate* 🔗🔗🔗🔗🔗🔗🔗🔗

□□□□□□□□□□: □□□□□

SSH [REDACTED]: *SSH [REDACTED]*

?????????????????: alerts

Django

??: ?????????????????????? Weblate ??????????

[illegible]

Site administration

REPORTS		
Weblate support status		
Status of repositories		
SSH keys		
Performance report		
Translation memory		
ACCOUNTS		
Audit logs	+ Add	✎ Change
Profiles	+ Add	✎ Change
Verified emails	+ Add	✎ Change
AUTH TOKEN		
Tokens	+ Add	✎ Change
AUTHENTICATION		
Groups	+ Add	✎ Change
Roles	+ Add	✎ Change
Users	+ Add	✎ Change
BILLING		
Billings	+ Add	✎ Change
Invoices	+ Add	✎ Change
Plans	+ Add	✎ Change
FONTS		
Font groups	+ Add	✎ Change
Fonts	+ Add	✎ Change
LEGAL		
Agreements	+ Add	✎ Change
PYTHON SOCIAL AUTH		
Associations	+ Add	✎ Change
Nonces	+ Add	✎ Change
User social auths	+ Add	✎ Change
SCREENSHOTS		
Screenshots	+ Add	✎ Change
TRANSLATION MEMORY		
Memorys	+ Add	✎ Change
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

Reports [View all reports](#) [View report for this project](#) [SSH](#) [Translations](#)
[Project configuration](#) [Component configuration](#)
Weblate languages [View all languages](#) [Add language](#)

[Add Project](#)

[Project configuration](#) [Component configuration](#)

Webate administration

WELCOME, **WEBLATE TEST** · [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

[Home](#) · [Weblate translations](#) · [Projects](#) · [Add Project](#)

Add Project

Required fields are marked in bold.

Project name:

Display name

URL slug:

Name used in URLs and filenames.

Project website:

Main website of translated project.

Translation instructions:

https://weblate.org/contribute/

You can use Markdown and mention users by @username.

☒ Set "Language-Team" header

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

☒ Use shared translation memory

Uses the pool of shared translations between projects.

☒ Contribute to shared translation memory

Contributes to the pool of shared translations between projects.

Access control:

Protected

How to restrict access to this project is detailed in the documentation.

☐ Enable reviews

Requires dedicated reviewers to approve translations.

☐ Enable source reviews

Requires dedicated reviewers to approve source strings.

☒ Enable hooks

Whether to allow updating this repository by remote hooks.

Language aliases:

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

Save and add another

Save and continue editing

SAVE

Project configuration

Webiate administration

WELCOME · WIBLATE TEST · RETURN TO WELCOME · DOCUMENTATION · CHANGE PASSWORD · LOG OUT

Home · Webiate translations · Components · Add component

Add Component

Required fields are marked in bold.

Component name:

Language names

Display name

URL slug:

language-names

Name used in URL, a-z and dashes.

Project:

WebiateCity

Version control system:

Git

Version control system you 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/WebiateOrg/demo.git

URL of a repository you webiate (or your) component to share it with other components

Repository push URL:

URL of a push-repository pointing to correct url if empty.

Repository browser:

https://github.com/WebiateOrg/demo/Mobile/branch/development

Link to repository browser use [branch] for branch, [fileview] and [diff] as filename and file placeholders.

Expected repository URL:

URL of repository where users can fetch changes from Webiate

Source string bug reporting address:

Email address for reports on errors in source strings. Leave empty for no emails.

Repository branch:

Repository branch to translate

Push branch:

Branch for pushing changes, leave empty to use repository branch

Flavorset:

webiate-language-flavorset-C_J_AREDACTED-10

Part of flavor to translate related to repository link, use " instead of language code, for example "part" or "flavorset-C_J_AREDACTED-10" or "part"

Monolingual base language file:

Filename of translation base file, containing all strings and their sources. It is recommended for monolingual translation formats.

Get base file

Whether users will be able to edit the base file for monolingual translations.

Intermediate language file:

Filename of intermediate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.

Template for new translations:

webiate-language-flavorset-language-part

Filename of the used for creating new translations. For gettext choose pot file.

File format:

gettext POT file

Locked

Whether component will not get any translation updates.

Allow translation propagation

Whether translation updates in other components will cause automatic translation in this one.

Turn on suggestions

Whether to allow translation suggestions or not.

Suggestion voting

Users can only vote for suggestions and can't make direct translations.

Autoaccept suggestions:

0

Automatically accept suggestions with this number of votes, use 0 to turn it off.

Translation flags:

Additional comment associated flags to enhance quality checks. Possible values can be found in the documentation.

Enforced checks:

List of checks which can not be ignored.

Translation license:

GNU General Public License v3.0 or later

Contributor agreement:

User agreement which needs to be approved before a user can translate this component.

Add new translation:

Create new language file

New translation requests for creating new translations.

Language code style:

Default based on file format

Custom or language code used to generate the filename for translations created by Webiate.

Store strings

Whether adding and removing strings happens from Webiate. If your strings are extracted from the source code or managed externally you probably want to keep it disabled.

Merge style:

Default

Before another Webiate should merge the upstream repository or release changes errors.

Control message when translating:

Translated using Webiate {{ language_name }}
Currently translated at {{ state.translated.percent }}% ({{ state.translated }} of {{ state.all }} strings)
Translation {{ project_name }}{{ component_name }}
Translation URL: {{ url }}

You can use template language for more info, please consult the documentation for more details.

Control message when adding translation:

Add translation using Webiate {{ language_name }}

You can use template language for more info, please consult the documentation for more details.

Control message when removing translation:

Deleted translation using Webiate {{ language_name }}

You can use template language for more info, please consult the documentation for more details.

Control message when merging translation:

Merge branch {{ component_name_branch }} into Webiate.

You can use template language for more info, please consult the documentation for more details.

Control message when adding a change:

Update translation files
Updated by {{ add_name }}'s fork in Webiate.
Translation {{ project_name }}{{ component_name }}
Translation URL: {{ url }}

You can use template language for more info, please consult the documentation for more details.

Push on commit

Whether the repository should be pushed upstream on every commit.

Age of changes to commit:

24

How many hours after which any pending changes will be submitted to the VCS.

Lock on error

Whether the component should be locked on repository errors.

Source language:

English

Language used for source strings in all components.

Language filter:

Translated

Regular expression used to filter translation files when scanning for flavorset.

Variant regular expression:

Regular expression used to determine variants of a string.

Priority:

Medium

Components with higher priority are offered first to translations.

Restricted component

Personal access to this component is only given through explicit given permission.

Share in projects:

WebiateCity

Use as a glossary

Choose additional projects where this component will be listed. Hold down "Control" or "Command" on a Mac, to select more than one.

Glossary color:

Silver

Save and add another

Save and update settings

Save

??:

Component configuration?Bilingual and monolingual formats

11

XXXXXXXXXXXXXXXXXXXX ID XX
XXXXXXXXXXXXXXXXXXXX *Component configuration* XXX:

Webalate administration

WELCOME WEBALATE TEAM | RETURN TO WEBALATE | DOCUMENTATION | CHANGE PASSWORD | LOG OUT

Home | Webalate translations | Components | Add component

ADD NEW COMPONENT

Add Component

Required fields are marked as **bold**.

Component name:

Android

URL slug:

android

Project:

WebalateOrg

Version control system:

Git

Source code repository:

webalate (/webalateorg/language-names)

Repository push URL:

Repository browser:

Expected repository URL:

Source stringing bug reporting address:

Repository branch:

Push branch:

Flormask:

app/src/main/res/values-strings.xml and

Monolingual base language file:

app/src/main/res/values-strings.xml

Use base file

Whether users will be able to edit the base file for monolingual translations.

Intermediate language file:

Template for new translations:

File format:

Android String Resource

Locked

Locked component will not get any translation updates.

Allow translation propagation

Whether translation updates in other components will cause automatic translation in this one.

Turn on suggestions

Whether to allow translation suggestions at all.

Suggestion editing

Users can only vote for suggestions and can't make direct translations.

Autosuggest suggestions:

0

Translation flags:

Enforced checks:

Translation license:

MIT License

Contribution agreement:

Adding new translation:

Choose new language file

Language code style:

Default based on the file format

Use string strings

Whether adding and removing string groups from Webalate. If your strings are extracted from the source code or managed externally you probably want to keep it disabled.

Merge style:

Rebase

Current message when handling:

Translated using Webalate ({ language_name })
Currently translated at { data.translated.percent }% ({ data.translated } of { data.all } strings)
Translation { project_name }({ component_name })
Translate URL: { url }

Current message when adding translation:

Add translation using Webalate ({ language_name })

Current message when removing translation:

Deleted translation using Webalate ({ language_name })

Current message when merging translation:

Merge branch { component_branch } into Webalate.

Current message when adding makes a change:

Update translation file
Updated by { address_name }({ track in Webalate)
Translation { project_name }({ component_name })
Translate URL: { url }

Push on commit

Whether new requests should be pushed (approved) or only commit.

Age of changes to commit:

24

Lock on error

Whether the component should be locked on repository errors.

Source language:

English

Language filter:

all

Variables regular expression:

Priority:

Medium

Restricted component

Restrict access to the component to only those explicitly given permission.

Share in projects:

WebalateOrg

Use as a glossary

Glossary color:

Blue

Lock and add another

Save and continue editing

Save

Component configuration?Bilingual and monolingual formats

Weblate

Weblate [REDACTED] Libre [REDACTED]
[REDACTED]<<https://weblate.org/support/>>
[REDACTED]

□□□□□□

□□□□□ 3.8 □□□.

Webplate

W Weblate

DashboardProjects ▾Languages ▾Checks ▾

⚙️ Manage

Weblate status

Backups

Translation memory

Performance report

SSH keys

Alerts

Repositories

Users

Appearance

Tools

Billing

Weblate support status

ⓘ

Weblate version

4.6 — 16021d9c710da36e157149d0f49a4b3547593f9c

Support status

Community support

Purchase support package

Donate to Weblate

Activate support package

ⓘ

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate

Purchase support package

Powered by Weblate 4.6

About Weblate

Legal

Contact

Documentation

Donate to Weblate

Weblate

Weblate URL

🔗🔗🔗🔗🔗 Weblate 🔗🔗🔗🔗🔗

Weblate
 SSH

Additionally, when *Weblate* [\[???](#) is turned on:

List of public projects (name, URL and website)

????????????????

??????

???? ?????????????????????

Weblate ?????????? ??????

Weblate ???

???: ??????? ???

Weblate ???

????? 4.5.2 ???.

?: ?????????????????????

Discover Weblate is an opt-in service that makes it easier for users to find Weblate servers and communities. Users can browse registered services on <<https://weblate.org/discover/>>, and find there projects to contribute.

?????????

?: Participating in Discover Weblate makes Weblate submit some information about your server, please see [Weblate ?????](#).

To list your server with an active support subscription (see [????????](#)) in Discover Weblate all you need to do is turn this on in the management panel:

The screenshot shows the Weblate management interface. At the top is a navigation bar with '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 'Appearance'. The 'Weblate status' tab is active, showing 'Tools' and 'Billing' sub-tabs. The main content area displays the 'Weblate support status' with the following details:

- Weblate version:** 4.6 — 16021d9c710da36e157149d0f49a4b3547593f9c
- Support status:** Community support
- Discover Weblate:** Your Weblate is not listed on weblate.org. A button 'Enable discovery' is present, along with a link 'Browse discovery'.

Below the status section are three buttons: 'Manage support package', 'Purchase support package', and 'Donate to Weblate'. The 'Activate support package' section follows, containing the text: 'The support packages include priority e-mail support, or cloud backups of your Weblate installation.' It includes an 'Activation token' input field and a note: 'Please enter the activation token obtained when making the subscription.' At the bottom of this section are 'Activate' and 'Purchase support package' buttons.

Listing your server without a support subscription in Discover Weblate:

- 1.<<https://weblate.org/user/>> ??????????
- 2.Weblate ?????????? <<https://weblate.org/subscription/discovery/>> ?????

3. Confirm the service activation in your Weblate and turn on the discovery listing in your Weblate management page using *Enable discovery* button:

W Weblate

DashboardProjectsLanguagesChecks

Manage

Weblate status

BackupsTranslation memoryPerformance reportSSH keysAlertsRepositoriesUsersAppearance

ToolsBilling

Weblate support status

Weblate version4.6 — 16021d9c710da36e157149d0f49a4b3547593f9c

Support statusCommunity support

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



You can customize the listing by providing a text and image (570 x 260 pixels) at <<https://weblate.org/user/>>.

Legal documents

??: Herein you will find various legal information you might need to operate Weblate in certain legal jurisdictions. It is provided as a means of guidance, without any warranty of accuracy or correctness. It is ultimately your responsibility to ensure that your use of Weblate complies with all applicable laws and regulations.

ITAR and other export controls

Weblate can be run within your own datacenter or virtual private cloud. As such, it can be used to store ITAR or other export-controlled information, however, end users are responsible for ensuring such compliance.

The Hosted Weblate service has not been audited for compliance with ITAR or other export controls, and does not currently offer the ability to restrict translations access by country.

US encryption controls

Weblate does not contain any cryptographic code, but might be subject export controls as it uses third party components utilizing cryptography for authentication, data-integrity and -confidentiality.

Most likely Weblate would be classified as ECCN 5D002 or 5D992 and, as publicly available libre software, it should not be subject to EAR (see [Encryption items NOT Subject to the EAR](#)).

Software components used by Weblate (listing only components related to cryptographic function):

See https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is_Python_subject_to_export_laws.3F

Optionally used by Weblate

Optionally used by Weblate

Used by Git

Used by Python and cURL

The strength of encryption keys depends on the configuration of Weblate and the third party components it interacts with, but in any decent setup it will include all export restricted cryptographic functions:

In excess of 56 bits for a symmetric algorithm

Factorisation of integers in excess of 512 bits for an asymmetric algorithm

Computation of discrete logarithms in a multiplicative group of a finite field of size greater than 512 bits for an asymmetric algorithm

Discrete logarithms in a group different than above in excess of 112 bits for an asymmetric algorithm

Weblate doesn't have any cryptographic activation feature, but it can be configured in a way where no cryptography code would be involved. The cryptographic features include:

Accessing remote servers using secure protocols (HTTPS)

Generating signatures for code commits (PGP)

??:

Export Controls (EAR) on Open Source Software

□□□□□□□□

Weblate ???

