

The Weblate Manual [?]????? 4.4.2

Michal Čihař

2021?01?14?

Weblate ???

??:

../devel/integration

??

????????

anguage Perferences Preferences I while completed translations on the dashboard Fundations Translation editor mode To bootom To bootom To boot	Weblate Dashboard	Projects 🗸 Lang	guages 🗸	Checks 🗸				ع	+ Add -	•	•••
Preferences Preferences I due do completed translations on the dashboard Translation editor mode I due do completed translations on the dashboard Top to otom Top to otom Comport of parby strings to show in each direction in the full editor. 2 show secondary translation sin the Zen mode 1 due source if a secondary translation sinsts 2 show secondary translation second	Your profile										
	Languages Preferences	Notifications	Account	Profile	Licenses	Audit log	API access				
Translation editor mode fulleditor fulleditor Top to bottom Top to bottom Sumber of nearby strings 15 Stow secondary translations in the Zen mode Ibide source if a secondary translation exists Editor tink Top to bottom 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 four 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 or our keyboard. Source if a secondary translations Component list Suggested translations Source	Preferences										G
Fulleditor Fulleditor Care editor mode Top to bottom Top to bottom Sumber of nearby strings 15 Sumber of nearby strings to show in each direction in the full editor. 2 show secondary translations in the Zen mode I lide source if a secondary translation exists Editor link I string 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 fou 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 or your keyboard. Porered by Webble to be source code. You can use (branch) for branch, (filename) and (line) as filename and line placeholders. Special characters	Hide completed translations	s on the dashboard									
Ten editor mode Top to bottom Sumber of nearby strings 15 Show secondary translations in the Zen mode I 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	Translation editor mode										
Top to bottom	Full editor										~
Number of nearby strings 15 16 17 Show secondary translations in the Zen mode 18 19 19 19 19 19 19 10 10 10 11 11 11 12 13 14 15 16 17 18 19 19 19 10 10 10 11 10 11 12 13 14 15 15 16 17 18 19 19 19 10 10 11 12 13 14 15 15 16 17 18 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 10 10 10 10 10 10 10 10 10 10 10 <tr< td=""><td>Zen editor mode</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr<>	Zen editor mode										
15 Show secondary translations in the Zen mode I show secondary translations in the Zen mode I shide 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 fou 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 or your keyboard. Special dashboard view Watched translations Component list Suggested translations Component list Stage Save	Top to bottom										~
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 cettor link Cettor link Content of the source code. You can use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders. Sepecial characters Vou 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 or vour keyboard. Pefault dashboard view Watched translations Component list Component l	Number of nearby strings										
Show secondary translations in the Zen mode Hide source if a secondary translation exists ditor link The source if a secondary translation exists ditor link Secondary translation exists Consponent lists Component list Suggested translations Performed by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate	15										
Hide source if a secondary translation exists Additor link Additor li	Number of nearby strings to sho	w in each direction i	in the full edi	itor.							
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 fou 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 or por keyboard. Default dashboard view Watched translations Component list Suggested translations Component list Save Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate	Show secondary translation	s in the Zen mode									
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 fou 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 or your keyboard. Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate	Hide source if a secondary tr	ranslation exists									
Special characters fou 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 or your keyboard. Default dashboard view Watched translations Component list Component list Suggested translations Default component list Save Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate	Editor link										
Special characters fou 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 or your keyboard. Default dashboard view Watched translations Component list Component list Suggested translations Default component list Save Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate											
fou 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 or your keyboard. Default dashboard view Watched translations Component list Component list Suggested translations Default component list Save Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate	Enter a custom URL to be used a	s link to the source of	code. You can	use {{branch}	} for branch, {{	filename}} and	{{line}} as filename a	and line p	aceholders.		
your keyboard. Default dashboard view Watched translations Component list Component list Suggested translations Performed by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate	Special characters										
your keyboard. Default dashboard view Watched translations Component list Component list Suggested translations Performed by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate											
Watched translations Component list Component list Suggested translations	You can specify additional special your keyboard.	l visual keyboard cha	aracters to b	e shown while	translating. I	can be useful i	or characters you us	se frequer	itly, but are h	ard to typ	e on
Component lists Component list Suggested translations	Default dashboard view										
Component list Suggested translations Default component list Compo	Watched translations										
Save Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate	O Component lists										
Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate	 Component list Component dataset 										
Save Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate											
Save Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate											
Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate											~
Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate											
	Save										
ากกลากกลาก		Powered by We	eblate 4.4 Ab	out Weblate	egal Contact	Documentation	Donate to Weblate				
	222222222222										

??:	settings.py	????????????????????????????????????	22???	SINGLE_PROJECT	?????	Weblate
????????????????????????????????????	<u>?????????????????????????????????????</u>	2????????????????????????????????????	?????????	2222222222222222222	????????????????????????????????????	????????????????????????????????????

???????

????

@ ^	l eblate Dashboard Projects - Languages -	Checks 🗸			🔧 🕂 Add 🗸	
🕐 Da	shboard					
Wato	hed translations 10 Suggested translations 5	Insights 🗸	Search			
	Component	Translated	Untranslated	Untranslated words	Checks Suggestions	Comments
1	WeblateOrg/Android — Czech 🛎 🖘 🕅 🗉	76%	3	3		
	WeblateOrg/Django — Hebrew 🛎 😋 GPL-3.0	92%	2	15		
-	WeblateOrg/Django — Czech 👾 🖘 GPL-3.0	96%	1	12	3	
/	WeblateOrg/Django — Hungarian 🛎 😄 GPL-3.0	69%	8	109	1	
/	WeblateOrg/Djangojs — Hungarian ≝⇔ GPL-3.0	96%	2	6		
1	WeblateOrg/Djangojs — Hebrew 坐 🕫 GPL-3.0	~				
/	WeblateOrg/Djangojs — Czech 🛎 😋 GPL-3.0	~				
/	WeblateOrg/Language names — Hebrew 坐 🖪 GPL-3.0	~				
	WeblateOrg/Language names — Czech 坐 🖪 GPL-3.0	~				
/	WeblateOrg/Language names — Hungarian 坐 🚦	81%	4	5		

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

????

Weblate Dasi	nboard Projects -	Languages 🗸 Checks 🗸				٩	+ Add -	- 🏶	•••
🛞 WeblateOrg / Djar	ngo / Czech / Transl	ate						translate	d 96%
IK K 1/26	> >I All strin	ngs 🗸		Position and priority -	E1			🔻 Zen	幸
Translation				Ð		Glossa	ary		
Hebrew				קבצים		English No related	C: d strings found	zech in the glosso	ıry.
English Files				Ū		+ Add	term to glo	ssary	
Czech Soubory		ſ	도 Clone source 🔄	← NBS " " , '		Source	e informatio	on	(i)
 Needs editing () 				7/100			hot context	y associated	. 🖍
Save 🔋	Suggest 🕨	Skip				Explana No explo	tion Ination current	ly provided.	/
Nearby strings 16	Comments A	utomatic suggestions Oth	nerlanguages	History		Labels No label	ls currently set.		
Language	Status	Translation	Edit			Flags No flags	currently set.		-
Hebrew	S	קבצים	Ec	dit			string locatio		
Hungarian	S	Fájlok	Ed	dit			e/templates/tr e/trans/forms.		ml:45
English	۵	Files	E	dit		String a 22 seco			
						Source 22 seco	string age nds ago		
							tion file /locale/cs/LC string 1	_MESSAGES	/djan

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

??:

2222222 2222222

22222 editor://open/?file={{filename}}&line={{line}} 222222

??:

??

Weblate Dashboard Project:	s 🗸 🛛 Languages 🗸	Checks 🗸				٩	+ Add -	• -	•••
Your profile									
Languages Preferences Notif	cations Accour	nt Profile	Licenses	Audit log	API access				
Watched projects									Û
Watched projects									
Search			*						
Available:			Chose						
WeblateOrg			Weblat	eorg					
			Ŧ						÷
You can receive notifications for watched Add all projects you want to translate to				efault.					
Aut an projects you want to transface to	see them as watched	projects on the	e dastiboard.						
Save									
Notification settings									G
		• -							
Other projects Watched project	s Managed proj	ects							
Component wide notifications		de e d'an un in altre							
You will receive a notification for every su Repository failure	Do not notify	ned projects.							~
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 notificat	ion							~
Mentioned in comment	Instant notificat	ion							~
Newlanguage	Do not notify								~
New translation component	Do not notify								~
New announcement	Instant notificat	ion							~
New alert	Do not notify								~
Translation notifications	bonochochy								
You will only receive these notifications f	or your translated la	nguages in your	watched projec	ts.					
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								~
_									

?????

 Image: Contract of the contract of the

@ \	Veblate	Dashboard	Projects 🗸	Languages 🗸	Checks -				+ Add -	•	•••
🔔 Yo	our profile										
Lan	guages	Preferences	Notificatio	Account	Profile	Licenses	Audit log	API access			
Acc	ount										Û
Use	rname										
te	stuser										
Use	rname ma	/only contain le	tters, numbers	or the following	characters: @ .	+					
Full	name										
W	eblate Tes	t									
E-m	ail										
w	eblate@e	ample.org									~
You	can add an	other e-mail add	lress below.								
Your	name and	e-mail will appe	ear as commit a	uthorship.							
	Save										
	rent user i	dentities									0
Ident	ity			UserID			Action	n			
Ô	Passwo	ord		testuser			Cł	nange password			
\sim	E-mail			weblate@examp	le.org		Di	sconnect			
G	Google)		weblate@examp	le.org		Di	sconnect			
	GitHub			123456			Di	sconnect			
	Bitbuc	ket		weblate			Di	sconnect			
Add	new asso	ociation									
	~										
E-r	mail										
Ren	noval										
Acco	ount remo	/al deletes all yo	ur private data.								
	Remove n	ny account									
Use	er data										
You	can downlo	oad all your priva	ite data.								
	Download	luser data									

API ????

????

Weblate ??????

????????

@ ^	Veblate Dashboard Projects -	Languages - Chec	<s <del="">-</s>				~ + A	\dd - 💮 - •••
💮 W	eblateOrg							translated 85%
Com	ponents Languages Info	Search Glossaries	Insights -	Files 🕶	Tools 🕶	Manage 👻	Share 🚽	O Not watching -
	Component	Translated Ur	translated	Untrans	ated words	Checks	Suggestions	Comments
لم	Android 🕊 😔 MIT	79%	30		30	3		
٦	Language names 🔟 🚦 GPL-3.0	95%	4		5			
Ad	d new translation component							

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

?????

Web	blateOrg / Diango / Czach							+ Add -	translated
wei	blateOrg / Django / Czech								translateu
verv	view Info Search Gloss	sary Insights -	Files 🕶	Tools 🕶	Manage 👻	Share 👻			Watching
rans	slation status								
. 6 S	Strings		969	%					
33 V	Words		930	%					Translate
string	gs status								
5	All strings — 183 words								
•	Translated strings — 171 words								
L 🛑	Strings needing action – 12 words	i							
L 🛑	Not translated strings – 12 words	i							
L	Strings needing action without sug	ggestions — 12 words							
3 📕	Strings with any failing checks — 1	1 words							
3 📕	Translated strings with any failing	g checks — 11 words							
L	Failed check: Unchanged translation	on — 4 words							
L	Failed check: Mismatched full stop	— 4 words							
L 🛑	Failed check: Python format — 3 w	ords							
)ther	r components								
	Component	Translated	Untranslate	ed U	ntranslated wor	ds Check	s Sugg	gestions	Comment
	Android 🔟 👄 MIT	76%		3		3			
	Language names 🔟 🚺 GPL-3.0	✓							
	Djangojs 🔟 🕞 GPL-3.0	×							

??

????

??:

report-source

???

variants

???

labels

??

???

??:

🜒 Weblate Dashboard Projects 🗸 Languages 🗸 Checks	•			ع	+ Add -	- 🏶	•••
📡 WeblateOrg / Django / Czech / Translate						translate	d 96%
I 1/1 > >I Custom Search - '%(count)s word	1	Position and priority -	E1			🖡 Zen	
Translation		Ø		Glossa	У		
English				English	C	zech	
Singular %(count)s word		Ū			strings found		ıry.
Plural		-11		C Add	erm to gio:	ssary	
%(count)s words		Ō		Sourco	informatio	20	0
Czech, One ①	도 Clone source 🔄	← NBS " " , '					0
%(count)s slovo		15/140			not context hishot currentl	y associated	
Czech, Few	🕒 Clone source 🛱	15/140 ← NBS ,, " , "		Explanat	ion		/
%(count)s slova					nation current	ly provided.	
Czech, Other 🛈	다 Clone source 🛱	15/140 ← NBS ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		Labels No labels	currently set.		1
%(count)s slov				Flags			
Plural formula: (n==1)?0: (n>=2 && n<=4)?1:2		14/140		Flags python-f	ormat		
Needs editing ()				Source s	tring locatio	'n	
Save 📮 Suggest 🕞 Skip				weblate, 9	templates/tr	anslation.ht	ml:14
				String a			
Nearby strings 20 Comments Automatic suggestions	Other languages	History		10 secor	ds ago		
New comment				Source s	tring age ds ago		
New comment				Translat	ion file		
Comment on this string for fellow translators and developers to read.				weblate, go.po, si	locale/cs/LC_ ring 5	_MESSAGES	/djan
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							

22222222222222222222222222222222222222	222 222 222 222 222 222 222 222 222 222 222 222 222 222 222 222 222 222 222 222
Cmd ↑ Alt PageDown or Ctrl ↓ or Alt ↓ or	22222222222222
Cmd ↓ Alt Enter or Ctrl Enter or	2022222222
Cmd Enter Ctrl Shift Enter or	2022222222222222222
Cmd Shift Enter Ctrl E or	222222222222222222
Cmd E Ctrl U or	2222 22222222222222222
Cmd U Ctrl M or	<u>[2][2][2][2][2][2][2][2][2][2][2][2][2][</u>
Cmd M Ctrl 1 []Ctrl 9 or	
Cmd 1 [?] Cmd 9	
Ctrl M1[2]9 or Cmd M1[2]9	222222222222222222222222222222222222222
Ctrl I1[]9 or Cmd I1 to 9	222222222222222222222222222222222222222
Ctrl Jor	2222222 22222
Cmd J Ctrl Sor	2222222222222222
Cmd S Ctrl O or	2022220222
Cmd O	
Ctrl Y or Cmd Y	222 2222222

SPECIAL_CHARS ?????????