Webplate

Weblate [\[?\]\[?\]\[?\]\[?\]\[?\]](#)

Weblate [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)

Weblate [???]

Weblate[?][?][?][?][?][?][?][?][?]

Weblate [???]

Weblate [?][?][?][?][?][?]

Weblate ???

Weblate Weblate Weblate Weblate Weblate

Weblate Weblate Weblate Weblate Weblate

If you find a possible mistake in the source string, you can mark it with a comment in the Weblate editor. This way, it can be discussed and corrected. If you're certain, you can also click on the link in the *Source string location* section and submit a PR with your correction.

Weblate

You are welcome to improve the documentation page of your choice. Do it easily by clicking the *Edit on GitHub* button in the top-right corner of the page.

Please respect these guidelines while writing:

1. Don't remove part of the documentation if it's valid.
2. Use clear and easily-understandable language. You are writing tech docs, not a poem. Not all docs readers are native speakers, be thoughtful.
3. Don't be afraid to ask if you are not certain. If you have to ask about some feature while editing, don't change its docs before you have the answer. This means: You change or ask. Don't do both at the same time.
4. Verify your changes by performing described actions while following the docs.
5. Send PR with changes in small chunks to make it easier and quicker to review and merge.
6. If you want to rewrite and change the structure of a big article, do it in two steps:
 1. Rewrite
 2. Once the rewrite is reviewed, polished, and merged, change the structure of the paragraphs in another PR.

Tip: You can [translate the docs](#).

Weblate

If you have an idea and not sure if it's suitable for an issue, don't worry. You can join the community in [GitHub discussions](#).

Weblate

Weblate [donate page](#) [Libre](#) [Weblate](#) [2022/2023](#)

Weblate

Weblate:

Yashiro Ccs

Cheng-Chia Tseng

Timon Reinhard

Cassidy James

Loic Dachary

Marozed

<https://freedombox.org/>

GNU Solidario (GNU Health)

BallotReady

Richard Nespithal

[Weblate](#)

Weblate

Weblate                          

11

good first issue Weblate

Weblate

Weblate ██████████ ██████████ Weblate ████████ virtualenv ██████████

1. Weblate

```
git clone https://github.com/WebateOrg/webate.git
cd webate
```

2.virtualenv

```
virtualenv .venv
.venv/bin/activate
```

3. Weblate :

```
pip install -e .
```

3.

```
pip install -r requirements-dev.txt
```

4. :

```
weblate runserver
```

5.Celery Worker 11111111111111:

```
./weblate/examples/celery start
```

6. Local testing:

```
. scripts/test-database
./manage.py test
```

??:

????????????

Docker????? Weblate ???

Docker ? docker-compose ???

```
./rundevel.sh
```

```
### Docker #####Weblate 2 <http://127.0.0.1:8080/> #####admin ##### admin
#####Adding translation projects and components #####
```

```
Dockerfile docker-compose.yml dev-docker
```

```
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? test ? ? ? ? ? ? ? ? ? ? ? ? ? ? test ? ? ? ? ? ? ? ? ? ? ? ? ? ? :
```

```
./rundevel.sh test --failfast weblate.trans
```

[illegible]

□□□□□□□□

```
./rundevel.sh logs
```

XXXXXXXXXXXXXXXXXXXX:

```
./rundev.sh stop
```

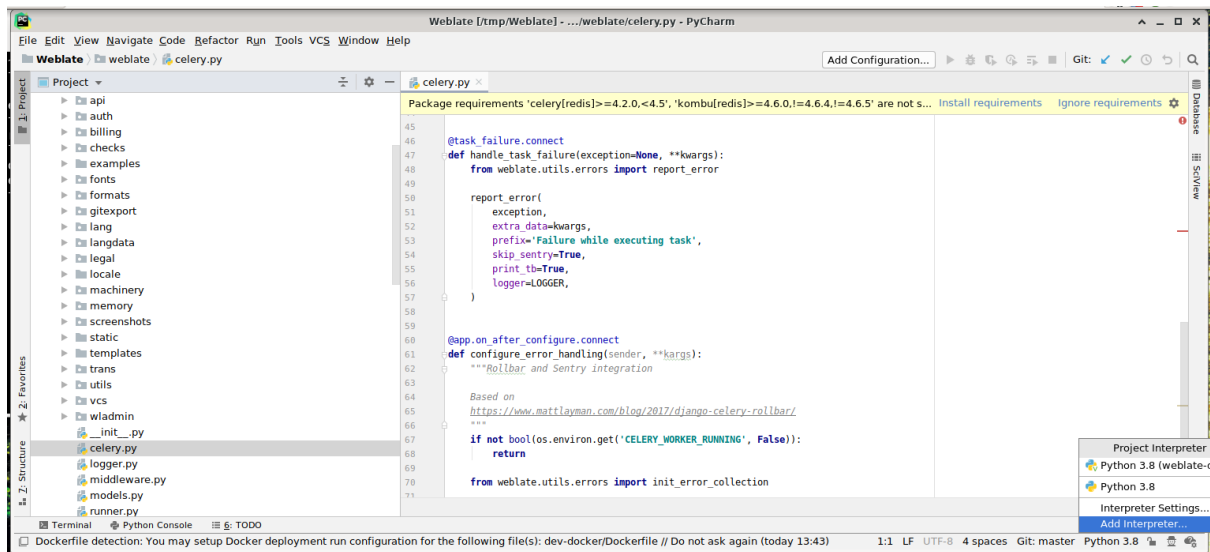
```
##### Docker #####
```

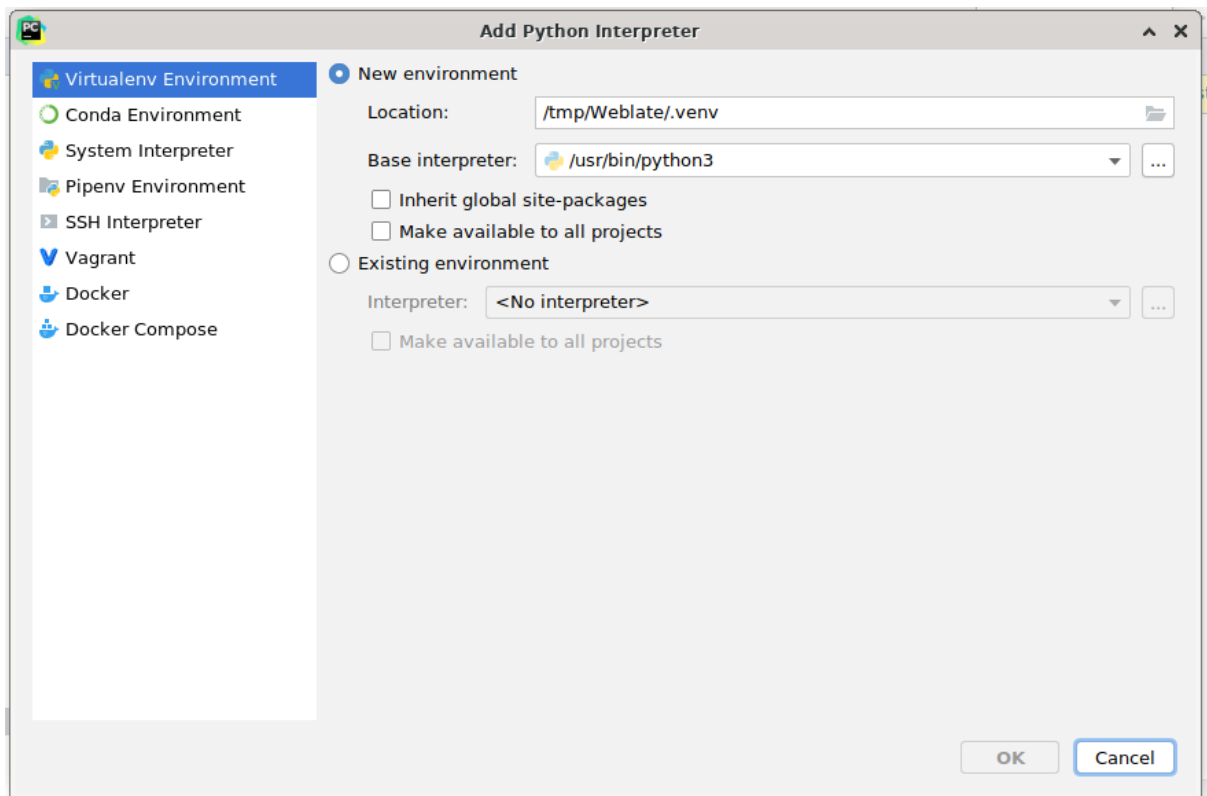
??: ???

PyCharm Weblate

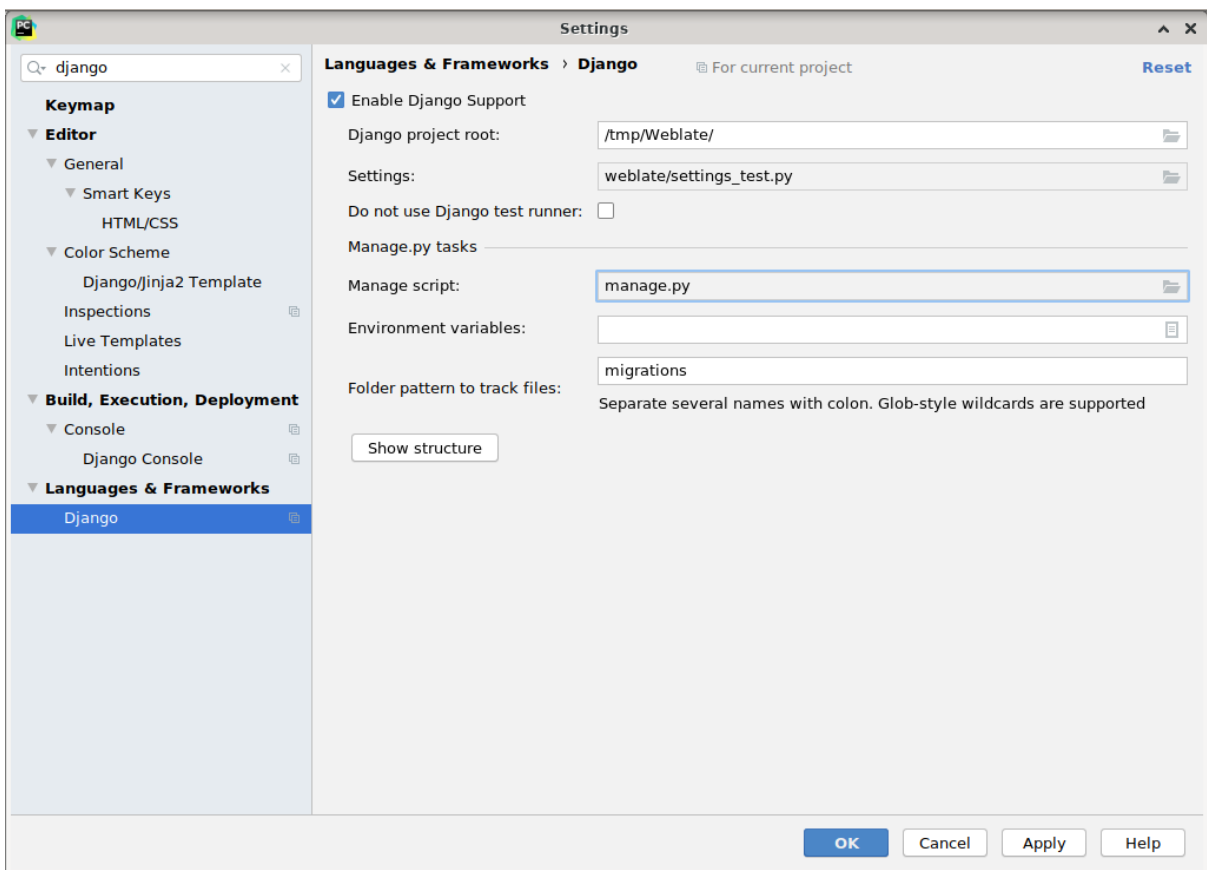
PyCharm ? Python ? IDE ?????????? Weblate ??????????

GitHub  PyCharm  IDE :

[illegible]



2. In the PyCharm IDE, create a new virtualenv environment for the Django project. To do this, go to **File > Settings > Languages & Frameworks > Django**. In the **Django** settings, check **Enable Django Support**. Set the **Django project root** to `/tmp/Webplate/` and the **Settings** to `webplate/settings_test.py. Set the Manage script to manage.py and the Folder pattern to track files to migrations. Click OK.`



3. In the Django project, create a new Django application. To do this, go to **File > New > Django Application**. In the **Create Django Application** dialog, set the **Application name** to `webplate` and the **Package name** to `webplate`. Click **OK**.

The screenshot shows the 'Run/Debug Configurations' dialog in PyCharm. The 'Configuration' tab is selected, and the configuration is named 'Unnamed'. The settings are as follows:

- Name:** Unnamed
- Host:** localhost
- Port:** 8000
- Additional options:** (empty)
- Run browser:** ☐ (unchecked)
- Start JavaScript debugger automatically when debugging:** ☐ (unchecked)
- Custom run command:** (empty)
- Test server:** ☐ (unchecked)
- No reload:** ☒ (checked)
- Environment:**
 - Environment variables:** IONUNBUFFERED=1;DJANGO_SETTINGS_MODULE=weblate.settings_test
 - Python interpreter:** Project Default (Python 3.8 (weblate)) ~/Github/weblate/.venv/bin/python
 - Interpreter options:** (empty)
 - Working directory:** (empty)
 - Add content roots to PYTHONPATH:** ☐ (unchecked)
 - Add source roots to PYTHONPATH:** ☒ (checked)
- Before launch:** Activate tool window

The 'Before launch' section is expanded, showing 'Activate tool window'. The 'OK' button is highlighted in blue.

Not processing background tasks

Lot of things happen in background Celery workers. In case things like sending out e-mails or component removal does not work, there might be some issue with it.

?????????:

Check Celery process is running, see [Celery](#) [????????????????????](#) [???](#)

Check Celery queue status either in [????](#) or using `celery_queues`

Look into Celery logs for errors (see [Weblate logs](#))

Not receiving e-mails from Weblate

You can verify whether outgoing e-mail is working correctly by using the `sendtestemail` management command (see [Invoking management commands](#) for instructions on how to invoke it in different environments) or using [????](#) under the *Tools* tab.

These send e-mail directly, so this verifies that your SMTP configuration is correct (see [????????](#)). Most of the e-mails from Weblate are however sent in the background and there might be some issues with Celery involved as well, please see [Not processing background tasks](#) for debugging that.

Analyzing application crashes

In case the application crashes, it is useful to collect as much info about the crash as possible. The easiest way to achieve this is by using third-party services which can collect such info automatically. You can find info on how to set this up in [??? ?????](#).

Silent failures

Lots of tasks are offloaded to Celery for background processing. Failures are not shown in the user interface, but appear in the Celery logs. Configuring [??? ?????](#) helps you to notice such failures easier.

Performance issues

In case Weblate performs badly in some situation, please collect the relevant logs showing the issue, and anything that might help figuring out where the code might be improved.

In case some requests take too long without any indication, you might want to install `dogslow` along with [????](#) [????](#) and get pinpointed and detailed tracebacks in the error collection tool.

Weblate ???

?: ?????Weblate ?????????????????

Weblate ?????????????? Django ?????????????

????????

Weblate ??? ?????????????????:

???????????????? Sphinx ?????????????

???????????? Docker ??????: [Docker](#) [????](#) Weblate [????](#)

Django<<https://www.djangoproject.com/>> _ ????????? Weblate ??? ??: [Weblate](#) [????](#)

?????? CSSJavaScript ??????: [Weblate](#) [????](#)

□□□□□

Weblate [?] Django [?]: *Optional Weblate modules* [?]

accounts

[illegible]

addons

api

Django REST framework API

auth

□□□□□□□□

billing

□□□□□□ □□ □□□□□□

checks

□□□□□□ *□□□□* □□□□□□

fonts

□□□□□□□□□□□□□□

```
formats
```

gitexport

Git exporter

lang

????????????????

legal

□□□□□□ *□□□□□□□□* □□□□□□

machinery

????????????

memory

?????????????: ???????

screenshots

????????? OCR ?????

trans

□□□□□□□□ □□□□□□

utils

□□□□□□□ □□□□□□□□

VCS

????????????????

wladmin

???????? Django ?????????

???????

???? Weblate ?????????????????????????????????

class weblate.addons.base.**BaseAddon** (storage=None)

classmethod **can_install** (component, user)

??

configure (settings)

??????????

daily (component)

??????????

classmethod **get_add_form** (user, component, **kwargs)

????????????????????????????????????

get_settings_form (user, **kwargs)

????????????????????????

post_add (translation)

????????????????????????????

post_commit (component)

????????????????????????????????

post_push (component)

?????? upstream ??????????????????????

post_update (component, previous_head: str, skip_push: bool)

?????? upstream ??????????????????????

??????

previous_head (str) -- ?????????? HEAD????????????????

skip_push (bool) -- ??????? upstream ??? commit_and_push

??? commit_pending ??????????????????????

pre_commit (translation, author)

????????????????????????????????

pre_push (component)

?????? upstream ??????????????????????

pre_update (component)

?????? upstream ??????????????????????

save_state ()

????????????????

stay_on_create = False

Weblate ??????????

store_post_load (translation, store)

????????????????????????

It receives an instance of a file format class as a argument.

This is useful to modify file format class parameters, for example adjust how the file will be saved.

unit_pre_create (unit)

????????????????????????

??????:

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
```

(????????)

(???????????)

```
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#

from django.utils.translation import gettext_lazy as _

from weblate.addons.base import BaseAddon
from weblate.addons.events import EVENT_PRE_COMMIT

class ExampleAddon(BaseAddon):
    # Filter for compatible components, every key is
    # matched against property of component
    compat = {"file_format": {"po", "po-mono"}}
    # List of events addon should receive
    events = (EVENT_PRE_COMMIT,)
    # Addon unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example addon")
    # Detailed addon description
    description = _("This addon does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return
```

Weblate ????????

???????????? BootstrapjQuery????????? ???

????????

Weblate supports the latest, stable releases of all major browsers and platforms.

Alternative browsers which use the latest version of WebKit, Blink, or Gecko, whether directly or via the platform's web view API, are not explicitly supported. However, Weblate should (in most cases) display and function correctly in these browsers as well.

Older browsers might work, but some features might be limited.

?????????

```
yarn      ???? ???? ?????????? ????????????????????????????????????????????????????????????? scripts/yarn
????????????????????????????????????????????? scripts/yarn-update ?????????????????????? weblate/static/
vendor ???? ?????????????????????????????????????????????????????????????????????????????????????????????????
```

Adding new third-party library typically consists of:

```
# 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 .
```

Getting started

Weblate [JavaScript](#) [CSS](#) [Prettier](#) [ESLint](#) [JavaScript](#)

Usage

gettext

```
document.write(gettext('this is to be translated'));

var object_count = 1 // or 0, or 2, or 3, ...
s = gettext('literal for the singular case',
            'literal for the plural case', object_count);

fmts = gettext('There is %s object. Remaining: %s',
               'There are %s objects. Remaining: %s', 11);
s = interpolate(fmts, [11, 20]);
// s is 'There are 11 objects. Remaining: 20'
```

22:

Translation topic in the Django documentation

Resources

Weblate [Material Design Icons](#) [Material Design Resources](#)

HTML [SVG](#) [:file:'scripts/optimize-svg'](#)

Weblate

Weblate [GitHub](#)

Weblate

Weblate

If you are not sure about your bug report or feature request, you can try [Weblate](#).

Security

In order to give the community time to respond and upgrade, you are strongly urged to report all security issues privately. HackerOne is used to handle security issues, and can be reported directly at [HackerOne](#). Once you submit it there, community has limited but enough time to solve the incident.

Alternatively, report to security@weblate.org, which ends up on HackerOne as well.

If you don't want to use HackerOne, for whatever reason, you can send the report by e-mail to michal@cihar.com. You can choose to encrypt it using this PGP key [3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D](#). You can also get the PGP key from [Keybase](#).

22: Weblate depends on third-party components for many things. In case you find a vulnerability affecting one of those components in general, please report it directly to the respective project.

Some of these are:

[Django](#)

[Django REST framework](#)

[Python Social Auth](#)

Weblate testsuite and continuous integration

Testsuites exist for most of the current code, increase coverage by adding testcases for any new functionality, and verify that it works.

Continuous integration

Current test results can be found on [GitHub Actions](#) and coverage is reported on [Codecov](#).

There are several jobs to verify different aspects:

Unit tests

Documentation build and external links

Migration testing from all supported releases

Code linting

Setup verification (ensures that generated dist files do not miss anything and can be tested)

The configuration for the CI is in `.github/workflows` directory. It heavily uses helper scripts stored in `ci` directory. The scripts can be also executed manually, but they require several environment variables, mostly defining Django settings file to use and database connection. The example definition of that is in `scripts/test-database`:

```
# Simple way to configure test database from environment

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=${1:-postgresql}

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

The simple execution can look like:

```
. scripts/test-database
./ci/run-migrate
./ci/run-test
./ci/run-docs
```

Local testing

To run a testsuite locally, use:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

!!!: You will need a database (PostgreSQL) server to be used for tests. By default Django creates separate database to run tests with `test_` prefix, so in case your settings is configured to use weblate, the tests will use `test_weblate` database. See [Weblate !!!!!!!!!](#) for setup instructions.

The `weblate/settings_test.py` is used in CI environment as well (see [Continuous integration](#)) and can be tuned using environment variables:

```
# Simple way to configure test database from environment

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=${1:-postgresql}

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1
```

(!!!!!!!!)

```
# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

Prior to running tests you should collect static files as some tests rely on them being present:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

You can also specify individual tests to run:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.
↪gitexport
```

????: The tests can also be executed inside developer docker container, see [Docker??????? Weblate ????](#).

??:

See [Django ????????](#) for more info on running and writing tests for Django.

??? ?????

Weblate [JSON Schema](#) to [???????? JSON ?????????????????????](#)

Weblate ????????

<https://weblate.org/schemas/weblate-memory.schema.json>

```
?
??
??
??
??
category
1 = ??????2 = ???10000000 ?? =
????????20000000 ?? = ?????
?
?:
??
0
1
origin
????
????????
?
?:
test.tmx
????/????
source
????
??
?
?:
Hello
minLength
1
source_language
????
ISO 639-1 / ISO 639-2 / IETF BCP 47
?
?:
en
^[\ ]+$
target
????
?
?:
Ahoj
minLength
1
target_language
????
ISO 639-1 / ISO 639-2 / IETF BCP 47
?
????
```

Table 7 – `dump_memory`

	<code>dump_memory</code>	<code>import_memory</code>	
	<code>False</code>		

`cs`

`^[^ ]+$`

`dump_memory`

Weblate `weblate-userdata.schema.json`

<https://weblate.org/schemas/weblate-userdata.schema.json>

`weblate-userdata.schema.json`

`basic`

`username`

`full_name`

`email`

`date_joined`

`profile`

`language`

`suggested`

`translated`

`uploaded`

`hide_completed`

`secondary_in_zen`

`hide_source_secondary`

`Username`

`full_name`

`email`

`date_joined`

`language`

`suggested`

`translated`

`uploaded`

`hide_completed`

`secondary_in_zen`

`hide_source_secondary`

`cs`

`^[^ ]+$`

`cs`

`^[^ ]+$`

`cs`

`^[^ ]+$`

`cs`

`^[^ ]+$`

`cs`

`^[^ ]+$`

`cs`

`^[^ ]+$`

`cs`

`^[^ ]+$`

`cs`

`^[^ ]+$`

`cs`

`^[^ ]+$`

`boolean`

`False`

`True`

`boolean`

`True`

`True`

`^[^ ]+$`

`^[^ ]+$`

Table 8 – `XXXXXXXXXX`

		<code>?</code>	<code>boolean</code>	
		<code>?:</code>	<code>False</code>	
		<code>?????</code>	<code>True</code>	
	<code>editor_link</code>	<code>???? ???</code>		
		<code>?</code>	<code>???</code>	
		<code>?:</code>		
		<code>?????</code>	<code>^.*\$</code>	
		<code>?????</code>		
	<code>translate_mode</code>	<code>??????? ???</code>		
		<code>?</code>	<code>??</code>	
		<code>?:</code>	<code>0</code>	
		<code>?????</code>	<code>0</code>	
	<code>zen_mode</code>	<code>Zen ?????</code>		
		<code>?</code>	<code>??</code>	
		<code>?:</code>	<code>0</code>	
		<code>?????</code>	<code>0</code>	
	<code>special_chars</code>	<code>?????</code>		
		<code>?</code>	<code>???</code>	
		<code>?:</code>		
		<code>?????</code>	<code>^.*\$</code>	
		<code>?????</code>		
	<code>dashboard_view</code>	<code>????????????????</code>		
		<code>?</code>	<code>??</code>	
		<code>?:</code>	<code>1</code>	
		<code>?????</code>	<code>0</code>	
	<code>dash-board_component_list</code>	<code>???????????????? ???</code>		
		<code>?????</code>	<code>null</code>	
		<code>?????</code>	<code>?</code>	<code>null</code>
		<code>?</code>	<code>?</code>	<code>??</code>
	<code>languages</code>	<code>?????</code>		
		<code>?</code>	<code>??</code>	
		<code>?????</code>		
		<code>??</code>	<code>?????</code>	
			<code>?</code>	<code>????</code>
			<code>?:</code>	<code>CS</code>
			<code>?????</code>	<code>^.*\$</code>
			<code>?????</code>	
	<code>secondary_languages</code>	<code>?????</code>		
		<code>?</code>	<code>??</code>	
		<code>?????</code>		
		<code>??</code>	<code>?????</code>	
			<code>?</code>	<code>????</code>
			<code>?:</code>	<code>sk</code>
			<code>?????</code>	<code>^.*\$</code>
			<code>?????</code>	
	<code>watched</code>	<code>????????????</code>		
		<code>?</code>	<code>??</code>	
		<code>?????</code>		
		<code>??</code>	<code>????????????</code>	
			<code>?</code>	<code>????</code>
			<code>?:</code>	<code>weblate</code>
			<code>?????</code>	<code>^.*\$</code>
			<code>?????</code>	
<code>auditlog</code>		<code>?????</code>		
		<code>?</code>	<code>??</code>	
		<code>?????</code>		
		<code>??</code>		
		<code>??</code>		
		<code>?</code>	<code>??????</code>	
		<code>?????</code>		
	<code>address</code>		<code>IP ?????</code>	
			<code>?</code>	<code>????</code>
			<code>?:</code>	<code>127.0.0.1</code>
			<code>?????</code>	<code>^.*\$</code>
			<code>?????</code>	
	<code>user_agent</code>		<code>???? ?????</code>	

`XXXXXXXXXX`


```
./scripts/create-release XXXXXXXXXXXXXXXXXXXXXXXXXXXX:
XXXXXXXXXXXXXXXXXXXXXXXXXXXX GnuPG
Weblate git XXXXXXXXXX push XXXXXXXXXX push
hub XXXXXXXXXX Weblate XXXXXXXXXXXXXXXXXXXX
Weblate XXXXXXXXXX XXXXXXXXXX SSH XXXXWeb XXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
XXXXXXXXXXXXXXXXXXXX
```

```
XXXX: Weblate XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
Weblate XXXXXX Linux Foundation Core Infrastructure Initiative XXXXXXXXXXXX <https://bestpractices.coreinfrastructure.org
XXXXXXXXXXXX
```

```
XX:
```

```
XXXXXXXXXXXX
```

Tracking dependencies for vulnerabilities

Security issues in our dependencies are monitored using [Dependabot](#). This covers the Python and JavaScript libraries, and the latest stable release has its dependencies updated to avoid vulnerabilities.

```
XXXX: There might be vulnerabilities in third-party libraries which do not affect Weblate, so those are not addressed
by releasing bugfix versions of Weblate.
```

Docker container security

The Docker containers are scanned using [Anchore](#) and [Trivy](#).