Weblate Dashb	oard Projects					٩	+ Add -		•••
🛞 WeblateOrg / Django	o / Hebrew / Tr	anslate						translate	d 92%
IK K 1/26 >	>I All st	rings 🕶		Position and priority -	£1			🖡 Zen	랆
Translation				9		Glossa	iry		
English Files				Ō		English No related	Het d strings found	rew in the glossa	ry.
Hebrew	फ्र ← NBS ⁻	ZWNJ ZW	U LRM RLM LRE RLE	PDF LRO RLO RTL LTR		🕂 Add	term to glos	ssary	
				קבצים		Source	e informatio	on	(i)
Needs editing ()				5/100			hot context	y associateo	
Save Save	Iggest	▶▶ Skip				Explana No expla	tion Ination current	ly provided.	
Nearby strings 16	Comments	Automatic suggestions 0	ther languages	History		Labels No label	's currently set.		
Language	Status	Translation	Edit			Flags			
Czech	\bigcirc	Soubory	Ed	lit		No flags	currently set.		
Hungarian	0	Fájlok	Ed	lit		weblate	s tring locatio /templates/tr /trans/forms.j	anslation.ht	ml:45
English	۵	Files	Ed	lit		String a 24 seco			
						Source : 25 seco	string age nds ago		
							tion file /locale/he/LC_ tring 1	_MESSAGES	/djan

?????????

22222 ID22222222 msgctxt 22222222

????

???????????

Bilingual and monolingual formats 2?????????

???

???????

	Projects - Language	es 🛨 Checks 🛨			🔧 🕂 Ad	d 🕶 🍈 👻	•••
WeblateOrg / Glossaries							
Components Languages	Info Search	Glossaries Insights	s → Files → Tools →	Manage 🗸	Share -		
WeblateOrg							(i)
Catalan	1 Dutch 0 E	nglish 0 Fransh	Calisian C	rmon o lieb			
Catalan 0 Czech		nglish 0 French		erman 0 Heb	rew 0 H	ungarian 0	
	Chinese (Sim	plified) 0 Polish	0 Russian 0 Sp	oanish 0			
C 1							
Glossary name WeblateOrg							
Color Navy Blue Aqua Teal Ol	ive Green Lime Yell	ow Orange Red Mar	oon <mark>Fuchsia Purple</mark> Bla	ck Gray Silver			
Source language							
English							~
Additional projects							
Search							
		*	Chasar				A
Available:			Chosen:				
Choose additional projects wher	re this glossary can be use	₹					Ŧ
	e this glossary can be use	÷					¥
Choose additional projects wher	e this glossary can be use	÷				Delet	.e
Save	e this glossary can be use	ed.				Delet	.e
	e this glossary can be use	ed.				Delet	.e
Save	e this glossary can be use	ed.				Dele	e e
Save Create new glossary	e this glossary can be use	ed.				Delet	.e
Save Create new glossary Glossary name Color		ed.				Delet	.e
Save Create new glossary Glossary name Color Navy Blue Aqua Teal Ol	re this glossary can be use		oon Fuchsia Purple Biz	ck Gray Silver		Delet	e
Save Create new glossary Glossary name Color Navy Blue Aqua Teal Ol Source language			oon Fuchsia Purple Biz	ck Gray Silver		Delet	
Save Create new glossary Glossary name Color Navy Blue Aqua Teal Ol			oon Fuchsia Purple Bla	ck Gray Silver		Dele	
Save Create new glossary Glossary name Color Navy Blue Aqua Teal Ol Source language			oon Fuchsia Purple Biz	ck Gray Silver		Delet	
Save Create new glossary Glossary name Color Navy Blue Aqua Teal Ol Source language English		ow Orange Red Mar	oon <mark>Fuchsia Purple Bia</mark>	ck Gray Silver		Dele	
Save Create new glossary Glossary name Color Navy Blue Aqua Teal OI Source language English Additional projects			oon Fuchsia Purple Bis	ck Gray Silver			
Save Create new glossary Glossary name Color Tavy Blue Aqua Teal OI Source language English Additional projects Search		ow Orange Red Mar		ck Gray Silver		Delet	
Save Create new glossary Glossary name Color Tavy Blue Aqua Teal OI Source language English Additional projects Search		ow Orange Red Mar		ck Gray Silver		Delet	
Save Create new glossary Glossary name Color Tavy Blue Aqua Teal OI Source language English Additional projects Search		ow Orange Red Mar		ck Gray Silver			
Save Create new glossary Glossary name Color Tavy Blue Aqua Teal OI Source language English Additional projects Search		ow Orange Red Mar		ck Gray Silver			
Save Create new glossary Glossary name Color Navy Blue Aqua Teal OL Source language English Additional projects Search Available:	ive Green Lime Yell	ow Orange Red Mar		ck Gray Silver		Delet	
Save Create new glossary Glossary name Color Tavy Blue Aqua Teal OI Source language English Additional projects Search	ive Green Lime Yell	ow Orange Red Mar		ck Gray Silver			
Save Create new glossary Glossary name Color Navy Blue Aqua Teal OL Source language English Additional projects Search Available:	ive Green Lime Yell	ow Orange Red Mar		ck Gray Silver			

()) Weblate	Dashboard Projects - Lang	lages 🗸 Checks 🗸		٩	+ Add -	•	•••
B WeblateOrg / Czech / Glossary							
Browse	dd new word Import glossary	Export glossary 🗸	History				
Search							
					Startingle	etter	Any 🗸
Source	Translation	Glossary					
language	jazyk	WeblateOrg	Edit Delete				

????

????

()) Weblate	Dash	board P	rojects -	Languages 🗸	Checks 🗸				ع	+ Add -		•••
🛞 WeblateOr	WeblateOrg / Django / Czech translated 96%											
Overview	Info	Search	Glossary	Insights +	Files 🕶	Tools -	Manage 🗸	Share -			⊙ Watch	ing -
						Data expo	orts					
Automatic t	ranslatio	n				Automati	c translation					Û
				ons in this projec anslate a new co		Search an Bulk edit	d replace	ent. It can be used to	push transl	ations to a	different	
Automatic tra	anslation	via machine	translation	uses active mac	hine translatio	on engines to	get the best p	oossible translations a	nd applies t	hem in this	project.	
Automatic tr	anslatio	nmode										
Add as sugg	gestion											~
Search filter												
Strings nee	ding acti	on										~
Automatic tr	anslatio	n source										
○ Other tran	slationco	omponents	Machine	translation								
Machine trar	slation	engines										
Search												
Available:						Chos	sen:					*
Weblate						Web	ate					
Weblate T	ranslatio	n Memory										
						Ŧ						Ŧ
	Score threshold											
80	80											
Apply												

?????? 2 ????????:

Keeping translations same across components

????

?????

????

??:

Bulk edit addon

??????????

♥ Weblate Dashboard Projects - Languages - C ◎ WeblateOrg / Django / Czech	ihecks -	🔧 🕂 Add	+ 🎲 + •••
Overview Info Search Glossary Insights -	Files - Tools - Manage - Share -		⊙ Watching -
Upload	Download original translation file (gettext PO file)		
The uploaded file will be merged with the current translation. In File () Choose File No file chosen	Download translation file as CSV Download translation file as gettext MO Download translation file as gettext PO Download translation file as TBX	rn it on.	
File upload mode ① O Add as translation O Add as suggestion O Add as translation needing edit	Download translation file as TMX Download translation file as XLIFF with gettext extensions Download translation file as XLIFF 1.1 Download translation file as Excel Open XML		
 Replace existing translation file Processing of strings needing edit ① 	Download strings needing action as CSV Download strings needing action as gettext MO		
Do not import Conflict handling ①	Download strings needing action as gettext PO Download strings needing action as TBX Download strings needing action as TMX		~
Update translated strings Whether to overwrite existing translations if the string is already Author name ①	Download strings needing action as XLIFF with gettext extensions Download strings needing action as XLIFF 1.1 Download strings needing action as Excel Open XML		~
Weblate Test	Customize download		
Author e-mail 🗊	Upload translation		
weblate@example.org		_	
Upload			

Supported file formats

Supported file formats

????????

??:

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

?????????

????????

???????

?????

w w	eblate Dashboard Projects •	Languages 🗸	Checks 🗸			🔦 🕂 Add	- 💮- •••
😗 We	blateOrg / Django / Czech						translated 96%
The t	ranslation has come to an end.						
Over	view Info Search Glossa	ary Insights -	Files 👻 🛛 Tools	▼ Manage ▼	Share 🛨		⊙ Watching -
Tran	slation status						
26	Strings		96%				Translate
183	Words		93%				Translate
Strin	gs status						Ū
26	-						U
25	Translated strings — 171 words						
1	Strings needing action – 12 words						
1	Not translated strings — 12 words						
1		gestions — 12 words					
3	Strings with any failing checks — 11	words					
3	Translated strings with any failing o	checks — 11 words					
1	Failed check: Unchanged translation	n — 4 words					
1	Failed check: Mismatched full stop -	– 4 words					
1	Failed check: Python format — 3 wo	rds					
Othe	r components	Translated	Universitated	Unbergelahed over	h Charles	Currentleur	Community
	Component	Translated	Untranslated	Untranslated word		Suggestions	Comments
	Android 🕍 🖘 MIT	76%	3		3		
1	Language names 坐 🖪 GPL-3.0	×					
1	Djangojs 🕊 😋 GPL-3.0	~					
В	rowse all components						

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

????

AUTOFIX_LIST

????

???????

BBcode ??????

???????

???????

???????

🕑 Weblate Dashboard Projects - Languages - Checks -	🔍 🕂 Add 🗕 💮 – 🚥
😚 WeblateOrg / Django / Czech / Translate	translated 96%
I 1/1 > >I Custom Search ~ '%(count)s word' Position and priority ~ ±I	↓ F Zen ∃
Translation co	Glossary
English	English Czech
Singular	No related strings found in the glossary.
%(count)s word	+ Add term to glossary
Plural	
%(count)s words	
Czech, One 🛈	Source information ①
%(count)s slovo	Screenshot context
15/140	No screenshot currently associated.
Czech, Few () 🛱 🕂 NBS , " , ' - –	Explanation 🧨
%(count)s slova	No explanation currently provided.
15/140	Labels
Czech, Other 🕧 🖫 🖬 🖬 NBS " " , '	No labels currently set.
%(count)s slov	
Plural formula: (n==1)?0: (n>=2 && n<=4)?1:2()	Flags 🖍
🗌 Needs editing 🗊	
	Source string location weblate/templates/translation.html:14
Save 📭 Suggest 🕞 Skip	9
	String age
Nearby strings 20 Comments Automatic suggestions Other languages History	10 seconds ago
History	Source string age
No matching activity found.	11 seconds ago
Browse all component changes	Translation file
	weblate/locale/cs/LC_MESSAGES/djan
	go.po, string 5

AngularJS ?????

??:

AngularJS: API: \$interpolate

C????

??:

C format strings ? C printf format

C# ????

??:

C# String Format

??:

Template literals

i18next ??

i18next 2222222222 22222 4.0 2222.

??:

i18next interpolation

Java ??

??:

Java Format Strings

Java MessageFormat

??:

Java MessageFormat

JavaScript ??

??:

JavaScript formatting strings

Perl ??

??:

Perl sprintf ?Perl Format Strings

PHP ??

??:

PHP sprintf documentation ? PHP Format Strings

Python ???????

??:

Python brace format? Python Format Strings

Python ??

??:

Python string formatting Python Format Strings

Qt ??

Qt 2222222222222222

??:

Qt QString::arg()

Qt ????

??:

Qt i18n guide

Ruby ??

??:

Ruby Kernel#sprintf

Vue I18n ??

??:

Vue I18n Formatting ? Vue I18n Linked locale messages

????

???????

??:

Keeping translations same across components

Kashida ?????

Markdown ????

Markdown ????

Markdown ??

?????????

?????? 3.7 ????.

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

??:

22222222**?**222222222222**?**222222222

\n **????**

????????

??? Wikipedia ??

?????????

??:

????? Wikipedia ???

???????

 53

 53

???? Wikipedia ???

???????

 55:

 53:

???? Wikipedia ???

???????

???? Wikipedia ???

??????????

??:

?????? Wikipedia ???

???????

22222222222222

???????

?????????

?????? 3.9 ????.

22222 4.3 222: 22222222222222222222222

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

placeholders:r"%[^%]%"

??:

?????????

????? 3.9 ????.

22: 22:

????

regex:^foo|bar\$

???????

?????

??:

????

?????

??:

??????

?????

???????

22222222222

??:

????? HTML

22222222222 HTML 2222222222222

?????? 3.9 ????.

??:

URL

?????? 3.5 ????.

XML ??????

XML ??

22222222 XML 22222 22222 2.8 222 XML 222222222222

???????

????

????

???????

????????

???????

from gettext import ngettext

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

??

?????? 3.9 ????.

Weblate Dashboard Projects -	Languages 🗸 Che	cks -	+ Add •	۰	•••
② Dashboard					
Watched translations 0 Suggested	translations 0 Ins	ights → Search			
Search					0
Custom Search -	Sort	By+ EI			
Advanced query builder					
Source strings - Search for	Exact Ac	d String has suggestion - Add			
String changed after - mm/dd/yyyy	Add				
Query examples					
Review strings changed by other users		changed:>=2020-11-14 AND NOT changed_by:testuser		Add	
Translated strings		state:>=translated		Add	
Strings with comments		has:comment		Add	
Strings with any failing checks		has:check		Add	
Strings with suggestions from others		has:suggestion AND NOT suggestion_author:testuser		Add	
Approved strings with suggestions		state:approved AND has:suggestion		Add	
All untranslated strings added the past	month	added:>=2020-11-14 AND state:<=needs-editing		Add	
Translated strings in a certain language	e	is:translated AND language:cs		Add	
Search					

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

?????

????

plural[context[]suggestion[]comment[]check[]dismissed-????change ac-

??????

?????????

??????

?????

22222 4.4 222.

????

?????????

Weblate Dashboard	Projects 🗸 🛛 Langu	ages 🗕 Checks 🗕				عر	+ Add -	- 🏶	•••
😚 WeblateOrg / Django / Czec	ch / Translate							translate	ed <mark>96%</mark>
I< ≺ 1/1 → →	Custom Search -	'%(count)s word'		Position and priority -	E1			🔻 Zen	
Translation		crings• state:empty action• state: <translated< td=""><td></td><td></td><td></td><td>ossa</td><td>iry</td><td></td><td>/</td></translated<>				ossa	iry		/
Strings needing action • state: <translated< th=""> Adv English Translated strings • state:>=translated sh Czech Singular Strings marked for edit • state:needs-editing ated strings found in the glossary. %(count)s word Strings with suggestions • has:suggestion Add term to glossary. Plural Strings with variants • has:variant </translated<>				©					
%(count)s slova Czech, Other ①	Approved strings • state:approved strings • state:tapproved strings • state:tapproved strings • state:translated strings waiting for review • state:translated to the state st			1					
%(count)s slov Plural formula: (n==1)?0: (n>=2 Needs editing () Save ! Suggest	&& n<=4)?1:2① ►► Skip			14/140			format string locatic /templates/tr		/ tml:14
Nearby strings 20 Comm	ents Automatic	suggestions Other languag	es	History		String a 10 seco Source 11 seco	nds ago string age		
New comment Comment on this string for fello Scope	ow translators and de	velopers to read.					tion file /locale/cs/LC tring 5	_MESSAGES	/djan
Translation comment, discussions with other translators									
Is your comment specific to this New comment	translation or generi	c for all of them?							
You can use Markdown and menti	on users by @usernar	ne.							
Save									

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

????????

WeblacOrg / Django / Czech / Translate Rottranslated strings • state empty Position and priority • if I * Zen # Translation Position and priority • Distion Position # English Position Priority Labels Translation Priority Labels Cecch Needs editing • I Suggest • > Skip Number of railing checks Needs editing • I Suggest • > Skip Sorce information Needs editing • Automatic suggestions • Other languages History Needs of this for fellow translators and developers to read. Sorce Sorpe Translation comment, discussions with other translators I You can use Markdown and mention users by @username. Sing ge Store Sorce string se Sorce Sing ge Translation comment, discussions with other translators Sing ge Sorce string se Sing ge Store	₩eblate Dashboard Projects - Languages - Checks -		🔧 + Add - 🌐 - 🚥
Translation English The string uses three dots () instead of an ellipsis character () Czech Cacch Needs editing () Needs editing () Save Save Suggest Skip Nearby strings 12 Comments New comment Comment on this string for fellow translators and developers to read. scope Translation comment, discussions with other translators Sour comment discussions with other translators New comment You can use Markdown and mention users by @username. Save	🛞 WeblateOrg / Django / Czech / Translate		translated <mark>96%</mark>
Translation Position English Czech Czech Ciccons source Needs editing ○ Number of source Needs editing ○ Number of failing checks Key Source information Nearby strings 10 Comments Number of failing checks Key Source information New comment Comments and developers to read. Scope Translation or generic for all of them? New comment Source string location Source string lo	I 1/1 Image: State Strings - State State Strings - State St	Position and priority - 1	V Zen ⊒≟
Priority The string uses three dots () instead of an ellipsis character () Caech Caech Number of words Number of comments Number of failing checks Key Save Comments Automatic suggestions Other languages History Labels No kabels currently associated. No kabels currently set. Plantin No kabels currently set. Scope Translation comment, discussions with other translators Is your comment Source string location You can use Markdown and mention users by @username. Save	Translation		Glossary
Czech Age of string Number of words Number of fording checks Save Save <td>English</td> <td>Priority</td> <td>English Czech</td>	English	Priority	English Czech
Number of words Number of words Number of ownds Number of failing checks Key Source information Source information	The string uses three dots () instead of an ellipsis character ()	Labels	No related strings found in the glossary.
Number of failing checks Save Save Suggest > Skip Nearby strings 1 Comments Automatic suggestions Other languages History Beplanation New comment Comment, discussions with other translators Is your comment, discussions with other translators New comment Source string location Weblate/focale/cs/LC_MESSAGES/djag	Czech 🔄 Clone source	, , , ,	➡ Add term to glossary
Save Suggest >> Skip Nearby strings 16 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 Save Save Save Save Save Save Not call of the call o	Needs editing ①		Source information ①
Nearby strings 1 Comments No explanation currently provided. 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 📭 Suggest	Key	
Comment on this string for fellow translators and developers to read. No flags currently set. Scope Source string location Translation comment, discussions with other translators weblate/checks/source.py:54 Is your comment String age New comment 13 seconds ago You can use Markdown and mention users by@username. Source string age Translation file weblate/locale/cs/LC_MESSAGES/djagg	Nearby strings 16 Comments Automatic suggestions Other languages History No explanation currently provided. Labels No labels currently set.		
Translation comment, discussions with other translators Source string location Is your comment specific to this translation or generic for all of them? String age New comment String age You can use Markdown and mention users by@username. Source string age Translation file weblate/checks/source.py.54	Flags		
Translation comment, discussions with other translators weblate/checks/source.py:54 Is your comment specific to this translation or generic for all of them? string age New comment Is seconds ago You can use Markdown and mention users by @username. Source string age Translation file weblate/cbc_dkc_/LC_MESSAGES/djang	Scope		Source string location
New comment String age Image: String age 13 seconds ago You can use Markdown and mention users by @username. Source string age Image: Save 13 seconds ago Translation file weblate/locale/cs/LC_MESSAGES/djang	Translation comment, discussions with other translators	~	
New comment 13 seconds ago You can use Markdown and mention users by@username. Source string age 13 seconds ago Save Translation file weblate/locale/cs/LC_MESSAGES/djang	Is your comment specific to this translation or generic for all of them?		String age
You can use Markdown and mention users by@username. Save Save 13 seconds ago Translation file weblate/locale/cs/LC_MESSAGES/djang	New comment		
save weblate/locale/cs/LC_MESSAGES/djang			
	Save		weblate/locale/cs/LC_MESSAGES/djang

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

?????????

???????

 222/2/2/2
 1222/2/2/2
 1222/2/2/2
 1222/2/2/2
 1222/2/2/2
 1222/2/2
 1222/2/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 122/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 1222/2
 122/2
 122/2
 122/2

Translation states

????

????

Value Value

 202022
 off
 222222

 202022
 off
 222222

 202022
 off
 222222

 202022
 off
 2222222

 202022
 off
 22222222

 20202222
 0
 2222222222

 20202222
 0
 2222222222

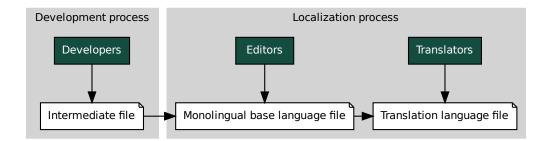
 20202222
 0
 222222222222

 2020222222222222222
 22222222222222

??????

???????

2222222222222222222222222222222222222	
😢 Weblate Dashboard Projects - Languages - Checks - 🔦 + Add - 💮	
WeblateOrg / Settings	
Basic Access Workflow Components	
Set "Language-Team" header Lets Weblate update the "Language-Team" file header of your project.	
✓ Use shared translation memory Uses the pool of shared translations between projects.	
Contribute to shared translation memory Contributes to the pool of shared translations between projects.	
Enable hooks Whether to allow updating this repository by remote hooks.	
Language aliases	
Comma-separated list of language code mappings, for example: en_GB:en,en_US:en	
Enable reviews	
Requires dedicated reviewers to approve translations.	
Enable source reviews	
Requires dedicated reviewers to approve source strings.	
Save	
Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate	
Weblate 2022222022222022222222222222222222222	Weblate



?????

??:

Bilingual and monolingual formats????????????abels

??????

??

??:

```
# 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/master
```

(?????????)

```
# 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
```

```
# 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 master branch:
git checkout master
git merge weblates-second/master
... # Resolve conflicts
git commit
# Push changes to the upstream repository, Weblate will fetch the merge_
\hookrightarrow from there:
git push
```

```
# Add remote:
git remote add weblate /path/to/weblate/snapshot/
# Update Weblate remote:
git remote update weblate
# Merge Weblate changes:
git merge weblate/master
# 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.
\hookrightarrow there:
git push
```

??:

Once you merge changes from Weblate, you might have to merge these branches (depending on your development workflow) discarding differences:

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 *Supported file 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:

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?

You can subscribe to any changes made in 22 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.

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.

??:

report-source???????

How can I use existing translations while translating?

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.

??:

<u>????</u>?????

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.

For monolingual files (see *Supported file formats*) Weblate might add new translation strings not present in the *template*, and not in actual translations. It does not however perform any automatic cleanup of stale strings as that might have unexpected outcomes. If you want to do this, please install an appropriate addon which will handle the cleanup according to your requirements.

Weblate also will not try to update bilingual files in any way, so if you need po files being updated from pot, you need to do it yourself or using an addon.

??:

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.

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.

[1]: 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 addon POT 22222 PO 222222 (msgmerge).

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 setup

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.

??

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.

Weblate also supports VCS less operation, see **2222 2222**.

??: 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.

??:

2222222222

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. If you feel your project should be translated as one, consider merging these po files. It will make life easier even for translators not using Weblate.

??: In case there is great demand for this feature, it might be implemented in future versions.

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.

Supported 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.

??:

Translation Related File Formats

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.

Additionally this workflow can be extended by utilizing 2022/2022 to include strings provided by developers, but not to be used as is in the final strings.

????

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	Linguality ¹	Plurals ²	Comments ³	Context ⁴	Location ⁵	Flags ⁸	Additional states ⁶
GNU gettext Monolingual gettext	bilingual mono	yes yes	yes yes	yes yes	yes yes	yes ⁹ yes ⁹	needs editing needs editing
XLIFF	both	yes	yes	yes	yes	yes ¹⁰	needs editing, approved
Java proper ties	r-both	no	yes	no	no	no	approved
	r-mono	yes	yes	no	no	no	
Joomla trans	s-mono	no	yes	no	yes	no	
Qt Linguist .ts	both	yes	yes	no	yes	yes ¹⁰	needs editing
Android strin resources	gmono	yes	yes ⁷	no	no	yes ¹⁰	_
Apple iO	Sbilingual	no	yes	no	no	no	
strings PHP [2][2][2]	mono	no ¹¹	yes	no	no	no	???????????????????????????????????????

			Table 1 - ??	?????????	2]?		
Format	Linguality ¹	Plurals ²	Comments ³	Context ⁴	Location ⁵	Flags ⁸	Additional states ⁶
JSON files	mono	no	no	no	no	no	
JSON i18ne	extmono	yes	no	no	no	no	
files							
go-i18n JSC	Nmono N	yes	no	no	no	no	
files							
ARB File	mono	yes	yes	no	no	no	
WebExtensio	on mono	yes	yes	no	no	no	
JSON						10	
.XML resour	cemono	no	yes	no	no	yes ¹⁰	
files	1 .1						1 1
CSV files	both	no	yes	yes	yes	no	needs editing
YAML files	mono	no	yes	no	no	no	
Ruby YAN	<i>IL</i> mono	yes	yes	no	no	no	
files							
DTD files	mono	no	no	no	no	no	
Flat XML	mono	no	no	no	no	yes ¹⁰	
	RCmono	no	yes	no	no	no	
files							1 1.
	enmono	no	yes	yes	yes	no	needs editing
XML	নানা						
	?.?mono	no	no	no	no	no	
<u>??????</u>							
????? Subtitle files							
Subtitle files	mono	no	no	no	yes	no	
HTML files OpenDocum	mono	no	no	no	no	no	
Format	emitorio	no	no	no	no	no	
IDML Form	at mono	no	no	no	no	no	
	a-mono	no	no	no	no	no	
tions	u-1110110	110	110	110	110	110	
Inno Setup I	NImono	no	no	no	no	no	
?????		110	110	110	110	110	

GNU gettext

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"
```

(?????????)

See *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.

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.

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

??:

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"
```

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

[2][2][2] 3.3 [2][2]: 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.