This allows us to detect vulnerabilities early and release improvements quickly.

You can get the results of these scans at GitHub — they are stored as artifacts on our CI in the SARIF format (Static Analysis Results Interchange Format).

```
XX:
```

```
Continuous integration
```

Weblate **XXXX**

```
XXXXXXXXXXXX
```

```
XXX XXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXX Web XXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
XXXXXXXXXX
```

```
"Weblate" XX"web" X "translate" XXXXXXXXXXXXXXXX
```

Web

<https://weblate.org> <https://hosted.weblate.org> <https://docs.weblate.org>

<https://github.com/WeblateOrg/graphics/>

Michal Čihář <michal@cihar.com>

Weblate Michal Čihář <michal@cihar.com> 2012

Copyright (C) 2012 - 2021 Michal Čihář <michal@cihar.com>

: FreeSoftware Foundation GNU General Public License
3

GNU
General Public License

GNU General Public License <<https://www.gnu.org/licenses/>>

Weblate 4.6.2

Released on May 8th 2021.

Fixed crash after moving shared component between projects.

Fixed adding new strings to empty properties files.

Fixed copy icon alignment in RTL languages.

Extended string statistics on the Info tab.

Fixed handling of translation files ignored in Git.

Improved metrics performance.

Fixed possible bug in saving glossaries.

Fixed consistency check behavior on languages with different plural rules.

[All changes in detail.](#)

Weblate 4.6.1

Released on May 2nd 2021.

Remove obsolete spam protection code.

Improve source plural check accuracy.

Update list of user interface languages in Docker.

Improved error messages when creating pull requests.

Fixed creating pull requests on Pagine.

Fixed triggering automatically installed add-ons.

Fixed possible caching issues on upgrade.

Fixed adding new units to monolingual translations using upload.
[All changes in detail.](#)

Weblate 4.6

Released on April 19th 2021.

The `auto_translate` management command has now a parameter for specifying translation mode.

`:ref:txt`[REDACTED]`

`[REDACTED]`

`[REDACTED]`

Added date filtering when browsing changes.

Improved activity charts.

Sender for contact form e-mails can now be configured.

`[REDACTED]` API `[REDACTED]`

The rate limiting no longer applies to superusers.

`[REDACTED]`

`[REDACTED]`Docker `[REDACTED]`

API for creating components now automatically uses [Weblate \[REDACTED\] URL](#).

Simplified state indication while listing strings.

Password hashing now uses Argon2 by default.

Simplified progress bars indicating translation status.

Renamed `[REDACTED]` to clarify the purpose.

Fixed saving string state to XLIFF.

Added language-wide search.

Initial support for *Scaling horizontally* the Docker deployment.

`[REDACTED]`

Weblate 4.5.3

Released on April 1st 2021.

`[REDACTED]`

`[REDACTED]`

`[REDACTED]`

Fixed possible loss of newly added strings on replace upload.

Weblate 4.5.2

Released on March 26th 2021.

`[REDACTED]`

Lua `[REDACTED]`

Ignore format strings in the `[REDACTED]` check.

Allow uploading screenshot from a translate page.

Added forced file synchronization to the repository maintenance.

`[REDACTED]`

`[REDACTED]`

`[REDACTED]`

Several performance improvements.

[Weblate \[REDACTED\]](#) `[REDACTED]`

XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXX

Weblate 4.5.1

Released on March 5th 2021.

Fixed editing of glossary flags in some corner cases.

Extend metrics usage to improve performance of several pages.

TMX XXXXXXXXXXXXXXXXXXXX

API XXXXXXXXXX PO XXXXXXXXXXXXXXX

Improved alerts behavior glossaries.

Markdown XXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXX

XX

Improved bulk edit performance.

Fixed preserving "Needs editing" and "Approved" states for ODF files.

XX

XXXXXXXXXX

Weblate 4.5

Released on February 19th 2021.

gettext PO XXXXX lua-format XXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Fixed multiple unnamed variables check behavior with multiple format flags.

Dropped mailing list field on the project in favor of generic instructions for translators.

Added pseudolocale generation addon.

TermBase eXchange XXXXXXXXX

XX

XXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXX

Strings can now be added and removed in bilingual formats as well.

Amazon Translate XXXXXXXXXXXXXXXXXXXXXXX

Java XXXXXX Java MessageFormat XXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

Glossaries are now stored as regular components.

Dropped specific API for glossaries as component API is used now.

Added simplified interface to toggle some of the flags.

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

Moved text direction toggle to get more space for the visual keyboard.

XX

Added check whether translation matches the glossary.

XXXXXXXXXXXXXX

XXXXXXXXXX

Weblate 4.4.2

Released on January 14th 2021.

Fixed corruption of one distributed MO file.

Weblate 4.4.1

Released on January 13th 2021.

????????????

Fixed displaying help for project settings.

????????????

???? PO ?????????????????????

Fixed cleanup addon behavior with HTML, ODF, IDML and Windows RC formats.

CSV ?????????????????????

Use content compression for file downloads.

Improved user experience on importing from ZIP file.

????????????????????

Avoid duplicate pull requests on Pagure.

????????????????????

Reimplemented translation editor to use native browser textarea.

Fixed cleanup addon breaking adding new strings.

Added API for addons.

????????

Weblate 4.4

Released on December 15th 2020.

????????????

Weblate now requires Django 3.1.

????????

Fixed read-only state handling in bulk edit.

CodeMirror ?????

????????????????????

????? ?????????????????

Syntax highlighting in translation editor for XML, HTML, Markdown and reStructuredText.

????????

Improved support for non-standard language codes.

????????

The user is now presented with a filtered list of languages when adding a new translation.

Extended search capabilities for changes in history.

Improved billing detail pages and libre hosting workflow.

???? API ???

???? [????] ?????

Added tasks API.

???? ?????????

Improved display of user defined special characters.

????

???? ?????????

Improved naming of ZIP downloads.

??
??????????????

Weblate 4.3.2

Released on November 4th 2020.

Fixed crash on certain component filemasks.

??????????????????

Pagure ??? ?????????????????

?????????????? ???????????

?????????? Markdown ?????????????????????????????

Simplified setup of Git repositories with different default branch than "master".

Newly created internal repositories now use main as the default branch.

reStructuredText ?????????????????????????

Fixed CodeMirror display issues in some situations.

Renamed Template group to "Sources" to clarify its meaning.

Fixed GitLab pull requests on repositories with longer paths.

??????????????

Weblate 4.3.1

Released on October 21st 2020.

????????????????

??????????????????????????????????????

????????????????????????????

Improve hooks compatibility with Bitbucket Server.

????????????????????

Reduced memory usage.

??????????????

??????????????????????????????????????

??????????????

Weblate 4.3

Released on October 15th 2020.

Include user stats in the API.

Fixed component ordering on paginated pages.

??????????????????

Rewritten support for GitHub and GitLab pull requests.

??????????????????????????????????????

????????????????????

Fixed configuration of enforced checks.

Improve documentation about built-in backups.

??????????????????????????????????????

Vue I18n ???????????

Generic placeholders check now supports regular expressions.

??????????????

????????????????????

Added support for interacting with multiple GitLab or GitHub instances.
Extended API to cover project updates, unit updates and removals and glossaries.
Unit API now properly handles plural strings.
Component creation can now handle ZIP file or document upload.
Consolidated API response status codes.

????????? markdown ?????

????????????

Improved JSON, YAML and CSV formats compatibility.

????????????

Improved performance of file downloads.

Improved repository management view.

Automatically enable java-format for Android.

????????????????????????????

Python 3.9 ?????

Fixed translating HTML files under certain conditions.

????????????

Weblate 4.2.2

Released on September 2nd 2020.

JSON ??????????????????

Fixed login redirect for some authentication configurations.

Fixed LDAP authentication with group sync.

????????????????????????????????

Git ??????????????????

Fixed creating local VCS components using API.

Weblate 4.2.1

Released on August 21st 2020.

Android ??????????????????????

Fixed crash in the cleanup addon for some XLIFF files.

Allow setting up localization CDN in Docker image.

Weblate 4.2

Released on August 18th 2020.

Improved user pages and added listing of users.

Dropped support for migrating from 3.x releases, migrate through 4.1 or 4.0.

Added exports into several monolingual formats.

Improved activity charts.

Number of displayed nearby strings can be configured.

????????????????????????????????

Simplified main navigation (replaced buttons with icons).

Improved language code handling in Google Translate integration.

The Git squash addon can generate Co-authored-by: trailers.

Improved query search parser.

- Improved user feedback from format strings checks.
- Improved performance of bulk state changes.
- Added compatibility redirects after project or component renaming.
- Added notifications for strings approval, component locking and license change.
- Added support for ModernMT.
- Allow to avoid overwriting approved translations on file upload.
- Dropped support for some compatibility URL redirects.
- ECMAScript [2020](#) [release](#)
- [Removed leading dot from JSON unit keys.](#)
- [Added Celery \[integration\]\(#\)](#)
- Allow to configure Content-Security-Policy HTTP headers.
- Added support for aliasing languages at project level.
- New addon to help with HTML or JavaScript localization, see [JavaScript \[CDN\]\(#\)](#).
- The Weblate domain is now configured in the settings, see [SITE_DOMAIN](#).
- [Added support for \[SAML\]\(#\) \[integration\]\(#\)](#)

Weblate 4.1.1

- Released on June 19th 2020.
- Fixed changing autofix or addons configuration in Docker.
- Fixed possible crash in "About" page.
- [Added support for \[SAML\]\(#\) \[integration\]\(#\)](#)
- Fixed adding words to glossary.
- Fixed keyboard shortcuts for machinery.
- Removed debugging output causing discarding log events in some setups.
- Fixed lock indication on project listing.
- Fixed listing GPG keys in some setups.
- Added option for which DeepL API version to use.
- Added support for acting as SAML Service Provider, see [SAML \[integration\]\(#\)](#).

Weblate 4.1

- Released on June 15th 2020.
- Added support for creating new translations with included country code.
- Added support for searching source strings with screenshot.
- Extended info available in the stats insights.
- Improved search editing on "Translate" pages.
- Improve handling of concurrent repository updates.
- [Added support for \[SAML\]\(#\) \[integration\]\(#\)](#)
- Include changes count in credits.
- Fixed UI language selection in some cases.
- Allow to whitelist registration methods with registrations closed.
- Improved lookup of related terms in glossary.
- Improved translation memory matches.
- Group same machinery results.
- Add direct link to edit screenshot from translate page.

- Improved removal confirmation dialog.
- Include templates in ZIP download.
- Add support for Markdown and notification configuration in announcements.
- Extended details in check listings.
- [XXXXXXXXXXXXXXXXXXXX](#): [Laravel PHP](#) [HTML](#) [OpenDocument Format](#) [IDML Format](#) [Windows RC files](#) [INI translations](#) [Inno Setup INI](#) [GWT](#) [go-i18n JSON files](#) [ARB File](#)
- Consistently use dismissed as state of dismissed checks.
- Add support for configuring default addons to enable.
- Fixed editor keyboard shortcut to dismiss checks.
- Improved machine translation of strings with placeholders.
- Show ghost translation for user languages to ease starting them.
- Improved language code parsing.
- Show translations in user language first in the list.
- [XX](#)
- Added new quality checks: [XXXXXXXXXX](#), [XXXXXXXXXX](#), [XXXXXXXXXX](#).
- Reintroduced support for wiping translation memory.
- Fixed option to ignore source checks.
- Added support for configuring different branch for pushing changes.
- API [XXXXHTTP XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX](#)
- Added support for Google Translate V3 API (Advanced).
- Added ability to restrict access on component level.
- Added support for whitespace and other special chars in translation flags, see [XXXXXXXXXXXXXXXXXXXX](#).
- [XX](#)
- API now supports filtering of changes.
- Added support for sharing glossaries between projects.

Weblate 4.0.4

- Released on May 7th 2020.
- Fixed testsuite execution on some Python 3.8 environments.
- Typo fixes in the documentation.
- Fixed creating components using API in some cases.
- Fixed JavaScript errors breaking mobile navigation.
- Fixed crash on displaying some checks.
- Fixed screenshots listing.
- Fixed monthly digest notifications.
- Fixed intermediate translation behavior with units non existing in translation.

Weblate 4.0.3

- Released on May 2nd 2020.
- Fixed possible crash in reports.
- User mentions in comments are now case insensitive.
- Fixed PostgreSQL migration for non superusers.
- Fixed changing the repository URL while creating component.
- Fixed crash when upstream repository is gone.

Weblate 4.0.2

Released on April 27th 2020.

Improved performance of translation stats.
Improved performance of changing labels.
Improved bulk edit performance.
Improved translation memory performance.
Fixed possible crash on component deletion.
Fixed displaying of translation changes in some corner cases.
Improved warning about too long celery queue.
Fixed possible false positives in the consistency check.
Fixed deadlock when changing linked component repository.
Included edit distance in changes listing and CSV and reports.
Avoid false positives of punctuation spacing check for Canadian French.
Fixed XLIFF export with placeholders.
Fixed false positive with zero width check.
Improved reporting of configuration errors.
Fixed bilingual source upload.
Automatically detect supported languages for DeepL machine translation.
Fixed progress bar display in some corner cases.
Fixed some checks triggering on non translated strings.

Weblate 4.0.1

Released on April 16th 2020.

Fixed package installation from PyPI.

Weblate 4.0

Released on April 16th 2020.

Weblate now requires Python 3.6 or newer.
Added management overview of component alerts.
Added component alert for broken repository browser URLs.
Improved sign in and registration pages.
Project access control and workflow configuration integrated to project settings.
Added check and highlighter for i18next interpolation and nesting.
Added check and highlighter for percent placeholders.
Record source string changes in history.
Upgraded Microsoft Translator to version 3 API.
Reimplemented translation memory backend.
Added support for several `is:` lookups in `??`.
Allow to make `??????` avoid internal blacklist.
Improved comments extraction from monolingual po files.
Renamed whiteboard messages to announcements.
Fixed occasional problems with registration mails.
LINGUAS `????????????????????????????????????????????????????????????`
Fixed editing monolingual XLIFF source file.

API

Added support for intermediate language from developers.

Extended download options for platform wide translation memory.

Weblate 3.11.3

Fixed searching for fields with certain priority.

Fixed searching returning duplicate matches.