See ???????.

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.

??

Specifying translation flags

You can specify additional translation flags (see 2022/2022) 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 22222222).

Unit keys or context

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
????????	localizations/*.xliff
22222222222222	Empty
222222222222	localizations/en-US.xliff
???????	XLIFF Translation File

Typical Weblate Component co	onfiguration for monolingual XLIFF
File mask	localizations/*.xliff
222222222222222	localizations/en-US.xliff
???????????????????????????????????????	localizations/en-US.xliff
????????	XLIFF Translation File

??:

XLIFF Wikipedia ??? ?XLIFF? 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.

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

??:

GWT properties

Native GWT format for translations. GWT properties are usually used as monolingual translations.

??:

INI translations

22222 4.1 222.INI file format for translations.

INI translations are usually used as monolingual translations.

?!?: Weblate only extracts keys from sections within a INI file. In case your INI file lacks sections, you might want to use *Joomla translations* or *Java properties* instead.

??:

INI Files? Java properties? Joomla translations? Inno Setup INI ????

Inno Setup INI ????

??: The only notable difference to *INI translations* is in supporting <code>%n</code> and <code>%t</code> placeholders for line break and tab.

Typical Weblate Cor	mponent configuration
????????	language/*.islu
????????????????????????????????????	language/en.islu
????????????????????????????????????	Empty
???????	Inno Setup INI File

Characteristics : Only Unicode files (.islu) are currently supported, ANSI variant (.isl) is currently not supported.

??:

INI Files? Joomla translations? INI translations

Joomla translations

?????? 2.12 ????.

Native Joomla format for translations.

Joomla translations are usually used as monolingual translations.

??:

Specification of Joomla language files ? Mozilla and Java properties files? INI translations? Inno 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 co	onfiguration when using as bilingual
222222	i18n/app.*.ts
2222222222222	Empty
22222222222	i18n/app.de.ts
22222	Qt Linguist Translation File
ypical Weblate Component con	<i>ifiguration</i> when using as monolingual

 Typical Weblate Component configuration when using as monoling

 [2]22222
 i18n/app.*.ts

 [2]2222222222
 i18n/app.en.ts

 [2]2222222222
 i18n/app.en.ts

 [2]222222
 i18n/app.en.ts

 [2]22222
 Qt Linguist Translation File

??:

Qt Linguist manual 2Qt .ts? Bilingual and monolingual formats

Android string resources

Android specific file format for translating applications.

??:

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:

become:

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.

??:

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';

| lang/^/texts.pnp |
|-------------------|
| 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 ????.

2222 2.16 222: Since Weblate 2.16 and with translate-toolkit at-least 2.2.4, nested structure JSON files are supported as well.

[2][2][2] 4.3 [2][2]: 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. The only difference between them is when adding new strings. The nested variant tries to parse the key and insert the new string into the matching structure.

| Typical Weblate Component configuration | | |
|---|----------------------------|--|
| ???????? | langs/translation-*.json | |
| ???????????????????????????????????? | langs/translation-en.json | |
| ???????????????????????????????????? | Empty | |
| ??????? | JSON nested structure file | |

??:

JSON i18next files

2222 2.17 222: 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.

Weblate supports the i18next JSON v3 format. The v2 and v1 variants are mostly compatible, with exception of how plurals are handled.

Example file:

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

| Typical Weblate Cor | mponent configuration |
|--------------------------------------|-----------------------|
| ???????? | langs/*.json |
| ???????????????????????????????????? | langs/en.json |
| ???????????????????????????????????? | Empty |
| ??????? | i18next JSON file |

??:

go-i18n JSON files

2????? 4.1 ????.

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

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.

??:

ARB File

2????? 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 | | | | | | | | |
|---|----------------------|--|--|--|--|--|--|--|
| ???????? | lib/l10n/intl_*.arb | | | | | | | |
| ???????????????????????????????????? | lib/l10n/intl_en.arb | | | | | | | |
| ???????????????????????????????????? | Empty | | | | | | | |
| ??????? | ARB file | | | | | | | |

??:

JSON Application Resource Bundle Specification Internationalizing Flutter apps JSON

WebExtension JSON

 [2][2][2][2]
 2.16
 [2][2][2]
 2.16
 [2][2][2]
 2.16
 [2][2][2]
 2.16
 [2][2][2]
 2.16
 [2][2][2]
 2.16
 [2][2][2]
 2.16
 [2][2][2]
 2.16
 [2][2][2]
 2.16
 [2][2][2]
 2.16
 [2][2][2]
 2.16
 [2][2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]
 [2][2]

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"
  }
}
```

??:

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.

??:

.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 :guilabel: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 translate-toolkit: location, source, target, ID, fuzzy, context, translator_comments, developer_comments Choose :guilabel:CSV file as a file format.

Remember to define 2020202020202020 when your files are monolingual (see Bilingual and monolingual formats).

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 Compon | ent configuration for monolingual CSV |
|--------------------------------------|---|
| 222222 | locale/*.csv |
| 222222222222222222222222222222222222 | locale/en.csv |
| ???????????????????????????????????? | locale/en.csv |
| ??????? | [<u>2</u>]?]?]?] CSV [<u>2</u>]?]?] |

??:

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.*.yml | | | | | | |
| ???????????????????????????????????? | translations/messages.en.yml | | | | | | |
| ???????????????????????????????????? | Empty | | | | | | |
| ??????? | YAML file | | | | | | |

??: YAML**?***Ruby YAML files*

Ruby YAML files

[2][2][2][2]2.9[2][2][2]Ruby i18n YAML files with language as root node.Example Ruby i18n YAML file:

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

??:

YAML? YAML files

DTD files

??????? 2.18 ????.
Example DTD file:

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

??:

Mozilla DTD format

Flat XML files

?????? 3.9 ????.

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>
```

??: Flat XML

Windows RC files

22: 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

IDS_MSG2

IDS_MSG3

IDS_MSG4

"Hello, world!\n"

"Orangutan has %d banana.\n"

"Try Weblate at http://demo.weblate.org/!\n"

END
```

Typical Weblate Component configuration

	Lang/^.rC
????????????????????????????????????	lang/en-US.rc
????????????????????????????????????	lang/en-US.rc
???????	RC file

??:

Windows RC files

????? 3.5 ???.

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/*						
????????????????????????????????????	<pre>fastlane/android/metadata/en-US</pre>						
????????????????????????????????????	<pre>fastlane/android/metadata/en-US</pre>						
???????	App store metadata files						

???: In case you don't want to translate certain strings (for example changelogs), mark them read-only (see **?????????**). This can be automated by the **?????**.

Subtitle files

202222 3.7 2022. Weblate 2020202020202020202020 SubRip subtitle file (*.srt) MicroDVD subtitle file (*.sub) Advanced Substation Alpha subtitles file (*.ass) Substation Alpha subtitle file (*.ssa) Typical Weblate Component configuration2222222path/*.srt22222222222222path/en.srt2222222222222path/en.srt222222222subkip 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 <code>source</code> (which contains the source string) and a column called <code>target</code> (which contains the translation). Additionally there should be the column called <code>context</code> (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 files

22222 4.1 222.

22: 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

OpenDocument Format

22???? 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

2????? 4.1 ????.

22: 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.

???

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.

??:

Translation Related File Formats

??????????

22.2.2.2.3.10 2.2.2.

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 202020202020.

??????????

 Display="block-color: black-color: black-color:

SSH ?????

Weblate Da	shboard Proje	cts - Langua	ges •	Checks -				عر	+ Add -		•••
🔧 Manage / SSH ke	ys										
Weblate status	Backups Tra	anslation memo	ry	Performance report	SSH keys	Alerts	Repositories	Users	Appea	rance	
Tools Billing											
Public SSH key											Ū
Weblate currently us	ses this SSH key:										
ssh-rsa AAAAB3NzaC1yc2EA	AAADAQABAAACA	QDNVZp127qFR6	+GH/N	IrkrFuWxGntSccH0C+yA	.C36qXnZ85y+h1	Z6j26UeALr	ıjHI63Bu4xQz1XCZ	UpGsGNw	/KKdUMmq	3WmikXZ	g /
Download priv	ate key										
Known host keys											Ū
Hostname	Key type	Fing	erprint	t							
github.com	ssh-rsa	nTh	bg6kXl	JpJWGI7E1IGOCspRom	TxdCARLviKw6E5	SY8					
Add host key											0
To access SSH hosts	, its host key need	s to be verified. Ye	ou can g	get the host key by ente	ering a domain na	ame or IP fo	or the host in the f	orm belov	N.		
Hostname	ame	Port Por	t								
Submit											

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

Weblate SSH ??

SSH ?????????

Veblate ?????	???????????????????????????????????????	?!?!?!? SSH [????????????????????????????????????	????????????????????????????????????	?					
2222222222 SH 222222222222	????????	????????????????????????????????????	??? ??????????????????????????????????	<i>Add</i> 22222 222222	host ????????????????????????????????????	key gitl ????	ab.c	??? om????	?????? ????Su	? ? ? ıbmi
Weblate Das	shboard Projects -	Languages 🗸	Checks 🗸				عر -	+ Add -	. - •	•••
🔌 Manage / SSH key	rs									
Added host key for git	thub.com with fingerprin	t nThbg6kXUpJWG	I7E1IGOCspRomTxc	ICARLviKw6E5SY	8 (ssh-rsa), please	verify that it i	s correct			
Weblate status Tools Billing	Backups Translatio	n memory Pe	rformance report	SSH keys	Alerts Rep	ositories	Users	Appeara	ince	
Public SSH key										0
Weblate currently use	es this SSH key:									
	AAADAQABAAACAQDNVZ;	o127qFR6+GH/Nrkr	FuWxGntSccH0C+yA	C36qXnZ85y+h1	Z6j26UeALnjHI63E	3u4xQz1XCZUp	oGsGNwk	(KdUMmq3)	VmikXZg	
Download priva	ісе кеу									
Known host keys										0
Hostname	Keytype	Fingerprint								
github.com	ssh-rsa	nThbg6kXUpJ	WGI7E1IGOCspRom	TxdCARLviKw6E	5SY8					
Add host key										(i)
To access SSH hosts, i Hostname github.c	its host key needs to be v	erified. You can get	the host key by ente	ering a domain na	ame or IP for the H	nost in the for	m below.			

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

GitHub ?????

??:

Weblate ??? URL

HTTPS ?????

22222222222 GitHub URL 2222222222: https://user:your_access_token@github.com/ WeblateOrg/weblate.git?

????????

2222222222cURL documentation 2222222222 http_proxy? https_proxy????? all_proxy 22222222VCS 22222222222222222

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

 Image: The second state in the second state

??:

The cURL manpage ?Git config documentation

Git

??:

Git ???????

Git ?????????

 202222Bazaar
 2
 Mercurial
 202222
 GitHub
 2022222
 GitHub
 git-remote-bg
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2

bzr::lp:gnuhello

Mercurial ??????? selenic.com ? hello ??????????

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

GitHub

?????? 2.3 ????.

Pushing changes from Weblate

??:

GITHUB_USERNAME []GITHUB_TOKEN []GITHUB_CREDENTIALS

GitLab

?????? 3.9 ????.

??:

Pushing changes from Weblate

22222222222API 22222222222222

??:

GITLAB_USERNAME GITLAB_TOKEN GITLAB_CREDENTIALS

Pagure

?????? 4.3.2 ????.

??:

Pushing changes from Weblate

??:

PAGURE_USERNAME PAGURE_TOKEN PAGURE_CREDENTIALS

Gerrit

????? 2.2 ???.

Mercurial

??:

Subversion

?????? 2.8 ????.

Subversion ????

??:

DATA_DIR

??????????

????? 3.8 ????.

Weblate's REST API

The API is accessible on the /api/ URL and it is based on Django REST framework. You can use it directly or by *Weblate* 22222222.

Authentication and generic parameters

The public project API is available without authentication, though unauthenticated requests are heavily throttled (by default to 100 requests per day), so it is recommended to use authentication. The authentication uses a token, which you can get in your profile. Use it in the Authorization header:

ANY

Generic request behaviour for the API, the headers, status codes and parameters here apply to all endpoints as well. Query Parameters

format -- Response format (overrides Accept). Possible values depends on REST framework setup, by default json and api are supported. The latter provides web browser interface for API. Request Headers

Accept -- the response content type depends on Accept header

Authorization -- optional token to authenticate Response Headers

Content-Type -- this depends on Accept header of request

Allow -- list of allowed HTTP methods on object Response JSON Object

detail (*string*) -- verbose description of failure (for HTTP status codes other than 200 OK)

count (*int*) -- total item count for object lists

next (*string*) -- next page URL for object lists

previous (*string*) -- previous page URL for object lists

results (array) -- results for object lists

url (string) -- URL to access this resource using API

web_url (string) -- URL to access this resource using web browser

Status Codes

200 OK -- when request was correctly handled

400 Bad Request -- when form parameters are missing

403 Forbidden -- when access is denied

429 Too Many Requests -- when throttling is in place

Authentication examples

Example request:

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

Example response:

```
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 example:

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

Passing Parameters Examples

For the POST method the parameters can be specified either as form submission (application/x-www-form-urlencoded) or as JSON (application/json).

Form request example:

```
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 request example:

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

CURL example:

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

CURL JSON example:

curl \

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

API ????

20222 4.1 222: 20222202222222 22: 22:22, 22:22

API Entry Point

GET /api/ The API root entry point. **Example request:**

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

Example response:

```
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/"
}
```

????

2????? 4.0 ????.

GET /api/users/

Returns a list of users if you have permissions to see manage users. If not, then you get to see only your own details.

Users object attributes are documented at GET /api/users/(str:username)/.

POST /api/users/ Creates a new user. Parameters

username (string) -- ??????

full_name (string) -- User full name

email (string) -- User email

is_superuser (boolean) -- Is user superuser? (optional)

is_active (boolean) -- Is user active? (optional)

GET /api/users/(str: username) / Returns information about users. Parameters

username (*string*) -- User's username Response JSON Object

username (string) -- username of a user

full_name (string) -- full name of a user

email (string) -- email of a user

is_superuser (boolean) -- whether the user is a super user

is_active (boolean) -- whether the user is active

date_joined (string) -- date the user is created

groups (array) -- link to associated groups; see GET /api/groups/(int:id)/

Example JSON data:

{

```
"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) / Changes the user parameters. Parameters

username (*string*) -- User's username Response JSON Object

username (*string*) -- username of a user

full_name (string) -- full name of a user

email (string) -- email of a user

is_superuser (boolean) -- whether the user is a super user

is_active (boolean) -- whether the user is active

date_joined (string) -- date the user is created

PATCH /api/users/(str: username)/ Changes the user parameters. Parameters

username (*string*) -- User's username Response JSON Object

username (*string*) -- username of a user

full_name (string) -- full name of a user

email (*string*) -- email of a user

is_superuser (*boolean*) -- whether the user is a super user

is_active (boolean) -- whether the user is active

date_joined (*string*) -- date the user is created

DELETE /api/users/(str: username)/ Deletes all user information and marks the user inactive. Parameters

username (*string*) -- User's username

POST /api/users/(str: username)/groups/ Associate groups with a user. Parameters

username (*string*) -- User's username Form Parameters

string group_id -- The unique group ID

GET /api/users/(str: username)/statistics/ List statistics of a user. Parameters

username (*string*) -- User's username Response JSON Object

GET /api/users/(str: username)/notifications/ List subscriptions of a user.

Parameters

username (string) -- User's username

POST /api/users/(**str:** *username*)**/notifications/** Associate subscriptions with a user. Parameters

username (*string*) -- User's username Request JSON Object

notification (*string*) -- Name of notification registered

scope (*int*) -- Scope of notification from the available choices

frequency (*int*) -- Frequency choices for notifications

GET /api/users/(str: username)/notifications/ int: subscription_id/ Get a subscription associated with a user. Parameters

username (string) -- User's username

subscription_id (int) -- ??????? ID

PUT /api/users/(str: username)/notifications/ int: subscription_id/ Edit a subscription associated with a user. Parameters

username (*string*) -- User's username

subscription_id (*int*) -- 2222222 ID Request JSON Object **notification** (*string*) -- Name of notification registered

scope (*int*) -- Scope of notification from the available choices

frequency (*int*) -- Frequency choices for notifications

PATCH /api/users/(str: username)/notifications/ int: subscription_id/ Edit a subscription associated with a user. Parameters

username (string) -- User's username

subscription_id (*int*) -- 2222222 ID Request JSON Object

notification (*string*) -- Name of notification registered

scope (*int*) -- Scope of notification from the available choices

frequency (*int*) -- Frequency choices for notifications

DELETE /api/users/(str: username)/notifications/ int: subscription_id/ Delete a subscription associated with a user. Parameters

username (string) -- User's username

subscription_id -- Name of notification registered

subscription_id -- int

????

22222 4.0 222.

GET /api/groups/

Returns a list of groups if you have permissions to see manage groups. If not, then you get to see only the groups the user is a part of.

??:

Group object attributes are documented at GET /api/groups/(int:id)/.

POST /api/groups/ Creates a new group. Parameters

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

project_selection (*int*) -- Group of project selection from given options

language_selection (int) -- Group of languages selected from given options

GET /api/groups/(int: *id*) / Returns information about group. Parameters

id (*int*) -- Group's ID Response JSON Object

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

project_selection (int) -- integer corresponding to group of projects

language_selection (*int*) -- integer corresponding to group of languages

roles (array) -- link to associated roles; see GET /api/roles/(int:id)/

projects (array) -- link to associated projects; see GET /api/projects/(string:project)/

components (array) -- link to associated components; see GET /api/components/ (string:project)/(string:component)/

Example JSON data:

```
{
    "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/",
   1,
    "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/"
   1
}
```

PUT /api/groups/(int: *id*) / Changes the group parameters. Parameters

id (*int*) -- Group's ID Response JSON Object

name (string) -- name of a group

project_selection (int) -- integer corresponding to group of projects

language_selection (*int*) -- integer corresponding to group of Languages

PATCH /api/groups/(int: *id*) / Changes the group parameters. Parameters

id (*int*) -- Group's ID Response JSON Object

name (string) -- name of a group

project_selection (*int*) -- integer corresponding to group of projects

language_selection (*int*) -- integer corresponding to group of languages

DELETE /api/groups/(int: *id*) / Deletes the group. Parameters

id (int) -- Group's ID

POST /api/groups/(int: *id*)/roles/ Associate roles with a group. Parameters

id (*int*) -- Group's ID Form Parameters

string role_id -- The unique role ID

POST /api/groups/(int: *id*)/components/ Associate components with a group. Parameters

id (*int*) -- Group's ID Form Parameters

string component_id -- The unique component ID

DELETE /api/groups/(int: *id*)/components/ int: *component_id* Delete component from a group. Parameters

id (int) -- Group's ID

component_id (int) -- The unique component ID

POST /api/groups/ (int: *id*) **/projects/** Associate projects with a group. Parameters

id (*int*) -- Group's ID Form Parameters

string project_id -- The unique project ID

DELETE /api/groups/(int: *id*)/projects/ int: *project_id* Delete project from a group. Parameters

id (int) -- Group's ID

project_id (int) -- The unique project ID

POST /api/groups/(int: *id*)/languages/ Associate languages with a group. Parameters

id (*int*) -- Group's ID Form Parameters

string language_code -- The unique language code

DELETE /api/groups/(int: *id*)/languages/ string: *language_code* Delete language from a group. Parameters

id (int) -- Group's ID

language_code (string) -- The unique language code

id (*int*) -- Group's ID Form Parameters

string component_list_id -- The unique componentlist ID

DELETE /api/groups/(int: *id*)/componentlists/ int: *component_list_id* Delete componentlist from a group. Parameters

id (int) -- Group's ID

component_list_id (int) -- The unique componentlist ID

???

```
GET /api/roles/
```

Returns a list of all roles associated with user. If user is superuser, then list of all existing roles is returned.

Roles object attributes are documented at GET /api/roles/(int:id)/.

POST /api/roles/ Creates a new role. Parameters

name (string) -- Role name

permissions (array) -- List of codenames of permissions

GET /api/roles/(int: *id*)/ Returns information about a role. Parameters

id (int) -- Role ID Response JSON Object

name (*string*) -- Role name

permissions (array) -- list of codenames of permissions

Example JSON data:

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

PUT /api/roles/(int: *id*) / Changes the role parameters. Parameters

id (*int*) -- Role's ID Response JSON Object

name (string) -- Role name

permissions (*array*) -- list of codenames of permissions

PATCH /api/roles/(int: *id*) / Changes the role parameters. Parameters

id (int) -- Role's ID Response JSON Object

name (string) -- Role name

permissions (array) -- list of codenames of permissions

DELETE /api/roles/(int: *id*)/ Deletes the role. Parameters

id(int) -- Role's ID

??

GET /api/languages/

Returns a list of all languages.

??:

Language object attributes are documented at GET /api/languages/(string:language)/.

POST /api/languages/ Creates a new language. Parameters

code (*string*) -- ????

name (string) -- ????

direction (string) -- Language direction

plural (object) -- Language plural formula and number

GET /api/languages/(string: language) / Returns information about a language. Parameters

language (string) -- ??????? Response JSON Object

code (*string*) -- ???????

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

plural (object) -- Object of language plural information

aliases (array) -- Array of aliases for language

Example JSON data:

```
{
    "code": "en",
    "direction": "ltr",
    "name": "English",
     "plural": {
"id": 75,
          "source": 0,
          "number": 2,
"formula": "n != 1",
          "type": 1
    },
"aliases": [
    ~~ish
          "english",
          "en_en",
          "base",
          "source",
          "eng"
    1,
     "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) / Changes the language parameters. Parameters

language (*string*) -- Language's code Request JSON Object

name (string) -- ????

direction (string) -- Language direction

plural (object) -- Language plural details

PATCH /api/languages/(string: language)/ Changes the language parameters. Parameters

language (*string*) -- Language's code Request JSON Object

name (string) -- ????

direction (string) -- Language direction

plural (object) -- Language plural details

DELETE /api/languages/(string: language) / Deletes the Language. Parameters

language (string) -- Language's code

GET /api/languages/(string: language)/statistics/ Returns statistics for a language. Parameters

language (string) -- ??????? Response JSON Object

total (int) -- total number of strings
total_words (int) -- total number of words
last_change (timestamp) -- last changes in the language
recent_changes (int) -- total number of changes
translated (int) -- number of translated strings
translated_percent (float) -- percentage of translated strings
translated_words (int) -- number of translated words
translated_words_percent (int) -- percentage of translated words
translated_chars (int) -- number of translated characters

??????

GET /api/projects/ Returns a list of all projects.

??:

Project object attributes are documented at GET /api/projects/(string:project)/.

POST /api/projects/ 22222 3.9 2222.

Creates a new project. Parameters

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

GET /api/projects/(**string:** *project*) / Returns information about a project. Parameters

project (string) -- 222222 URL 2222 Response JSON Object

name (*string*) -- project name

web (string) -- project website

components_list_url (string) -- URL to components list; see GET /api/projects/ (string:project)/components/

repository_url (string) -- URL to repository status; see GET /api/projects/ (string:project)/repository/

changes_list_url(string)--URL to changes list; see GET /api/projects/(string:project)/
changes/

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

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

set_language_team (boolean) -- Set Language-Team header

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

mail(string) -- ????????

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

Example JSON data:

```
"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) / 22222 4.3 2222.

Edit a project by a PATCH request.

Parameters

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

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

PUT /api/projects/(string: *project*)/ [2][2][2] 4.3 [2][2].

Edit a project by a PUT request. Parameters

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

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

222222 3.9 2222. Deletes a project. Parameters

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

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

Returns a list of project changes. This is essentially a project scoped *GET* /*api/changes*/ accepting same params. Parameters

project (string) -- 222222 URL 2022 Response JSON Object

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

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

Returns information about VCS repository status. This endpoint contains only an overall summary for all repositories for the project. To get more detailed status use *GET* /api/components/(string:project)/ (string:component)/repository/. Parameters

needs_commit (boolean) -- whether there are any pending changes to commit

needs_merge (boolean) -- whether there are any upstream changes to merge

needs_push (boolean) -- whether there are any local changes to push

Example JSON data:

{

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

POST /api/projects/(string: project)/repository/ Performs given operation on the VCS repository. Parameters

operation (string) -- Operation to perform: one of push, pull, commit, reset, cleanup
Response JSON Object

result (boolean) -- result of the operation

CURL example:

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

JSON request example:

```
POST /api/projects/hello/repository/ HTTP/1.1
Host: example.com
Accept: application/json
```

(?????????)

```
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20
```

{"operation":"pull"}

JSON response example:

```
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/ Returns a list of translation components in the given project. Parameters

project (*string*) -- 222222 URL 2222 Response JSON Object

results (array) -- array of component objects; see GET /api/components/(string:project)/
(string:component)/

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

?????? 3.9 ????.

20222 4.3 2022: The zipfile and docfile parameters are now accepted for VCS less components, see 20202 20202.

Creates translation components in the given project.

Most of the component creation happens in the background. Check the task_url attribute of created component and follow the progress there.

Parameters

project (string) -- 222222 URL 2222 Request JSON Object

zipfile (file) -- ZIP file to upload into Weblate for translations initialization

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

CURL example:

```
curl \
    --data-binary '{
        "branch": "master",
        "file_format": "po",
        "filemask": "po/*.po",
        "git_export": "",
        "license": "",
        "license_url": "",
        "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 request example:

```
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": "master",
    "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 response example:

```
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": "master",
    "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/"
     "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/ Returns paginated statistics for all languages within a project. 22222 3.8 222. Parameters project (string) -- 2222222 URL 2222 Response JSON Object **results** (*array*) -- array of translation statistics objects language (string) -- language name code (string) -- language code total (*int*) -- total number of strings translated (*int*) -- number of translated strings translated_percent (float) -- percentage of translated strings total_words (int) -- total number of words translated_words (int) -- number of translated words words_percent (float) -- percentage of translated words GET /api/projects/(string: project)/statistics/ Returns statistics for a project. 2???? 3.8 ???? Parameters project (string) -- ???????? URL ????? **Response JSON Object** total (*int*) -- total number of strings translated (*int*) -- number of translated strings **translated_percent** (*float*) -- percentage of translated strings total_words (int) -- total number of words translated words (int) -- number of translated words **words_percent** (*float*) -- percentage of translated words ????????GET /api/components/ Returns a list of translation components. ??:

Component object attributes are documented at *GET* /*api/components/(string:project)/(string:component)/.*

GET /api/components/(string: project)/ string: component/ Returns information about translation component. Parameters

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

component (string) -- 2222222 URL 2222 Response JSON Object project (object) -- the translation project; see GET /api/projects/(string:project)/

```
edit_template(string) -- ??????????
intermediate (string) -- ?????????
file_format (string) -- ??????
license (string) -- ?????????
new_lang(string) -- ?????????
language_code_style (string) -- 22222 2222
source_language (object) -- source language
                                          object;
                                                 see GET
                                                             /api/languages/
(string:language)/
push (string) -- ?????????? URL
check flags (string) -- 22222
priority (string) -- ?????
enforced_checks (string) -- ??????
restricted (string) -- Restricted access
report_source_bugs (string) -- 222222222222
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
enable_suggestions (string) -- ????????
suggestion_voting (string) -- ???????
suggestion_autoaccept (string) -- ????????
auto_lock_error (string) -- 2222222
language_regex (string) -- 2222222
variant_regex (string) -- ????????
repository_url (string) -- URL to repository status;
                                                 see GET
                                                            /api/components/
(string:project)/(string:component)/repository/
translations_url (string) -- URL to translations list; see GET
                                                            /api/components/
(string:project)/(string:component)/translations/
lock_url (string) -- URL to lock status; see GET /api/components/(string:project)/
(string:component)/lock/
changes_list_url (string) -- URL to changes list; see GET
                                                            /api/components/
(string:project)/(string:component)/changes/
task_url (string) -- URL to a background task (if any); see GET /api/tasks/(str:uuid)/
```

Example JSON data:

```
{
    "branch": "master",
    "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/"
ļ
```

PATCH /api/components/(string: project)/ string: component/ Edit a component by a PATCH request. Parameters

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

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

source_language (*string*) -- Project source language code (optional) Request JSON Object

name (string) -- name of component

repo (*string*) -- VCS repository URL

CURL example:

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

JSON request example:

```
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 response example:

```
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": "master",
    "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<sup>†</sup>,
    "web_url": "http://example.com/projects/hello/weblate/"
}
```

PUT /api/components/(string: project)/ string: component/ Edit a component by a PUT request. Parameters

component (*string*) -- 22222222 URL 22222 Request JSON Object

branch (*string*) -- VCS repository branch

file_format (string) -- file format of translations

filemask (string) -- mask of translation files in the repository

name (string) -- name of component

repo (string) -- VCS repository 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. Parameters

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. Parameters

component (string) -- 2222222 URL 2222 Response JSON Object

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

GET /api/components/(string: project)/ string: component/screenshots/ Returns a list of component screenshots. Parameters

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

component (*string*) -- 202022222 URL 20202 Response JSON Object

results (array) -- array of component screenshots; see GET /api/screenshots/(int:id)/

GET /api/components/(string: project)/ string: component/lock/ Returns component lock status. Parameters

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

component (*string*) -- 2222222 URL 22222 Response JSON Object

locked (boolean) -- whether component is locked for updates

Example JSON data:

{

}

"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/. Parameters

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

component (*string*) -- 22222222 URL 2222 Request JSON Object

lock -- Boolean whether to lock or not.

CURL example:

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

JSON request example:

```
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 response example:

```
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/. Parameters

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

needs_commit (*boolean*) -- whether there are any pending changes to commit

needs_merge (*boolean*) -- whether there are any upstream changes to merge

needs_push (*boolean*) -- whether there are any local changes to push

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. Parameters

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

component (*string*) -- 22222222 URL 22222 Request JSON Object

operation (string) -- Operation to perform: one of push, pull, commit, reset, cleanup
Response JSON Object

result (boolean) -- result of the operation

CURL example:

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

JSON request example:

```
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 response example:

```
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. Parameters component (string) -- ????????? URL ????? GET /api/components/(string: project)/ **string:** *component*/**new_template**/ Downloads template file for new translations. Parameters component (string) -- ????????? URL ????? **GET** /api/components/(string: *project*) / string: *component*/translations/ Returns a list of translation objects in the given component. Parameters **Response JSON Object 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. Parameters project (string) -- ????????? URL ????? Request JSON Object language_code (string) -- translation language code; see GET /api/languages/ (string:language)/ **Response JSON Object result** (*object*) -- new translation object created

CURL example:

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

JSON request example:

```
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 response example:

```
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][2][2] 2.7 [2][2]. Parameters

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

component (*string*) -- 22222222 URL 2222 Response JSON Object

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

/api/translations/(string:project)/

??

GET /api/translations/ Returns a list of translations.

Translation object attributes are documented at GET (string:component)/(string:language)/.

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

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

language (*string*) -- Translation language code Response JSON Object

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

filename (*string*) -- translation filename

fuzzy_percent (*float*) -- percentage of fuzzy (marked for edit) strings

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

language (object) -- source language object; see 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 number of strings

total_words (int) -- total number of words

translate_url (string) -- URL for translating

translated (*int*) -- number of translated strings

translated_percent (float) -- percentage of translated strings

translated_words (int) -- number of translated words

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/

Example JSON data:

{
"component": {
"branch": "master",
"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/"
},
<pre>"repo": "file:///home/nijel/work/weblate-hello", "slug": "weblate",</pre>
"template": "",
cemprate . ,

(2222222)

(???????????)

```
"url": "http://example.com/api/components/hello/weblate/",
        "vcs": "git",
        "web_url": "http://example.com/projects/hello/weblate/"
    "failing_checks": 3,
    "failing_checks_percent": 75.0,
    "failing_checks_words": 11,
    "filename": "po/cs.po",
    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "fuzzy_words": 0,
    "have_comment": 0,
    "have_suggestion": 0,
    "is_template": false,
    "language": {
        "code": "cs",
"direction": "ltr",
        "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/ 22222 3.9 222.

Deletes a translation. Parameters

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

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. Parameters

language (*string*) -- Translation language code Response JSON Object

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

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

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

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

language (string) -- Translation language code

q (*string*) -- Search query string 22 (optional) Response JSON Object

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. Parameters

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

language (*string*) -- Translation language code Request JSON Object

key (string) -- Name of translation unit

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

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

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

language (*string*) -- Translation language code Request JSON Object

filter_type (string) -- Automatic translation filter type

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

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

string: *component/string*: *language/file/* Download current translation file as stored in VCS (without format parameter) or as converted to a standard format (currently supported: Gettext PO, MO, XLIFF and TBX).

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.

Query Parameters

format -- File format to use; if not specified no format conversion happens; supported file formats: po, mo, xliff, xliff11, tbx, csv, xlsx, json, aresource, strings Parameters

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

language (string) -- Translation language code

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

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

language (*string*) -- Translation language code Form Parameters

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

file file -- Uploaded file

string author -- ?????

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

CURL example:

```
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/.

Parameters

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

language (*string*) -- Translation language code

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

string: component/string: language/repository/ Performs given operation on the VCS repository.
See POST /api/projects/(string:project)/repository/ for documentation.
Parameters

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

language (*string*) -- Translation language code Request JSON Object

operation (*string*) -- **Operation** to perform: one of push, pull, commit, reset, cleanup Response JSON Object

result (boolean) -- result of the operation

GET /api/translations/(string: project)/ **string:** component/**string:** language/**statistics**/ Returns detailed translation statistics. [2][2][2] 2.7 [2][2]. Parameters

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

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

language (*string*) -- Translation language code Response JSON Object

code (*string*) -- language code

failing (*int*) -- number of failing checks

failing_percent (*float*) -- percentage of failing checks

fuzzy_percent (float) -- percentage of fuzzy (marked for edit) strings

total_words (int) -- total number of words

translated_words (int) -- number of translated words

last_author (string) -- name of last author

last_change (timestamp) -- date of last change

name (*string*) -- language name

total (*int*) -- total number of strings

translated (*int*) -- number of translated strings

translated_percent (float) -- percentage of translated strings

url (*string*) -- URL to access the translation (engagement URL)

url_translate (*string*) -- URL to access the translation (real translation URL)

Units

?????? 2.10 ????. GET /api/units/ Returns list of translation units. ??: Unit object attributes are documented at GET /api/units/(int:id)/. GET /api/units/(int: id)/ 2222 4.3 222: The target and source are now arrays to properly handle plural strings. Returns information about translation unit. Parameters id (int) -- Unit ID Response JSON Object translation (string) -- URL of a related translation object **source** (*array*) -- source string 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 **position** (*int*) -- unit position in translation file 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 web_url (string) -- URL where the unit can be edited souce_unit (string) -- Source unit link; see GET /api/units/(int:id)/ PATCH /api/units/(int: id)/ ????? 4.3 ????. Parameters id (int) -- Unit ID Request JSON Object 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

PUT /api/units/(int: *id*) / 22222 4.3 2222.

2222222222222222 Parameters

id (*int*) -- Unit ID Request JSON Object

state (*int*) -- unit state, 0 - not translated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see 22222)

target (array) -- target string

explanation (string) -- String explanation, available on source units, see Additional info on source strings

DELETE /api/units/(int: *id*)/ (2020224.3 (2022) 202020202020202 Parameters **id**(*int*) -- Unit ID

??

?????? 2.10 ????.

GET /api/changes/ 20222 4.1 222: 20222222222 4.1 222222222222

Returns a list of translation changes.

??:

Change object attributes are documented at GET /api/changes/(int:id)/. Query Parameters

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. Parameters

id (*int*) -- Change ID Response JSON Object

unit (string) -- URL of a related unit object translation (string) -- URL of a related translation object component (string) -- URL of a related component object glossary_term (string) -- URL of a related glossary term 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. Parameters

id (*int*) -- Screenshot ID Response JSON Object

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. Parameters

id (int) -- Screenshot ID

POST /api/screenshots/(int: *id*)**/file/** Replace screenshot image. Parameters

id (*int*) -- Screenshot ID Form Parameters

file image -- Uploaded file

CURL example:

curl -X POST \
 -F image=@image.png \
 -H "Authorization: Token TOKEN" \
 http://example.com/api/screenshots/1/file/

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

Associate source string with screenshot. Parameters

id (int) -- Screenshot ID Form Parameters

string unit_id -- Unit ID Response JSON Object

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

translation (string) -- URL of a related translation 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)/

id (int) -- Screenshot ID

unit_id -- ??????? ID

POST /api/screenshots/ Creates a new screenshot. Form Parameters

file image -- Uploaded file

string language_code -- ??????? Response JSON Object

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)/

id (*int*) -- Screenshot ID Response JSON Object

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)/

id (*int*) -- Screenshot ID Response JSON Object

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*)/ 22222222222222 Parameters

id (int) -- Screenshot ID

????

?????? 4.4.1 ????.

GET /api/addons/ Returns a list of addons.

??:

Addon object attributes are documented at GET /api/addons/(int:id)/.

GET /api/addons/(int: *id*)/ Returns information about addon information. Parameters

id (*int*) -- Addon ID Response JSON Object

name (*string*) -- name of an addon

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

configuration (*object*) -- Optional addon configuration

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

string: component/addons/ Creates a new addon. Parameters

name (*string*) -- name of an addon

configuration (object) -- Optional addon configuration

PATCH /api/addons/(int: *id*)/ Edit partial information about addon. Parameters

id (*int*) -- Addon ID Response JSON Object

configuration (*object*) -- Optional addon configuration

PUT /api/addons/(int: *id*) / Edit full information about addon. Parameters

id (*int*) -- Addon ID Response JSON Object

configuration (*object*) -- Optional addon configuration

DELETE /api/addons/(int: *id*)/ Delete addon. Parameters

id (int) -- Addon ID

22???? 4.0 ????.

GET /api/component-lists/ 2222222 2222222222222222

??:

GET /api/component-lists/(str: *slug*)/ Returns information about component list. Parameters

slug (*string*) -- 2222222 222222 Response JSON Object

name (string) -- name of a component list

show_dashboard (*boolean*) -- whether to show it on a dashboard

components (array) -- link to associated components; see GET ,
(string:project)/(string:component)/

/api/components/

auto_assign (array) -- automatic assignment rules

PUT /api/component-lists/(str: *slug*) / Changes the component list parameters. Parameters

name (string) -- name of a component list

show_dashboard (boolean) -- whether to show it on a dashboard

PATCH /api/component-lists/(str: *slug*)/ Changes the component list parameters. Parameters

slug (*string*) -- 22222222 2222222 Request JSON Object

name (string) -- name of a component list

show_dashboard (boolean) -- whether to show it on a dashboard

DELETE /api/component-lists/(str: *slug*)/ Deletes the component list. Parameters

POST /api/component-lists/(str: *slug*)/components/ Associate component with a component list. Parameters

string component_id -- Component ID

DELETE /api/component-lists/(str: *slug*)/components/ str: *component_slug* Disassociate a component from the component list. Parameters

???

GET /api/glossary/ Returns a list of all glossaries which are associated with a project that user has access to.

Language object attributes are documented at GET /api/languages/(string:language)/.

id (int) -- 22 id Response JSON Object

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

color (*string*) -- ?????????

source_language (object) -- Object of language plural information

projects (array) -- link to associated projects; see GET /api/projects/(string:project)/

Example JSON data:

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

(?????????)