Fixed reverting changes from the history.

Fixed email for account removal confirmation.

Avoid sending notifications for not subscribed languages.

Fixed documentation links for addons.

Raised bleach dependency to address CVE-2020-6802.

Fixed stats invalidation in some corner cases.

Improved format string checks behavior on missing percent.

Fixed authentication using some third party providers.

Released on February 22nd 2020.

Fixed some strings wrongly reported as having no words.

Released on February 20th 2020.

Improved filename validation on component creation.

Fixed adding groups with certain Django versions.

Fixed manual pushing to upstream repository.

Improved glossary matching.

Weblate 3.11

Released on February 17th 2020.

Allow using VCS push URL during component creation via API.

????????????????????????????????

Fixed links in notifications e-mails.

Improved look of plaintext e-mails.

Display ignored checks and allow to make them active again.

Display nearby keys on monolingual translations.

????????????????????????????????

Recommend upgrade to new Weblate versions in the system checks.

Provide more detailed analysis for duplicate language alert.

Include more detailed license info on the project pages.

Automatically unshallow local copies if needed.

Fixed download of strings needing action.

New alert to warn about using the same filemask twice.

Improve XML placeables extraction.

The `SINGLE_PROJECT` can now enforce redirection to chosen project.

Added option to resolve comments.

Added bulk editing of flags.

Added support for labels.

Added bulk edit addon.

Added option for `??????`.

Increased default validity of confirmation links.

Improved Matomo integration.

Fixed `?????` to correctly handle source string change.

Extended automatic updates configuration by `AUTO_UPDATE`.

LINGUAS addons now do full sync of translations in Weblate.

Weblate 3.10.3

Released on January 18th 2020.

Support for translate-toolkit 2.5.0.

Weblate 3.10.2

Released on January 18th 2020.

Add lock indication to projects.

Fixed CSS bug causing flickering in some web browsers.

????????????????????????????????

Improved repository matching for GitHub and Bitbucket hooks.

Fixed data migration on some Python 2.7 installations.

Allow configuration of Git shallow cloning.

Improved background notification processing.

Fixed broken form submission when navigating back in web browser.

New addon to configure YAML formatting.

Fixed same plurals check to not fire on single plural form languages.

Fixed regex search on some fields.

Weblate 3.10.1

Released on January 9th 2020.

- Extended API with translation creation.
- Fixed several corner cases in data migrations.
- Compatibility with Django 3.0.
- Added support for customizable security.txt.
- Improved breadcrumbs in changelog.
- Improved translations listing on dashboard.
- Improved HTTP responses for webhooks.
- Added support for GitLab merge requests in Docker container.

Weblate 3.10

Released on December 20th 2019.

- Improved application user interface.
- Added doublespace check.
- Fixed creating new languages.
- Avoid sending auditlog notifications to deleted e-mails.
- Added support for Markdown in comments.
- Allow placing translation instruction text in project info.
- Improved support for Mercurial.
- Improved Git repository fetching performance.
- Add search lookup for age of string.
- Show context for nearby strings.
- Added support for notifications on repository operations.
- Improved translation listings.
- Extended search capabilities.
- Added support for automatic translation strings marked for editing.
- Avoid sending duplicate notifications for linked component alerts.
- Improve default merge request message.
- Better indicate string state in Zen mode.
- Added support for more languages in Yandex Translate.
- Improved look of notification e-mails.
- Provide choice for translation license.

Weblate 3.9.1

Released on October 28th 2019.

Remove some unneeded files from backups.

Fixed potential crash in reports.

Fixed cross database migration failure.

Added support for force pushing Git repositories.

Reduced risk of registration token invalidation.

Added search based on priority.

Fixed possible crash on adding strings to JSON file.

Safe HTML check and fixup now honor source string markup.

Avoid sending notifications to invited and deleted users.

Fix SSL connection to redis in Celery in Docker container.

Weblate 3.9

Released on October 15th 2019.

Include Weblate metadata in downloaded files.

Improved UI for failing checks.

Indicate missing strings in format checks.

Separate check for French punctuation spacing.

Add support for fixing some of quality checks errors.

Add separate permission to create new projects.

Extend stats for char counts.

Improve support for Java style language codes.

Added new generic check for placeholders.

Added support for WebExtension JSON placeholders.

Added support for flat XML format.

Extended API with project, component and translation removal and creation.

Added support for Gitea and Gitee webhooks.

Added new custom regex based check.

Allow to configure contributing to shared translation memory.

Added ZIP download for more translation files.

Make XLIFF standard compliant parsing of maxwidth and font.

Added new check and fixer for safe HTML markup for translating web applications.

Add component alert on unsupported configuration.

Added automatic translation addon to bootstrap translations.

Extend automatic translation to add suggestions.

Display addon parameters on overview.

Sentry is now supported through modern Sentry SDK instead of Raven.

Changed example settings to be better fit for production environment.

Added automated backups using BorgBackup.

Split cleanup addon for RESX to avoid unwanted file updates.

Added advanced search capabilities.

Allow users to download their own reports.

Added localization guide to help configuring components.

Added support for GitLab merge requests.
Improved display of repository status.
Perform automated translation in the background.

Weblate 3.8

Released on August 15th 2019.
Added support for simplified creating of similar components.
Added support for parsing translation flags from the XML based file formats.
Log exceptions into Celery log.
Improve performance of repository scoped addons.
Improved look of notification e-mails.
Fixed password reset behavior.
Improved performance on most of translation pages.
Fixed listing of languages not known to Weblate.
Add support for cloning addons to discovered components.
Add support for replacing file content with uploaded.
Add support for translating non VCS based content.
Added OpenGraph widget image to use on social networks.
Added support for animated screenshots.
Improved handling of monolingual XLIFF files.
Avoid sending multiple notifications for single event.
Add support for filtering changes.
Extended predefined periods for reporting.
Added webhook support for Azure Repos.
New opt-in notifications on pending suggestions or untranslated strings.
Add one click unsubscribe link to notification e-mails.
Fixed false positives with Has been translated check.
New management interface for admins.
String priority can now be specified using flags.
Added language management views.
Add checks for Qt library and Ruby format strings.
Added configuration to better fit single project installations.
Notify about new string on source string change on monolingual translations.
Added separate view for translation memory with search capability.

Weblate 3.7.1

Released on June 28th 2019.
Documentation updates.
Fixed some requirements constraints.
Updated language database.
Localization updates.
Various user interface tweaks.
Improved handling of unsupported but discovered translation files.
More verbosely report missing file format requirements.

Weblate 3.7

Released on June 21st 2019.

Added separate Celery queue for notifications.

Use consistent look with application for API browsing.

Include approved stats in the reports.

Report progress when updating translation component.

Allow to abort running background component update.

Extend template language for filename manipulations.

Use templates for editor link and repository browser URL.

Indicate max length and current characters count when editing translation.

????????????????????????????????

Refreshed landing page for new contributors.

Add support for configuring msgmerge addon.

Delay opening SMTP connection when sending notifications.

Improved error logging.

Allow custom location in MO generating addon.

Added addons to cleanup old suggestions or comments.

Added option to enable horizontal mode in the Zen editor.

Improved import performance with many linked components.

Fixed examples installation in some cases.

????????????????

Added new horizontal stats widget.

Improved format strings check on plurals.

Added font management tool.

????????????????????????????

Added support for subtitle formats.

Include overall completion stats for languages.

Added reporting at project and global scope.

Improved user interface when showing translation status.

New Weblate logo and color scheme.

New look of bitmap badges.

Weblate 3.6.1

Released on April 26th 2019.

Improved handling of monolingual XLIFF files.

Fixed digest notifications in some corner cases.

Fixed addon script error alert.

Fixed generating MO file for monolingual PO files.

Fixed display of uninstalled checks.

Indicate administered projects on project listing.

Allow update to recover from missing VCS repository.

Weblate 3.6

Released on April 20th 2019.

Add support for downloading user data.

Addons are now automatically triggered upon installation.

Improved instructions for resolving merge conflicts.

Cleanup addon is now compatible with app store metadata translations.

Configurable language code syntax when adding new translations.

Warn about using Python 2 with planned termination of support in April 2020.

Extract special characters from the source string for visual keyboard.

Extended contributor stats to reflect both source and target counts.

Admins and consistency addons can now add translations even if disabled for users.

Fixed description of toggle disabling Language-Team header manipulation.

Notify users mentioned in comments.

Removed file format autodetection from component setup.

Fixed generating MO file for monolingual PO files.

Added digest notifications.

Added support for muting component notifications.

Added notifications for new alerts, whiteboard messages or components.

Notifications for administered projects can now be configured.

Improved handling of three letter language codes.

Weblate 3.5.1

Released on March 10th 2019.

Fixed Celery systemd unit example.

Fixed notifications from HTTP repositories with login.

Fixed race condition in editing source string for monolingual translations.

Include output of failed addon execution in the logs.

Improved validation of choices for adding new language.

Allow to edit file format in component settings.

Update installation instructions to prefer Python 3.

Performance and consistency improvements for loading translations.

Microsoft Terminology service  Zeep                                 

Localization updates.

Weblate 3.5

Released on March 3rd 2019.

Improved performance of built-in translation memory.

Added interface to manage global translation memory.

Improved alerting on bad component state.

Added user interface to manage whiteboard messages.

Addon commit message now can be configured.

Reduce number of commits when updating upstream repository.

Fixed possible metadata loss when moving component between projects.

Improved navigation in the Zen mode.

Added several new quality checks (Markdown related and URL).

Added support for app store metadata files.
Added support for toggling GitHub or Gerrit integration.
Kashida ??????????
Added option to squash commits based on authors.
Improved support for XLSX file format.
Compatibility with Tesseract 4.0.
Billing addon now removes projects for unpaid billings after 45 days.

Weblate 3.4

Released on January 22nd 2019.
Added support for XLIFF placeholders.
Celery can now utilize multiple task queues.
Added support for renaming and moving projects and components.
Include characters counts in reports.
Added guided adding of translation components with automatic detection of translation files.
Customizable merge commit messages for Git.
Added visual indication of component alerts in navigation.
Improved performance of loading translation files.
New addon to squash commits prior to push.
Improved displaying of translation changes.
Changed default merge style to rebase and made that configurable.
Better handle private use subtags in language code.
Improved performance of fulltext index updates.
Extended file upload API to support more parameters.

Weblate 3.3

Released on November 30th 2018.
Added support for component and project removal.
Improved performance for some monolingual translations.
Added translation component alerts to highlight problems with a translation.
Expose XLIFF string resname as context when available.
Added support for XLIFF states.
Added check for non writable files in DATA_DIR.
Improved CSV export for changes.

Weblate 3.2.2

Released on October 20th 2018.
Remove no longer needed Babel dependency.
Updated language definitions.
Improve documentation for addons, LDAP and Celery.
Fixed enabling new dos-eol and auto-java-messageformat flags.
Fixed running setup.py test from PyPI package.
Improved plurals handling.
Fixed translation upload API failure in some corner cases.
Fixed updating Git configuration in case it was changed manually.

Weblate 3.2.1

Released on October 10th 2018.

Document dependency on backports.csv on Python 2.7.

Fix running tests under root.

Improved error handling in gitexport module.

Fixed progress reporting for newly added languages.

Correctly report Celery worker errors to Sentry.

Fixed creating new translations with Qt Linguist.

Fixed occasional fulltext index update failures.

Improved validation when creating new components.

Added support for cleanup of old suggestions.

Weblate 3.2

Released on October 6th 2018.

Add install_addon management command for automated addon installation.

????????????

Added support for export and import of Excel files.

Improve component cleanup in case of multiple component discovery addons.

Microsoft Terminology ??????????????????????

Weblate now uses Celery to offload some processing.

Improved search capabilities and added regular expression search.

Added support for Youdao Zhiyun API machine translation.

Added support for Baidu API machine translation.

Integrated maintenance and cleanup tasks using Celery.

Improved performance of loading translations by almost 25%.

Removed support for merging headers on upload.

Removed support for custom commit messages.

Configurable editing mode (zen/full).

Added support for error reporting to Sentry.

Added support for automated daily update of repositories.

Added support for creating projects and components by users.

????????????????????????????????

Users and projects can import their existing translation memories.

Better management of related strings for screenshots.

Added support for checking Java MessageFormat.

See [3.2 milestone on GitHub](#) for detailed list of addressed issues.

Weblate 3.1.1

Released on July 27th 2018.

Fix testsuite failure on some setups.

Weblate 3.1

Released on July 27th 2018.

Upgrades from older version than 3.0.1 are not supported.

Allow to override default commit messages from settings.

Improve webhooks compatibility with self hosted environments.

Added support for Amazon Translate.

Compatibility with Django 2.1.

Django system checks are now used to diagnose problems with installation.

Removed support for soon shutdown libavatar service.

??

Add support for jumping to specific location while translating.

Downloaded translations can now be customized.

Improved calculation of string similarity in translation memory matches.

Added support by signing Git commits by GnuPG.

Weblate 3.0.1

Released on June 10th 2018.

Fixed possible migration issue from 2.20.

Localization updates.

Removed obsolete hook examples.

Improved caching documentation.

Fixed displaying of admin documentation.

Improved handling of long language names.

Weblate 3.0

Released on June 1st 2018.

Rewritten access control.

Several code cleanups that lead to moved and renamed modules.

New addon for automatic component discovery.

The import_project management command has now slightly different parameters.

Added basic support for Windows RC files.

New addon to store contributor names in PO file headers.

The per component hook scripts are removed, use addons instead.

Add support for collecting contributor agreements.

Access control changes are now tracked in history.

New addon to ensure all components in a project have same translations.

Support for more variables in commit message templates.

Add support for providing additional textual context.

Weblate 2.x series

Weblate 2.20

Released on April 4th 2018.

Improved speed of cloning subversion repositories.

Changed repository locking to use third party library.

Added support for downloading only strings needing action.

Added support for searching in several languages at once.

New addon to configure gettext output wrapping.

New addon to configure JSON formatting.

Added support for authentication in API using RFC 6750 compatible Bearer authentication.

Added support for automatic translation using machine translation services.

Added support for HTML markup in whiteboard messages.

Added support for mass changing state of strings.

Translate-toolkit at least 2.3.0 is now required, older versions are no longer supported.

????????????