```
"slug": "hello",
         "id": 1,
         "source_language": {
             "code": "en",
"name": "English",
             "plural": {
"id": 75,
                  "source": 0,
                  "number": 2,
"formula": "n != 1",
                  "type": 1
             },
             "aliases": [
                  "english",
                  "en_en",
                  "base",
                  "source",
                  "eng"
             1,
             "direction": "ltr",
             "web_url": "http://example.com/languages/en/",
"url": "http://example.com/api/languages/en/",
             "statistics_url": "http://example.com/api/languages/en/
⇔statistics/"
         "url": "http://example.com/api/projects/demo1/",
"components_list_url": "http://example.com/api/projects/demo1/
⇔components/
         "repository url": "http://example.com/api/projects/demo1/
∽repository/"
         "statistics_url": "http://example.com/api/projects/demo1/
↔statistics/",
         "changes_list_url": "http://example.com/api/projects/demo1/changes/
\rightarrow",
         "languages_url": "http://example.com/api/projects/demo1/languages/"
    },
    "projects_url": "http://example.com/api/glossary/7/projects/",
    "terms_url": "http://example.com/api/glossary/7/terms/",
    "url": "http://example.com/api/glossary/7/"
}
```

PUT /api/glossary/(int: *id*)/ 2022/2020/2022/2020/2 Parameters

id (int) -- ????? id Request JSON Object

name (string) -- ????

color (*string*) -- Language direction

source_language (object) -- Language plural details

id (*int*) -- ????? id Request JSON Object

name (string) -- ????

color (*string*) -- Language direction

source_language (object) -- Language plural details

DELETE /api/glossary/(int: *id*) / Parameters

id (int) -- ???? id

GET /api/glossary/(int: *id*)/projects/ Returns projects linked with a glossary. Parameters

id (*int*) -- ???? id Response JSON Object

POST /api/glossary/(int: *id*)/projects/ 22222222222222222 Parameters

id (*int*) -- ???? id Form Parameters

DELETE /api/glossary/(int: *id*)/projects/ 2020202020202020202020 Parameters

id (int) -- ???? id Form Parameters

GET /api/glossary/(int: *id*)/terms/ 2022/2022/2022/2022 Parameters

id (int) -- ???? id

POST /api/glossary/(int: *id*)/terms/ 2022/202022222 Parameters

id (*int*) -- ???? id

Request JSON Object

language (*object*) -- ???????

source (string) -- ??????

GET /api/glossary/(int: *id*)/terms/ int: *term_id*/ 2022/2022/2022/2022/2022 Parameters

id (int) -- ???? id

term_id (int) -- ID of term

id (int) -- ???? id

term_id (*int*) -- ID of term Request JSON Object

language (object) -- ???????

source (*string*) -- ??????

id (int) -- ???? id

term_id (*int*) -- ID of term Request JSON Object

language (*object*) -- ??????

source (string) -- ??????

DELETE /api/glossary/(int: *id*)/terms/ int: *term_id*/ Delete a term associated with a glossary. Parameters

id (int) -- ???? id

term_id (*int*) -- ID of term

Tasks

22222 4.4 222.

uuid (*string*) -- Task UUID Response JSON Object

progress (int) -- Task progress in percent

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/ 2222 2.6 2222: 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][2][2][2][2.6[2][2][2]: 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/free-pro-team@latest/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 2222222222 Pagure 22222222222 https://docs.pagure.org/pagure/usage/using_webhooks.html Pagure WEB 222222222222 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/ 22222 3.9 2222.

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/ [2][2][2] 3.9 [2][2].

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/ Query Parameters

format (string) -- Output format: either json or csv

22222.2.6 2222: 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/master/ 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/
←master/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/
∽master/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/
>master/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.

??:

RSS Wikipedia ???

Weblate ?????

??

pip3 install wlc

Docker ???

??????:

docker pull weblate/wlc

docker run --rm weblate/wlc [WLC_ARGS]

??

You might want to pass your 22222222 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

????

[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]

??

??

--format {csv, json, text, html} 22222222222

--url URL

????

version

list-projects

list-components

list-translations

show

ls

commit

pull

reset

?????? 0.7 ????: wlc 0.7 ??????

cleanup

?????? 0.9 ????: wlc 0.9 ??????

repo

statistics

lock-status

22222 0.5 222: wlc 0.5 22222

lock ?????? 0.5 ????: wlc 0.5 ??????

unlock

changes

2222 0.7 222: wlc 0.7 222 Weblate 2.10 222222

download

?????? 0.7 ????: wlc 0.7 ??????

-convert

--output

upload

?????? 0.9 ????: wlc 0.9 ??????

--input

???????

 22222
 1.6
 2222
 22222

 22222
 22222
 2222

 22222
 2222
 2222

 22222
 2222
 2222

 22222
 2222
 2222

 22222
 2222
 2222

key Weblate ?????????? API ????

url

API ?????? URL??????? http://127.0.0.1:8000/api/?

[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/master

2222API 22222 [keys] 22222222222:

[keys]

https://hosted.weblate.org/api/ = APIKEY

?

\$ 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/
```

```
$ cat .weblate
[weblate]
url = https://hosted.weblate.org/api/
translation = weblate/master
$ wlc show
branch: master
file_format: po
source_language: en
filemask: weblate/locale/*/LC_MESSAGES/django.po
git_export: https://hosted.weblate.org/git/weblate/master/
license: GPL-3.0+
license_url: https://spdx.org/licenses/GPL-3.0+
name: master
new_base: weblate/locale/django.pot
project: weblate
repo: git://github.com/WeblateOrg/weblate.git
slug: master
template:
url: https://hosted.weblate.org/api/components/weblate/master/
vcs: git
web_url: https://hosted.weblate.org/projects/weblate/master/
```

\$ wlc commit

Weblate ? Python API

??

pip install wlc

wlc

WeblateException

exception wlc.WeblateException 222222222222

Weblate

API 222222222API 22222222 API URL 2222222

get (*path*) 22222 **path** (*str*) -- 222222 22222 object

post (*path*, ***kwargs*) 22222 path (*str*) -- 22222222 ?????? object

wlc.config

WeblateConfig

class wlc.config.WeblateConfig(section='wlc') ????? load (path=None) 22222202222222222222222222222222222XDG 22222/etc/xdg/wlc 2222222 wlc 2222222~/.

wlc.main

wlc.main.main(settings=None, stdout=None, args=None) ??????2222 222 22222222222 2222 22222

Command

class wlc.main.**Command** (args, config, stdout=None)

????

Weblate ???????

Installing using 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 all 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
```

[1]: 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][2][2] 2.15-2 [2][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.

2222 3.7.1-6 222: 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 22 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:

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.

22: 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

Docker environment variables

Many of Weblate's 🖉 can be set in the Docker container using environment variables:

Generic settings

WEBLATE_DEBUG Configures Django debug mode using *DEBUG*.

??

environment: WEBLATE_DEBUG: 1

??:

Disable debug mode.

WEBLATE_LOGLEVEL Configures the logging verbosity.

WEBLATE_SITE_TITLE Changes the site-title shown in the header of all pages.

222: In case it is not set, the first item from WEBLATE_ALLOWED_HOSTS is used.

??:

Set correct site domain?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
```

??:

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*, *WE-BLATE_ADMIN_NAME* and *WEBLATE_ADMIN_EMAIL*.

12: 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.

??:

222222 22 WEBLATE_ADMIN_PASSWORD WEBLATE_ADMIN_NAME WEBLATE_ADMIN_EMAIL

WEBLATE_SERVER_EMAIL

WEBLATE_DEFAULT_FROM_EMAIL

Configures the address for outgoing e-mails.

??:

Configure e-mail sending

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 ?? Allowed hosts setup?? Set correct site domain

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 *REGISTRA-TION_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.

22: 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 Set correct site domain 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 LO-GIN_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

WEBLATE_PAGURE_TOKEN

??:

Pagure

WEBLATE_SIMPLIFY_LANGUAGES

Configures the language simplification policy, see SIMPLIFY_LANGUAGES.

WEBLATE_DEFAULT_ACCESS_CONTROL

Configures the default <u>22222</u> for new projects, see *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Configures the default value for *Restricted access* for new components, see *DE*-*FAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Configures the default value for <u>222222222</u> for new components, see *DE*-*FAULT_TRANSLATION_PROPAGATION*.

WEBLATE DEFAULT COMMITER EMAIL

Configures DEFAULT_COMMITER_EMAIL.

WEBLATE_DEFAULT_COMMITER_NAME

Configures DEFAULT_COMMITER_NAME.

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.

??:

Content security policy CSP_SCRIPT_SRC CSP_IMG_SRC CSP_CONNECT_SRC CSP_STYLE_SRC CSP_FONT_SRC

WEBLATE_LICENSE_FILTER LICENSE_FILTER ?????

WEBLATE_HIDE_VERSION HIDE_VERSION ????

WEBLATE_BASIC_LANGUAGES BASIC_LANGUAGES ?????

Machine translation settings

WEBLATE_MT_APERTIUM_APY

Enables Apertium machine translation and sets MT_APERTIUM_APY

WEBLATE_MT_AWS_REGION

WEBLATE_MT_AWS_ACCESS_KEY_ID

WEBLATE_MT_AWS_SECRET_ACCESS_KEY Configures *AWS* machine translation.

environment:

WEBLATE_MT_AWS_REGION: us-east-1 WEBLATE_MT_AWS_ACCESS_KEY_ID: AKIAIOSFODNN7EXAMPLE WEBLATE_MT_AWS_SECRET_ACCESS_KEY: wJalrXUtnFEMI/K7MDENG/ ↔ bPxRfiCYEXAMPLEKEY

WEBLATE_MT_DEEPL_KEY

Enables *DeepL* machine translation and sets *MT_DEEPL_KEY*

WEBLATE_MT_DEEPL_API_VERSION

2222 DeepL API **222 222 222 222** *Mt_deepl_api_version*?

WEBLATE_MT_GOOGLE_KEY Enables *Google Translate* and sets *MT_GOOGLE_KEY*

WEBLATE_MT_MICROSOFT_COGNITIVE_KEY

Enables Microsoft Cognitive Services Translator and sets MT_MICROSOFT_COGNITIVE_KEY

WEBLATE_MT_MICROSOFT_ENDPOINT_URL

Sets MT_MICROSOFT_ENDPOINT_URL, please note this is supposed to contain domain name only.

WEBLATE_MT_MICROSOFT_REGION Sets MT_MICROSOFT_REGION

WEBLATE_MT_MICROSOFT_BASE_URL Sets *MT_MICROSOFT_BASE_URL*

WEBLATE_MT_MODERNMT_KEY Enables *ModernMT* and sets *MT_MODERNMT_KEY*.

WEBLATE_MT_MYMEMORY_ENABLED

Enables *MyMemory* machine translation and sets *MT_MYMEMORY_EMAIL* to *WEBLATE_ADMIN_EMAIL*. ??

```
environment:
 WEBLATE_MT_MYMEMORY_ENABLED: 1
```

```
WEBLATE_MT_GLOSBE_ENABLED
```

Enables *Glosbe* machine translation.

```
environment.
 WEBLATE MT GLOSBE ENABLED: 1
```

WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED Enables Microsoft Terminology Service machine translation.

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,
\cdc=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 authentication

GitHub

WEBLATE_SOCIAL_AUTH_GITHUB_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_SECRET Enables *GitHub authentication*.

Bitbucket

WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET Enables *Bitbucket authentication*.

Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY

WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET Enables *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 Enables Google OAuth 2.

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY WEBLATE_SOCIAL_AUTH_GITLAB_SECRET WEBLATE_SOCIAL_AUTH_GITLAB_API_URL Enables 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 authentication*.

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.

??:

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* 2222222 Webrate 222.

Database backup settings

??:

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.

??:

Enable caching

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.

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

REDIS_PASSWORD

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
```

??:

22222222

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 22222222222

??:

EMAIL_HOST_USER

WEBLATE_EMAIL_HOST_PASSWORD 2222222222

??:

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.

??:

Configure e-mail sending []EMAIL_BACKEND

Error reporting

It is recommended to collect errors from the installation systematically, see *Collecting error reports*.

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

???????

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 matches the number of processors (except the backup worker, which is supposed to run only once).

??

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

??:

Celery worker options? Background tasks using Celery

UWSGI_WORKERS

Configure how many uWSGI workers should be executed.

It defaults to number of processors + 1.

??