??

Added support for DeepL machine translation service.

Machine translation results are now cached inside Weblate.

??

Weblate 2.19.1

Released on February 20th 2018.

Fixed migration issue on upgrade from 2.18.

Improved file upload API validation.

Weblate 2.19

Released on February 15th 2018.

Fixed imports across some file formats.

Display human friendly browser information in audit log.

Added TMX exporter for files.

Various performance improvements for loading translation files.

Added option to disable access management in Weblate in favor of Django one.

Improved glossary lookup speed for large strings.

Compatibility with django_auth_ldap 1.3.0.

Configuration errors are now stored and reported persistently.

Honor ignore flags in whitespace autofixer.

Improved compatibility with some Subversion setups.

????????????????????

Added support for SAP Translation Hub service.

Microsoft Terminology service ??????????????

Removed support for advertisement in notification e-mails.

Improved translation progress reporting at language level.

Improved support for different plural formulas.

Added support for Subversion repositories not using stdlayout.

Added addons to customize translation workflows.

Weblate 2.18

Released on December 15th 2017.

Extended contributor stats.

Improved configuration of special characters virtual keyboard.

Added support for DTD file format.

Changed keyboard shortcuts to less likely collide with browser/system ones.

Improved support for approved flag in XLIFF files.

Added support for not wrapping long strings in gettext PO files.

Added button to copy permalink for current translation.

Dropped support for Django 1.10 and added support for Django 2.0.

Removed locking of translations while translating.

Added support for adding new strings to monolingual translations.

Added support for translation workflows with dedicated reviewers.

Weblate 2.17.1

Released on October 13th 2017.

Fixed running testsuite in some specific situations.

🐛🐛🐛🐛🐛

Weblate 2.17

Released on October 13th 2017.

Weblate by default does shallow Git clones now.

Improved performance when updating large translation files.

Added support for blocking certain e-mails from registration.

Users can now delete their own comments.

Added preview step to search and replace feature.

Client side persistence of settings in search and upload forms.

Extended search capabilities.

More fine grained per project ACL configuration.

Default value of BASE_DIR has been changed.

Added two step account removal to prevent accidental removal.

Project access control settings is now editable.

Added optional spam protection for suggestions using Akismet.

Weblate 2.16

Released on August 11th 2017.

Various performance improvements.

Added support for nested JSON format.

Added support for WebExtension JSON format.

Fixed git exporter authentication.

Improved CSV import in certain situations.

Improved look of Other translations widget.

The max-length checks is now enforcing length of text in form.

Make the commit_pending age configurable per component.
Various user interface cleanups.
Fixed component/project/site wide search for translations.

Weblate 2.15

Released on June 30th 2017.
Show more related translations in other translations.
Add option to see translations of current string to other languages.
Use 4 plural forms for Lithuanian by default.
Fixed upload for monolingual files of different format.
Improved error messages on failed authentication.
Keep page state when removing word from glossary.
Added Perl format quality check.
Added support for rejecting reused passwords.
Extended toolbar for editing RTL languages.

Weblate 2.14.1

Released on May 24th 2017.
Fixed possible error when paginating search results.
Fixed migrations from older versions in some corner cases.
Fixed possible CSRF on project watch and unwatch.
The password reset no longer authenticates user.
Fixed possible CAPTCHA bypass on forgotten password.

Weblate 2.14

Released on May 17th 2017.
Add glossary entries using AJAX.
The logout now uses POST to avoid CSRF.
The API key token reset now uses POST to avoid CSRF.
Weblate sets Content-Security-Policy by default.
The local editor URL is validated to avoid self-XSS.
The password is now validated against common flaws by default.
Notify users about important activity with their account such as password change.
The CSV exports now escape potential formulas.
Various minor improvements in security.
Suggestion content is stored in the history.
Store important account activity in audit log.
Ask for password confirmation when removing account or adding new associations.
Show time when suggestion has been made.
There is new quality check for trailing semicolon.
Ensure that search links can be shared.
Included source string information and screenshots in the API.
Allow to overwrite translations through API upload.

Weblate 2.13.1

Released on Apr 12th 2017.

Fixed listing of managed projects in profile.

Fixed migration issue where some permissions were missing.

Fixed listing of current file format in translation download.

Return HTTP 404 when trying to access project where user lacks privileges.

Weblate 2.13

Released on Apr 12th 2017.

Fixed quality checks on translation templates.

Added quality check to trigger on losing translation.

Add option to view pending suggestions from user.

???????? ?????????????????????????????????????

Default dashboard for unauthenticated users can be configured.

Add option to browse 25 random strings for review.

History now indicates string change.

Better error reporting when adding new translation.

Added per language search within project.

Group ACLs can now be limited to certain permissions.

The per project ALCs are now implemented using Group ACL.

Added more fine grained privileges control.

Various minor UI improvements.

Weblate 2.12

Released on Mar 3rd 2017.

Improved admin interface for groups.

Added support for Yandex Translate API.

Improved speed of site wide search.

Added project and component wide search.

Added project and component wide search and replace.

????????????????

Added support for opening source files in local editor.

Added support for configuring visual keyboard with special characters.

Improved screenshot management with OCR support for matching source strings.

Default commit message now includes translation information and URL.

Added support for Joomla translation format.

Improved reliability of import across file formats.

Weblate 2.11

Released on Jan 31st 2017.

Include language detailed information on language page.

Mercurial backend improvements.

Added option to specify translation component priority.

More consistent usage of Group ACL even with less used permissions.

Added WL_BRANCH variable to hook scripts.

Improved developer documentation.

Better compatibility with various Git versions in Git exporter addon.

Included per project and component stats.

Added language code mapping for better support of Microsoft Translate API.

Moved fulltext cleanup to background job to make translation removal faster.

Fixed displaying of plural source for languages with single plural form.

Improved error handling in import_project.

Various performance improvements.

Weblate 2.10.1

Released on Jan 20th 2017.

Do not leak account existence on password reset form (CVE-2017-5537).

Weblate 2.10

Released on Dec 15th 2016.

Added quality check to check whether plurals are translated differently.

Fixed GitHub hooks for repositories with authentication.

Added optional Git exporter module.

Support for Microsoft Cognitive Services Translator API.

Simplified project and component user interface.

Added automatic fix to remove control characters.

Added per language overview to project.

Added support for CSV export.

Added CSV download for stats.

Added matrix view for quick overview of all translations.

Added basic API for changes and strings.

Added support for Apertium APy server for machine translations.

Weblate 2.9

Released on Nov 4th 2016.

Extended parameters for createadmin management command.

Extended import_json to be able to handle with existing components.

Added support for YAML files.

Project owners can now configure translation component and project details.

Use "Watched" instead of "Subscribed" projects.

Projects can be watched directly from project page.

Added multi language status widget.

????????????????????????????????

Record suggestion deletion in history.

Improved UX of languages selection in profile.

Fixed showing whiteboard messages for component.

????????????????????????????????

Show source string comment more prominently.

Automatically install Gettext PO merge driver for Git repositories.

Added search and replace feature.

Added support for uploading visual context (screenshots) for translations.

Weblate 2.8

Released on Aug 31st 2016.

Documentation improvements.

Translations.

Updated bundled javascript libraries.

Added list_translators management command.

Django 1.8 is no longer supported.

Fixed compatibility with Django 1.10.

Added Subversion support.

Separated XML validity check from XML mismatched tags.

Fixed API to honor HIDE_REPO_CREDENTIALS settings.

Show source change in Zen mode.

Alt+PageUp/PageDown/Home/End now works in Zen mode as well.

Add tooltip showing exact time of changes.

Add option to select filters and search from translation page.

Added UI for translation removal.

Improved behavior when inserting placeables.

Fixed auto locking issues in Zen mode.

Weblate 2.7

Released on Jul 10th 2016.

Removed Google web translate machine translation.

Improved commit message when adding translation.

Fixed Google Translate API for Hebrew language.

Compatibility with Mercurial 3.8.

Added import_json management command.

Correct ordering of listed translations.

Show full suggestion text, not only a diff.

Extend API (detailed repository status, statistics, ...).

Testsuite no longer requires network access to test repositories.

Weblate 2.6

Released on Apr 28th 2016.

Fixed validation of components with language filter.

Improved support for XLIFF files.

Fixed machine translation for non English sources.

Added REST API.

Django 1.10 compatibility.

Added categories to whiteboard messages.

Weblate 2.5

Released on Mar 10th 2016.

Fixed automatic translation for project owners.

Improved performance of commit and push operations.

New management command to add suggestions from command line.

Added support for merging comments on file upload.

Added support for some GNU extensions to C printf format.

Documentation improvements.

Added support for generating translator credits.

Added support for generating contributor stats.

Site wide search can search only in one language.

Improve quality checks for Armenian.

Support for starting translation components without existing translations.

Support for adding new translations in Qt TS.

Improved support for translating PHP files.

Performance improvements for quality checks.

????????????????????????????????

????????????????????????

Improved support for XLIFF files.

Extended list of options for import_project.

Improved targeting for whiteboard messages.

Support for automatic translation across projects.

Optimized fulltext search index.

Added management command for auto translation.

Added placeables highlighting.

Added keyboard shortcuts for placeables, checks and machine translations.

Improved translation locking.

Added quality check for AngularJS interpolation.

Added extensive group based ACLs.

???????????????????????????????? "fuzzy"??

????????????????????????????????

Support for Python 3.

Dropped support for Django 1.7.

Dropped dependency on msginit for creating new gettext PO files.

Added configurable dashboard views.

Improved notifications on parse errors.

Added option to import components with duplicate name to `import_project`.
Improved support for translating PHP files.
Added XLIFF export for dictionary.
Added XLIFF and gettext PO export for all translations.
Documentation improvements.
Added support for configurable automatic group assignments.
Improved adding of new translations.

Weblate 2.4

Released on Sep 20th 2015.
Improved support for PHP files.
Ability to add ACL to anonymous user.
Improved configurability of `import_project` command.
Added CSV dump of history.
Avoid copy/paste errors with whitespace characters.
Added support for Bitbucket webhooks.
Tighter control on fuzzy strings on translation upload.
Several URLs have changed, you might have to update your bookmarks.
Hook scripts are executed with VCS root as current directory.
Hook scripts are executed with environment variables describing current component.
Add management command to optimize fulltext index.
Added support for error reporting to Rollbar.
Projects now can have multiple owners.
Project owners can manage themselves.
Added support for `javascript-format` used in gettext PO.
Support for adding new translations in XLIFF.
Improved file format autodetection.
Extended keyboard shortcuts.
Improved dictionary matching for several languages.
Improved layout of most of pages.
Support for adding words to dictionary while translating.
Added support for filtering languages to be managed by Weblate.
Added support for translating and importing CSV files.
Rewritten handling of static files.
Direct login/registration links to third-party service if that's the only one.
Commit pending changes on account removal.
Add management command to change site name.
Add option to configure default committer.
Add hook after adding new translation.
Add option to specify multiple files to add to commit.

Weblate 2.3

Released on May 22nd 2015.

Dropped support for Django 1.6 and South migrations.

Support for adding new translations when using Java Property files.

Allow to accept suggestion without editing.

Improved support for Google OAuth 2.0.

Added support for Microsoft .resx files.

Tuned default robots.txt to disallow big crawling of translations.

Simplified workflow for accepting suggestions.

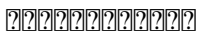
Added project owners who always receive important notifications.

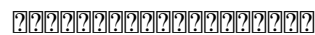
Allow to disable editing of monolingual template.

More detailed repository status view.

Direct link for editing template when changing translation.

Allow to add more permissions to project owners.

Zen 



Weblate 2.2

Released on Feb 19th 2015.

Performance improvements.

Fulltext search on location and comments fields.

New SVG/javascript based activity charts.

Support for Django 1.8.

Support for deleting comments.

Added own SVG badge.

Added support for Google Analytics.

Improved handling of translation filenames.

Added support for monolingual JSON translations.

Record component locking in a history.

Support for editing source (template) language for monolingual translations.

Added basic support for Gerrit.

Weblate 2.1

Released on Dec 5th 2014.

Added support for Mercurial repositories.

Replaced Glyphicon font by Awesome.

Added icons for social authentication services.

Better consistency of button colors and icons.

Documentation improvements.

Various bugfixes.

Automatic hiding of columns in translation listing for small screens.

Changed configuration of filesystem paths.

Improved SSH keys handling and storage.

Improved repository locking.

Customizable quality checks per source string.

Allow to hide completed translations from dashboard.

Weblate 2.0

Released on Nov 6th 2014.
New responsive UI using Bootstrap.
Rewritten VCS backend.
Documentation improvements.
Added whiteboard for site wide messages.
Configurable strings priority.
Added support for JSON file format.
Fixed generating mo files in certain cases.
Added support for GitLab notifications.
Added support for disabling translation suggestions.
Django 1.7 support.
ACL projects now have user management.
Extended search possibilities.
Give more hints to translators about plurals.
Fixed Git repository locking.
Compatibility with older Git versions.
Improved ACL support.
Added buttons for per language quotes and other special characters.
Support for exporting stats as JSONP.

Weblate 1.x series

Weblate 1.9

Released on May 6th 2014.
Django 1.6 compatibility.
No longer maintained compatibility with Django 1.4.
Management commands for locking/unlocking translations.
Improved support for Qt TS files.
Users can now delete their account.
Avatars can be disabled.
Merged first and last name attributes.
Avatars are now fetched and cached server side.
Added support for shields.io badge.

Weblate 1.8

Released on November 7th 2013.
Please check manual for upgrade instructions.
Nicer listing of project summary.
Better visible options for sharing.
More control over anonymous users privileges.
Supports login using third party services, check manual for more details.
Users can login by e-mail instead of username.

Documentation improvements.
Improved source strings review.
Searching across all strings.
Better tracking of source strings.
Captcha protection for registration.

Weblate 1.7

Released on October 7th 2013.
Please check manual for upgrade instructions.
Support for checking Python brace format string.
Per component customization of quality checks.
Detailed per translation stats.
Changed way of linking suggestions, checks and comments to strings.
Users can now add text to commit message.
Support for subscribing on new language requests.
Support for adding new translations.
Pango + Cairo Pillow
Add status badge widget.
Changes in dictionary are now logged in history.
Performance improvements for translating view.

Weblate 1.6

Released on July 25th 2013.
Nicer error handling on registration.
Browsing of changes.
Fixed sorting of machine translation suggestions.
Improved support for MyMemory machine translation.
Added support for Amagama machine translation.
Various optimizations on frequently used pages.
Highlights searched phrase in search results.
Support for automatic fixups while saving the message.
Tracking of translation history and option to revert it.
Added support for Google Translate API.
Added support for managing SSH host keys.
Various form validation improvements.
Various quality checks improvements.
Performance improvements for import.
Added support for voting on suggestions.
Cleanup of admin interface.

Weblate 1.5

Released on April 16th 2013.

Please check manual for upgrade instructions.

Added public user pages.

Better naming of plural forms.

Added support for TBX export of glossary.

Added support for Bitbucket notifications.

Activity charts are now available for each translation, language or user.

Extended options of import_project admin command.

Compatible with Django 1.5.

Avatars are now shown using libravatar.

Added possibility to pretty print JSON export.

Various performance improvements.

Indicate failing checks or fuzzy strings in progress bars for projects or languages as well.

Added support for custom pre-commit hooks and committing additional files.

Rewritten search for better performance and user experience.

New interface for machine translations.

Added support for monolingual po files.

Extend amount of cached metadata to improve speed of various searches.

Now shows word counts as well.

Weblate 1.4

Released on January 23rd 2013.

Fixed deleting of checks/comments on string deletion.

Added option to disable automatic propagation of translations.

Added option to subscribe for merge failures.

Correctly import on projects which needs custom tkit loader.

Added sitemaps to allow easier access by crawlers.

Provide direct links to string in notification e-mails or feeds.

Various improvements to admin interface.

Provide hints for production setup in admin interface.

Added per language widgets and engage page.

Improved translation locking handling.

Indicate failing checks or fuzzy strings in progress bars.

More options for formatting commit message.

Fixed error handling with machine translation services.

Improved automatic translation locking behaviour.

Support for showing changes from previous source string.

Added support for substring search.

Various quality checks improvements.

Support for per project ACL.

Basic code coverage by unit tests.

Weblate 1.3

Released on November 16th 2012.

Compatibility with PostgreSQL database backend.

Removes languages removed in upstream git repository.

Improved quality checks processing.

Added new checks (BB code, XML markup and newlines).

Support for optional rebasing instead of merge.

Possibility to relocate Weblate (for example to run it under /weblate path).

Support for manually choosing file type in case autodetection fails.

Better support for Android resources.

Support for generating SSH key from web interface.

More visible data exports.

New buttons to enter some special characters.

Support for exporting dictionary.

Support for locking down whole Weblate installation.

Checks for source strings and support for source strings review.

Support for user comments for both translations and source strings.

Better changes log tracking.

Changes can now be monitored using RSS.

Improved support for RTL languages.

Weblate 1.2

Released on August 14th 2012.

Weblate now uses South for database migration, please check upgrade instructions if you are upgrading.

Fixed minor issues with linked git repos.

New introduction page for engaging people with translating using Weblate.

Added widgets which can be used for promoting translation projects.

Added option to reset repository to origin (for privileged users).

Project or component can now be locked for translations.

Possibility to disable some translations.

Configurable options for adding new translations.

Configuration of git commits per project.

Simple antispam protection.

Better layout of main page.

Support for automatically pushing changes on every commit.

Support for e-mail notifications of translators.

????????????????????????????????

Improved handling of not known languages when importing project.

Support for locking translation by translator.

Optionally maintain Language-Team header in po file.

Include some statistics in about page.

Supports (and requires) django-registration 0.8.

????????????????????????????

Checking of requirements during setup.

Documentation improvements.

Weblate 1.1

Released on July 4th 2012.
Improved several translations.
Better validation while creating component.
Added support for shared git repositories across components.
Do not necessary commit on every attempt to pull remote repo.
Added support for offloading indexing.

Weblate 1.0

Released on May 10th 2012.
Improved validation while adding/saving component.
Experimental support for Android component files (needs patched tkit).
Updates from hooks are run in background.
Improved installation instructions.
Improved navigation in dictionary.

Weblate 0.x series

Weblate 0.9

Released on April 18th 2012.
Fixed import of unknown languages.
Improved listing of nearby messages.
Improved several checks.
Documentation updates.
Added definition for several more languages.
Various code cleanups.
Documentation improvements.
Changed file layout.
Update helper scripts to Django 1.4.
Improved navigation while translating.
Better handling of po file renames.
Better validation while creating component.
Integrated full setup into syncdb.
Added list of recent changes to all translation pages.
Check for not translated strings ignores format string only messages.

Weblate 0.8

Released on April 3rd 2012.
Replaced own full text search with Whoosh.
Various fixes and improvements to checks.
New command updatechecks.
Lot of translation updates.
Added dictionary for storing most frequently used terms.
Added /admin/report/ for overview of repositories status.
Machine translation services no longer block page loading.

Management interface now contains also useful actions to update data.
Records log of changes made by users.
Ability to postpone commit to Git to generate less commits from single user.
Possibility to browse failing checks.
Automatic translation using already translated strings.
New about page showing used versions.
Django 1.4 compatibility.
Ability to push changes to remote repo from web interface.
Added review of translations done by others.

Weblate 0.7

Released on February 16th 2012.
Direct support for GitHub notifications.
Added support for cleaning up orphaned checks and translations.
Displays nearby strings while translating.
Displays similar strings while translating.
Improved searching for string.

Weblate 0.6

Released on February 14th 2012.
Added various checks for translated messages.
Tunable access control.
Improved handling of translations with new lines.
Added client side sorting of tables.
Please check upgrading instructions in case you are upgrading.

Weblate 0.5

Released on February 12th 2012.

ng online services:

Apertium
Microsoft Translator
MyMemory
Several new translations.
Improved merging of upstream changes.
Better handle concurrent git pull and translation.
Propagating works for fuzzy changes as well.
Propagating works also for file upload.
Fixed file downloads while using FastCGI (and possibly others).

Weblate 0.4

Released on February 8th 2012.
Added usage guide to documentation.
Fixed API hooks not to require CSRF protection.

Weblate 0.3

Released on February 8th 2012.
Better display of source for plural translations.
New documentation in Sphinx format.
????????????
Improved error page to give list of existing projects.
New per language stats.

Weblate 0.2

Released on February 7th 2012.
Improved validation of several forms.
Warn users on profile upgrade.
Remember URL for login.
Naming of text areas while entering plural forms.
Automatic expanding of translation area.

Weblate 0.1

Released on February 6th 2012.
Initial release.

W

wlc, 112

wlc.config, 113

wlc.main, 113

```

/
ANY /, 73 /api
GET /api/, 75 /api/addons
GET /api/addons/, 103
GET /api/addons/(int:id)/, 103
PUT /api/addons/(int:id)/, 103
DELETE /api/addons/(int:id)/, 103
PATCH /api/addons/(int:id)/, 103 /api/changes
GET /api/changes/, 101
GET /api/changes/(int:id)/, 101 /api/component-lists
GET /api/component-lists/, 104
GET /api/component-lists/(str:slug)/, 104
POST /api/component-lists/(str:slug)/components/, 104
PUT /api/component-lists/(str:slug)/, 104
DELETE /api/component-lists/(str:slug)/, 104
DELETE /api/component-lists/(str:slug)/components/(str:component_slug), 104
PATCH /api/component-lists/(str:slug)/, 104 /api/components
GET /api/components/, 88
GET /api/components/(string:project)/(string:component)/, 88
GET /api/components/(string:project)/(string:component)/changes/, 91
GET /api/components/(string:project)/(string:component)/links/, 94
GET /api/components/(string:project)/(string:component)/lock/, 91
GET /api/components/(string:project)/(string:component)/monolingual_base/, 93
GET /api/components/(string:project)/(string:component)/new_template/, 93
GET /api/components/(string:project)/(string:component)/repository/, 92
GET /api/components/(string:project)/(string:component)/screenshots/, 91
GET /api/components/(string:project)/(string:component)/statistics/, 94
GET /api/components/(string:project)/(string:component)/translations/, 93
POST /api/components/(string:project)/(string:component)/addons/, 103
POST /api/components/(string:project)/(string:component)/links/, 95
POST /api/components/(string:project)/(string:component)/lock/, 91
POST /api/components/(string:project)/(string:component)/repository/, 92
POST /api/components/(string:project)/(string:component)/translations/, 93
PUT /api/components/(string:project)/(string:component)/, 90
DELETE /api/components/(string:project)/(string:component)/, 91
DELETE /api/components/(string:project)/(string:component)/links/(string:project_slug)/, 95
PATCH /api/components/(string:project)/(string:component)/, 89 /api/groups
GET /api/groups/, 78
GET /api/groups/(int:id)/, 78
POST /api/groups/, 78
POST /api/groups/(int:id)/componentlists/, 80
POST /api/groups/(int:id)/components/, 79
POST /api/groups/(int:id)/languages/, 80
POST /api/groups/(int:id)/projects/, 80
POST /api/groups/(int:id)/roles/, 79

```

```

PUT /api/groups/(int:id)/, 79
DELETE /api/groups/(int:id)/, 79
DELETE /api/groups/(int:id)/componentlists/(int:component_list_id), 80
DELETE /api/groups/(int:id)/components/(int:component_id), 79
DELETE /api/groups/(int:id)/languages/(string:language_code), 80
DELETE /api/groups/(int:id)/projects/(int:project_id), 80
PATCH /api/groups/(int:id)/, 79 /api/languages
GET /api/languages/, 81
GET /api/languages/(string:language)/, 81
GET /api/languages/(string:language)/statistics/, 82
POST /api/languages/, 81
PUT /api/languages/(string:language)/, 82
DELETE /api/languages/(string:language)/, 82
PATCH /api/languages/(string:language)/, 82 /api/projects
GET /api/projects/, 83
GET /api/projects/(string:project)/, 83
GET /api/projects/(string:project)/changes/, 84
GET /api/projects/(string:project)/components/, 85
GET /api/projects/(string:project)/languages/, 87
GET /api/projects/(string:project)/repository/, 84
GET /api/projects/(string:project)/statistics/, 87
POST /api/projects/, 83
POST /api/projects/(string:project)/components/, 85
POST /api/projects/(string:project)/repository/, 84
PUT /api/projects/(string:project)/, 84
DELETE /api/projects/(string:project)/, 84
PATCH /api/projects/(string:project)/, 83 /api/roles
GET /api/roles/, 80
GET /api/roles/(int:id)/, 80
POST /api/roles/, 80
PUT /api/roles/(int:id)/, 81
DELETE /api/roles/(int:id)/, 81
PATCH /api/roles/(int:id)/, 81 /api/screenshots
GET /api/screenshots/, 101
GET /api/screenshots/(int:id)/, 101
GET /api/screenshots/(int:id)/file/, 101
POST /api/screenshots/, 102
POST /api/screenshots/(int:id)/file/, 102
POST /api/screenshots/(int:id)/units/, 102
PUT /api/screenshots/(int:id)/, 102
DELETE /api/screenshots/(int:id)/, 103
DELETE /api/screenshots/(int:id)/units/(int:unit_id), 102
PATCH /api/screenshots/(int:id)/, 102 /api/tasks
GET /api/tasks/, 105
GET /api/tasks/(str:uuid)/, 105 /api/translations

```

```

GET /api/translations/, 95
GET /api/translations/(string:project)/(string:component)/(string:language)/,
95
GET /api/translations/(string:project)/(string:component)/(string:language)/changes/,
97
GET /api/translations/(string:project)/(string:component)/(string:language)/file/,
98
GET /api/translations/(string:project)/(string:component)/(string:language)/repository/,
98
GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/,
99
GET /api/translations/(string:project)/(string:component)/(string:language)/units/,
97
POST /api/translations/(string:project)/(string:component)/(string:language)/autotransla
97
POST /api/translations/(string:project)/(string:component)/(string:language)/file/,
98
POST /api/translations/(string:project)/(string:component)/(string:language)/repository/,
99
POST /api/translations/(string:project)/(string:component)/(string:language)/units/,
97
DELETE /api/translations/(string:project)/(string:component)/(string:language)/,
97/api/units
GET /api/units/, 99
GET /api/units/(int:id)/, 99
PUT /api/units/(int:id)/, 100
DELETE /api/units/(int:id)/, 100
PATCH /api/units/(int:id)/, 100/api/users
GET /api/users/, 76
GET /api/users/(str:username)/, 76
GET /api/users/(str:username)/notifications/, 77
GET /api/users/(str:username)/notifications/(int:subscription_id)/, 77
GET /api/users/(str:username)/statistics/, 77
POST /api/users/, 76
POST /api/users/(str:username)/groups/, 77
POST /api/users/(str:username)/notifications/, 77
PUT /api/users/(str:username)/, 76
PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 78
DELETE /api/users/(str:username)/, 77
DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 78
PATCH /api/users/(str:username)/, 77
PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 78 /ex-
ports
GET /exports/rss/, 108
GET /exports/rss/(string:project)/, 108
GET /exports/rss/(string:project)/(string:component)/, 108
GET /exports/rss/(string:project)/(string:component)/(string:language)/, 108
GET /exports/rss/language/(string:language)/, 108
GET /exports/stats/(string:project)/(string:component)/, 106/hooks
GET /hooks/update/(string:project)/, 105

```

GET /hooks/update/{string:project}/{string:component}/, 105
POST /hooks/azure/, 106
POST /hooks/bitbucket/, 105
POST /hooks/gitea/, 106
POST /hooks/gitee/, 106
POST /hooks/github/, 105
POST /hooks/gitlab/, 105
POST /hooks/pagure/, 106

```
.XML resource file file format, 62
--add auto_translate [options], 285
--addon ADDON install_addon [options], 290
--age HOURS commit_pending [options], 286
--author USER@EXAMPLE.COM add_suggestions [options], 285
--base-file-template TEMPLATE import_project [options], 288
--check importusers [options], 290
--config PATH wlc [options], 109
--config-section SECTION wlc [options], 109
--configuration CONFIG install_addon [options], 290
--convert wlc [options], 110
--email USER@EXAMPLE.COM createadmin [options], 286
--file-format FORMAT import_project [options], 288
--force loadpo [options], 291
--force-commit pushgit [options], 292
--format {csv,json,text,html} wlc [options], 109
--ignore import_json [options], 287
--inconsistent auto_translate [options], 285
--input wlc [options], 110
--key KEY wlc [options], 109
--lang LANGUAGE loadpo [options], 291
--language-code list_translators [options], 290
--language-map LANGMAP import_memory [options], 288
--language-regex REGEX import_project [options], 288
--license NAME import_project [options], 288
--license-url URL import_project [options], 289
--main-component import_project [options], 288
--main-component COMPONENT import_json [options], 287
--mode MODE auto_translate [options], 285
--mt MT auto_translate [options], 285
--name createadmin [options], 286
--name-template TEMPLATE import_project [options], 288
--new-base-template TEMPLATE import_project [options], 288
--no-password createadmin [options], 286
--no-privs-update setupgroups [options], 292
--no-projects-update setupgroups [options], 292
--no-update setuplang [options], 292
--output wlc [options], 110
--overwrite auto_translate [options], 285 wlc [options], 110
--password PASSWORD createadmin [options], 286
--project PROJECT import_json [options], 287
--source PROJECT/COMPONENT auto_translate [options], 285
--threshold THRESHOLD auto_translate [options], 285
--update createadmin [options], 286 import_json [options], 287 in-
stall_addon [options], 290
--url URL wlc [options], 109
--user USERNAME auto_translate [options], 285
```

--username USERNAME createadmin [XXXXXXXXXXXX](#), 286
 --vcs NAME import_project [XXXXXXXXXXXX](#), 289
 add_suggestions weblate admin command, 285
 add_suggestions [XXXXXXXXXXXX](#) --author USER@EXAMPLE.COM, 285
 ADMINS setting, 148
 AKISMET_API_KEY setting, 246
 ALLOWED_HOSTS setting, 148
 Android file format, 58
 ANONYMOUS_USER_NAME setting, 246
 API, 73, 108, 112
 Apple strings file format, 59
 ARB file format, 61
 AUDITLOG_EXPIRY setting, 246
 AUTH_LOCK_ATTEMPTS setting, 247
 AUTH_TOKEN_VALID setting, 247
 auto_translate weblate admin command, 285
 auto_translate [XXXXXXXXXXXX](#) --add, 285 --inconsistent, 285 --mode MODE, 285 --mt MT, 285 --overwrite, 285 --source PROJECT/COMPONENT, 285 --threshold THRESHOLD, 285 --user USERNAME, 285
 AUTO_UPDATE setting, 247
 AUTOFIX_LIST setting, 248
 AVATAR_URL_PREFIX setting, 247
 BACKGROUND_TASKS setting, 248
 BASE_DIR setting, 248
 BaseAddon (*weblate.addons.base* [XXXXXX](#)), 322
 BASIC_LANGUAGES setting, 249
 bilingual translation, 52
 can_install() (*weblate.addons.base.BaseAddon* [XXXXXXXXXX](#)), 322
 CELERY_BACKUP_OPTIONS, 116, 128
 CELERY_BEAT_OPTIONS, 116, 128
 CELERY_MAIN_OPTIONS, 116, 128
 CELERY_MEMORY_OPTIONS, 116, 128
 CELERY_NOTIFY_OPTIONS, 116, 128
 celery_queues weblate admin command, 285
 CELERY_TRANSLATE_OPTIONS, 116, 128
 changes wlc [XXXXXXXXXXXX](#), 110
 CHECK_LIST setting, 249
 checkgit weblate admin command, 286
 cleanup wlc [XXXXXXXXXXXX](#), 110
 cleanuptrans weblate admin command, 286
 Comma separated values file format, 63
 Command (*wlc.main* [XXXXX](#)), 113
 COMMENT_CLEANUP_DAYS setting, 250
 commit wlc [XXXXXXXXXXXX](#), 110
 commit_pending weblate admin command, 286
 commit_pending [XXXXXXXXXXXX](#) --age HOURS, 286
 COMMIT_PENDING_HOURS setting, 250

commitgit weblate admin command, 286
 configure() (*weblate.addons.base.BaseAddon* [??????](#)), 322
 CONTACT_FORM setting, 250
 createadmin weblate admin command, 286
 createadmin [????????????](#) --email USER@EXAMPLE.COM, 286 --name, 286 --no-password, 286 --password PASSWORD, 286 --update, 286 --username USERNAME, 286
 CSP_CONNECT_SRC setting, 249
 CSP_FONT_SRC setting, 249
 CSP_IMG_SRC setting, 249
 CSP_SCRIPT_SRC setting, 249
 CSP_STYLE_SRC setting, 249
 CSV file format, 63
 daily() (*weblate.addons.base.BaseAddon* [??????](#)), 322
 DATA_DIR setting, 250
 DATABASE_BACKUP setting, 250
 DATABASES setting, 149
 DEBUG setting, 149
 DEFAULT_ACCESS_CONTROL setting, 251
 DEFAULT_ADD_MESSAGE setting, 251
 DEFAULT_ADDON_MESSAGE setting, 251
 DEFAULT_ADDONS setting, 251
 DEFAULT_AUTO_WATCH setting, 251
 DEFAULT_COMMIT_MESSAGE setting, 251
 DEFAULT_COMMITTER_EMAIL setting, 252
 DEFAULT_COMMITTER_NAME setting, 252
 DEFAULT_DELETE_MESSAGE setting, 251
 DEFAULT_FROM_EMAIL setting, 149
 DEFAULT_LANGUAGE setting, 252
 DEFAULT_MERGE_MESSAGE setting, 251
 DEFAULT_MERGE_STYLE setting, 252
 DEFAULT_PULL_MESSAGE setting, 253
 DEFAULT_RESTRICTED_COMPONENT setting, 251
 DEFAULT_SHARED_TM setting, 253
 DEFAULT_TRANSLATION_PROPAGATION setting, 253
 downloadwlc [????????????](#), 110
 DTD file format, 64
 dump_memory weblate admin command, 287
 dumpuserdata weblate admin command, 287
 ENABLE_AVATARS setting, 253
 ENABLE_HOOKS setting, 253
 ENABLE_HTTPS setting, 253
 ENABLE_SHARING setting, 253
 file format .XML resource file, 62 Android, 58 Apple strings, 59 ARB, 61 Comma separated values, 63 CSV, 63 DTD, 64 gettext, 54 go-i18n, 61 GWT properties, 57 i18next, 61 INI translations, 57 Java properties, 56 Joomla translations, 58 JSON, 60 PHP strings, 59 PO, 54 Qt, 58 RC, 65 RESX, 62 Ruby YAML, 64 Ruby YAML Ain't Markup Language, 64 string resources, 58 TS, 58 XLIFF, 55 XML, 64 YAML, 63 YAML Ain't Markup Language, 63
 get() (*wlc.Weblate* [??????](#)), 112

[get_add_form\(\)](#) (*weblate.addons.base.BaseAddon* [????????](#)), 322
[GET_HELP_URL](#) setting, 254
[get_settings_form\(\)](#) (*weblate.addons.base.BaseAddon* [????](#)), 322
[gettext](#) file format, 54
[GITHUB_CREDENTIALS](#) setting, 254
[GITHUB_TOKEN](#) setting, 255
[GITHUB_USERNAME](#) setting, 255
[GITLAB_CREDENTIALS](#) setting, 254
[GITLAB_TOKEN](#) setting, 254
[GITLAB_USERNAME](#) setting, 254
[go-i18n](#) file format, 61
[GOOGLE_ANALYTICS_ID](#) setting, 255
[GWT](#) properties file format, 57
[HIDE_REPO_CREDENTIALS](#) setting, 255
[HIDE_VERSION](#) setting, 255
[i18next](#) file format, 61
[import_demo](#) weblate admin command, 287
[import_json](#) weblate admin command, 287
[import_json](#) [????????????](#) --ignore, 287 --main-component COMPONENT, 287 --project PROJECT, 287 --update, 287
[import_memory](#) weblate admin command, 288
[import_memory](#) [????????????](#) --language-map LANGMAP, 288
[import_project](#) weblate admin command, 288
[import_project](#) [????????????](#) --base-file-template TEMPLATE, 288 --file-format FORMAT, 288 --language-regex REGEX, 288 --license NAME, 288 --license-url URL, 289 --main-component, 288 --name-template TEMPLATE, 288 --new-base-template TEMPLATE, 288 --vcs NAME, 289
[importuserdata](#) weblate admin command, 290
[importusers](#) weblate admin command, 290
[importusers](#) [????????????](#) --check, 290
[INI](#) translations file format, 57
[install_addon](#) weblate admin command, 290
[install_addon](#) [????????????](#) --addon ADDON, 290 --configuration CONFIG, 290 --update, 290
[IP_BEHIND_REVERSE_PROXY](#) setting, 255
[IP_PROXY_HEADER](#) setting, 256
[IP_PROXY_OFFSET](#) setting, 256
[iPad](#) translation, 59
[iPhone](#) translation, 59
[Java](#) properties file format, 56
[Joomla](#) translations file format, 58
[JSON](#) file format, 60
[LEGAL_URL](#) setting, 256
[LICENSE_EXTRA](#) setting, 256
[LICENSE_FILTER](#) setting, 256
[LICENSE_REQUIRED](#) setting, 257
[LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH](#) setting, 257
[list_languages](#) weblate admin command, 290

list_translators weblate admin command, 290
 list_translators `??????????????` --language-code, 290
 list_versions weblate admin command, 291
 list-components wlc `??????????????`, 109
 list-languages wlc `??????????????`, 109
 list-projects wlc `??????????????`, 109
 list-translations wlc `??????????????`, 109
 load() (*wlc.config.WeblateConfig* `??????`), 113
 loadpo weblate admin command, 291
 loadpo `??????????????` --force, 291 --lang LANGUAGE, 291
 LOCALIZE_CDN_PATH setting, 257
 LOCALIZE_CDN_URL setting, 257
 lock wlc `??????????????`, 110
 lock_translation weblate admin command, 291
 lock-status wlc `??????????????`, 110
 LOGIN_REQUIRED_URLS setting, 257
 LOGIN_REQUIRED_URLS_EXCEPTIONS setting, 258
 ls wlc `??????????????`, 110
 MACHINE_TRANSLATION_SERVICES setting, 258
 main() (*wlc.main* `??????`), 113
 MATOMO_SITE_ID setting, 258
 MATOMO_URL setting, 258
 monolingual translation, 52
 move_language weblate admin command, 291
 MT_APERTIUM_APY setting, 259
 MT_AWS_ACCESS_KEY_ID setting, 259
 MT_AWS_REGION setting, 259
 MT_AWS_SECRET_ACCESS_KEY setting, 259
 MT_BAIDU_ID setting, 259
 MT_BAIDU_SECRET setting, 259
 MT_DEEPL_API_VERSION setting, 260
 MT_DEEPL_KEY setting, 260
 MT_GOOGLE_CREDENTIALS setting, 260
 MT_GOOGLE_KEY setting, 260
 MT_GOOGLE_LOCATION setting, 260
 MT_GOOGLE_PROJECT setting, 260
 MT_MICROSOFT_BASE_URL setting, 260
 MT_MICROSOFT_COGNITIVE_KEY setting, 261
 MT_MICROSOFT_ENDPOINT_URL setting, 261
 MT_MICROSOFT_REGION setting, 261
 MT_MODERNMT_KEY setting, 261
 MT_MODERNMT_URL setting, 261
 MT_MYMOMETRY_EMAIL setting, 261
 MT_MYMOMETRY_KEY setting, 261
 MT_MYMOMETRY_USER setting, 261
 MT_NETEASE_KEY setting, 262
 MT_NETEASE_SECRET setting, 262

MT_SAP_BASE_URL setting, 262
 MT_SAP_PASSWORD setting, 263
 MT_SAP_SANDBOX_APIKEY setting, 263
 MT_SAP_USE_MT setting, 263
 MT_SAP_USERNAME setting, 263
 MT_SERVICES setting, 258
 MT_TMSERVER setting, 262
 MT_YANDEX_KEY setting, 262
 MT_YOUDAO_ID setting, 262
 MT_YOUDAO_SECRET setting, 262
 NEARBY_MESSAGES setting, 263
 PAGURE_CREDENTIALS setting, 263
 PAGURE_TOKEN setting, 264
 PAGURE_USERNAME setting, 263
 PHP strings file format, 59
 PIWIK_SITE_ID setting, 258
 PIWIK_URL setting, 258
 PO file format, 54
 post() (*wlc.Weblate* [\[?/?/?/?/?\]](#)), 112
 post_add() (*weblate.addons.base.BaseAddon* [\[?/?/?/?/?\]](#)), 322
 post_commit() (*weblate.addons.base.BaseAddon* [\[?/?/?/?/?\]](#)), 322
 post_push() (*weblate.addons.base.BaseAddon* [\[?/?/?/?/?\]](#)), 322
 post_update() (*weblate.addons.base.BaseAddon* [\[?/?/?/?/?\]](#)), 322
 pre_commit() (*weblate.addons.base.BaseAddon* [\[?/?/?/?/?\]](#)), 322
 pre_push() (*weblate.addons.base.BaseAddon* [\[?/?/?/?/?\]](#)), 322
 pre_update() (*weblate.addons.base.BaseAddon* [\[?/?/?/?/?\]](#)), 322
 pull wlc [\[?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?\]](#), 110
 push wlc [\[?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?\]](#), 110
 pushgit weblate admin command, 292
 pushgit [\[?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?\]](#) --force-commit, 292
 Python, 112
 Qt file format, 58
 RATELIMIT_ATTEMPTS setting, 264
 RATELIMIT_LOCKOUT setting, 264
 RATELIMIT_WINDOW setting, 264
 RC file format, 65
 register_command() (*wlc.main* [\[?/?/?/?/?\]](#)), 113
 REGISTRATION_ALLOW_BACKENDS setting, 264
 REGISTRATION_CAPTCHA setting, 265
 REGISTRATION_EMAIL_MATCH setting, 265
 REGISTRATION_OPEN setting, 265
 repo wlc [\[?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?\]](#), 110
 REPOSITORY_ALERT_THRESHOLD setting, 265
 REQUIRE_LOGIN setting, 265
 reset wlc [\[?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?\]](#), 110
 REST, 73
 RESX file format, 62