```
environment:
UWSGI_WORKERS: 32
```

In case you have a lot of CPU cores and hit out of memory issues, try reducing number of workers:

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

Docker container volumes

There is single data volume 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.

??:

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 after all environment settings are loaded, so it gets completely set up, and can be used to customize anything.

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 all 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.

22: 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 222222):

```
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 202020202020):

apt install tesseract-ocr libtesseract-dev libleptonica-dev

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 [22][22][22][2][2][2]]:

pip install ruamel.yaml aeidon boto3 zeep chardet tesserocr

Configuring Weblate

[2]: 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.

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 *Running server* and *Serving static files*):

weblate collectstatic

6.Compress JavaScript and CSS files (optional, see Compressing client assets):

weblate compress

7.Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See *Background tasks using Celery* for more info:

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

8.Start the development server (see *Running server* 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 *Production setup*.

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 SUSE and openSUSE

Hardware requirements

Weblate should run on all 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 2222222):

Install wanted optional dependencies depending on features you intend to use (see 202020202022):

zypper install tesseract-ocr tesseract-devel leptonica-devel

Optionally install software for running production server, see *Running server*, *Weblate* 22222222, *Background tasks using 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 **2222222222**):

pip install ruamel.yaml aeidon boto3 zeep chardet tesserocr

Configuring Weblate

[2]: 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.

3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Weblate* 2222222222 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 *Running server* and *Serving static files*):

weblate collectstatic

6.Compress JavaScript and CSS files (optional, see Compressing client assets):

weblate compress

7.Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See *Background tasks using Celery* for more info:

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

8.Start the development server (see *Running server* 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 *Production setup*.

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 all 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 2222222):

Install wanted optional dependencies depending on features you intend to use (see 22222222222):

dnf install tesseract-langpack-eng tesseract-devel leptonica-devel

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
# 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 [20202020202020]:

pip install ruamel.yaml aeidon boto3 zeep chardet tesserocr

Configuring Weblate

[1]: 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.
- 3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Weblate* 222222222 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 *Running server* and *Serving static files*):

weblate collectstatic

6. Compress JavaScript and CSS files (optional, see Compressing client assets):

weblate compress

7.Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See *Background tasks using Celery* for more info:

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

8.Start the development server (see *Running server* 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 *Production setup*.

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 macOS

Hardware requirements

Weblate should run on all 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 2222222):

```
brew install pango libjpeg python git libyaml gobject-introspection 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 2222222222):

brew install tesseract

Optionally install software for running production server, see *Running server*, *Weblate* [2022]22[22]22], *Background tasks using 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
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

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
```

pip install ruamel.yaml aeidon boto3 zeep chardet tesserocr

Configuring Weblate

[2]: 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.
- 3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Weblate* 222222222 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 Running server and Serving static files):

weblate collectstatic

6. Compress JavaScript and CSS files (optional, see Compressing client assets):

weblate compress

7.Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See *Background tasks using Celery* for more info:

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

8.Start the development server (see *Running server* 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 *Production setup*.

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 Debian and Ubuntu

Installing on SUSE and openSUSE

Installing on RedHat, Fedora and CentOS

2.Git 2222 Weblate 22222222222 tarball 222222222222222

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

. ~/weblate-env/bin/activate
pip install -e weblate-src

4.weblate/settings_example.py 2 weblate/settings.py 2022222

```
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 $\circ\circ$ 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:

The template is now available for selection using the web console or the CLI.

Parameters

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:

????????

You can also use the CLI to process templates and use the configuration that is generated to create objects immediately.

The Weblate instance should be available after successful migration and deployment at the specified WE-BLATE_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.

????

??

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

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
```

Installing using Docker 2222222222222 Virtualenv 22222222222222222 Installing on Debian and Ubuntu Installing on SUSE and openSUSE Installing on RedHat, Fedora and CentOS Installing on macOS 22222222222222222222222 Installing on OpenShift

Installing on Kubernetes

?????????

?????????

Python ?????

???????????

https://docs.python-zeep.org/ https://pypi.org/project/aeidon/

Weblate	<pre>???PostgreSQL??MySQL</pre>	???	MariaDB	2222222222222	Weblate	[2][2][2][2][2][2][2][2][2][2][2][2][2][
????????	2??????????????????????????????????????					

???????????

https://git-scm.com/ https://cairographics.org/2https://pango.gnome.org/222 Pango 2 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

??????????

Pango ? Cairo

?????? 3.7 ????.

??????????

63CB 1DF1 EF12 CF2A COEE 5A32 9C27 B313 42B7 511D

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg: using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Can't check signature: public key not found
```

\$ gpg --import wmxth3chu9jfxdxywj1skpmhsj311mzm

```
$ gpg --keyserver hkp://pgp.mit.edu --recv-keys_

→87E673AF83F6C3A0C344C8C3F4AA229D4D58C245

gpg: key 9C27B31342B7511D: "Michal Čihař <michal@cihar.com>" imported

gpg: Total number processed: 1

gpg: unchanged: 1
```

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
                    using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
abd:
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
                     aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:
                     aka "[jpeg image of size 8848]" [ultimate]
abd:
                     aka "Michal Čihař (Braiins) <michal.cihar@braiins.cz>
gpg:
\hookrightarrow" [ultimate]
gpg: WARNING: This key is not certified with a trusted signature!
              There is no indication that the signature belongs to the.
gpg:
⊖owner.
Primary key fingerprint: 63CB 1DF1 EF12 CF2A COEE 5A32 9C27 B313 42B7 511D
```

keyring

WEB

\$ 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 (Braiins) <michal.cihar@braiins.cz> →" [ultimate]

\$ 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]

??:

Serving static files

??:

Use a powerful database engine? Databases? Migrating from other databases to PostgreSQL

PostgreSQL

??:

PostgreSQL notes

PostgreSQL ???????????

```
# 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 -0 weblate weblate
```

CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;

PostgreSQL ??????? Webrate ???

settings.py PostgreSQL ?????????

```
DATABASES = \{
    "default": {
         # Database engine
        "ENGINE": "django.db.backends.postgresql",
         # Database name
        "NAME": "weblate",
         # Database user
        "USER": "weblate",
         # Name of role to alter to set parameters in PostgreSQL,
         # use in case role name is different than user used for.
→authentication.
        # "ALTER_ROLE": "weblate",
         # Database password
        "PASSWORD": "password",
         # Set to empty string for localhost
        "HOST": "database.example.com",
# Set to empty string for default
        "PORT": "",
    }
}
```

MySQL ? MariaDB

Weblate ????????:

SQL ????? STRICT_TRANS_TABLES ?????????

Below is an example /etc/my.cnf.d/server.cnf for a server with 8 GB of RAM. These settings should be sufficient for most installs. MySQL and MariaDB have tunables that will increase the performance of your server that are considered not necessary unless you are planning on having large numbers of concurrent users accessing the system. See the various vendors documentation on those details.

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

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": {},
    }
}
```

You should also create the weblate user account in MySQL or MariaDB before you begin the install. Use the commands below to achieve that:

GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password'; FLUSH PRIVILEGES;

??????

?????????

??:

Not receiving e-mails from Weblate? Configuring outgoing e-mail in Docker container

????????

?????HTTP IP

??:

HTTP ????

import os

```
os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

??:

Proxy Environment Variables

?????

??:

Copy weblate/settings_example.py to weblate/settings.py and adjust it to match your setup. You will probably want to adjust the following options: ADMINS

List of site administrators to receive notifications when something goes wrong, for example notifications on failed merges, or Django errors.

??:

ADMINS

ALLOWED HOSTS

You need to set this to list the hosts your site is supposed to serve. For example:

ALLOWED_HOSTS = ["demo.weblate.org"]

Alternatively you can include wildcard:

```
ALLOWED_HOSTS = ["*"]
```

??:

ALLOWED HOSTS ?? WEBLATE ALLOWED HOSTS ?? Allowed hosts setup

SESSION_ENGINE

Configure how your sessions will be stored. In case you keep the default database backend engine, you should schedule: weblate clearsessions to remove stale session data from the database.

If you are using Redis as cache (see *Enable caching*) it is recommended to use it for sessions as well:

SESSION_ENGINE = "django.contrib.sessions.backends.cache"

??:

Configuring the session engine? SESSION_ENGINE

DATABASES

Connectivity to database server, please check Django's documentation for more details. ??:

Weblate ?????????????DATABASES?Databases

DEBUG

Disable this for any production server. With debug mode enabled, Django will show backtraces in case of error to users, when you disable it, errors will be sent per e-mail to ADMINS (see above).

Debug mode also slows down Weblate, as Django stores much more info internally in this case.

??:

DEBUG

DEFAULT_FROM_EMAIL

E-mail sender address for outgoing e-mail, for example registration e-mails.

??:

DEFAULT_FROM_EMAIL

SECRET_KEY

Key used by Django to sign some info in cookies, see Django secret key for more info.

??:

SECRET_KEY

SERVER_EMAIL

E-mail used as sender address for sending e-mails to the administrator, for example notifications on failed merges.

??:

SERVER_EMAIL

Filling up the database

After your configuration is ready, you can run weblate migrate to create the database structure. Now you should be able to create translation projects using the admin interface.

In case you want to run an installation non interactively, you can use weblate migrate --noinput, and then create an admin user using *createadmin* command.

Once you are done, you should also check the *Performance report* in the admin interface, which will give you hints of potential non optimal configuration on your site.

??:

??**?**???????

Production setup

For a production setup you should carry out adjustments described in the following sections. The most critical settings will trigger a warning, which is indicated by an exclamation mark in the top bar if signed in as a superuser:

🗶 Weblate	Dashboard	Projects 🗸	Languages -	Checks 🗸				عر	▲	+ Add -	•	•••
② Dashboard												
Watched translat	tions 0	Suggested tra	anslations 0	Insights -	Search							
Choose what languages you want in the preferences, to see overview of available translations for those languages in your watched projects.												
		Power	ed by Weblate 4.4	About Weblate	Legal Contact	Documentation	Donate to Webla	te				

It is also recommended to inspect checks triggered by Django (though you might not need to fix all of them):

weblate check --deploy

You can also review the very same checklist from the *Management interface*.

Deployment checklist

Disable debug mode

Disable Django's debug mode (DEBUG) by:

DEBUG = False

With debug mode on, Django stores all executed queries and shows users backtraces of errors, which is not desired in a production setup.

??:

22222

Properly configure admins

Set the correct admin addresses to the *ADMINS* setting to defining who will receive e-mails in case something goes wrong on the server, for example:

ADMINS = (("Your Name", "your_email@example.com"),)

??:

22222

Set correct site domain

Adjust site name and domain in the admin interface, otherwise links in RSS or registration e-mails will not work. This is configured using *SITE_DOMAIN* which should contain site domain name.

[2][2][2] 4.2 [2][2]: Prior to the 4.2 release the Django sites framework was used instead, please see The "sites" framework.

??:

Allowed hosts setup? Correctly configure HTTPS? SITE_DOMAIN? WEBLATE_SITE_DOMAIN? ENABLE_HTTPS

Correctly configure HTTPS

It is strongly recommended to run Weblate using the encrypted HTTPS protocol. After enabling it, you should set *ENABLE_HTTPS* in the settings:

ENABLE_HTTPS = **True**

???: You might want to set up HSTS as well, see SSL/HTTPS for more details.

??:

ENABLE_HTTPS [] Allowed hosts setup [] Set correct site domain

Set properly SECURE_HSTS_SECONDS

If your site is served over SSL, you have to consider setting a value for SECURE_HSTS_SECONDS in the settings.py to enable HTTP Strict Transport Security. By default it's set to 0 as shown below.

 $SECURE_HSTS_SECONDS = 0$

If set to a non-zero integer value, the django.middleware.security.SecurityMiddleware sets the HTTP Strict Transport Security header on all responses that do not already have it.

??: Setting this incorrectly can irreversibly (for some time) break your site. Read the HTTP Strict Transport Security documentation first.

Use a powerful database engine

Enable caching

If possible, use Redis from Django by adjusting the CACHES configuration variable, for example:

```
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",
        },
    }
}
```

The set of the set of

??:

In addition to caching of Django, Weblate performs caching of avatars. It is recommended to use a separate, filebacked cache for this purpose:

```
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 ? Enable caching ? Django's cache framework

Configure e-mail sending

Weblate needs to send out e-mails on several occasions, and these e-mails should have a correct sender address, please configure *SERVER_EMAIL* and *DEFAULT_FROM_EMAIL* to match your environment, for example:

SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"

To disable sending e-mails by Weblate set EMAIL_BACKEND to django.core.mail.backends. dummy.EmailBackend.

This will disable all e-mail delivery including registration or password reset e-mails.

??:

Allowed hosts setup

Django requires *ALLOWED_HOSTS* to hold a list of domain names your site is allowed to serve, leaving it empty will block any requests.

In case this is not configured to match your HTTP server, you will get errors like Invalid HTTP_HOST header: '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS.

CONDOCKER CONTAINER, this is available as *WEBLATE_ALLOWED_HOSTS.*

??:

ALLOWED_HOSTS ?? WEBLATE_ALLOWED_HOSTS ?? Set correct site domain

Django secret key

The *SECRET_KEY* setting is used by Django to sign cookies, and you should really generate your own value rather than using the one from the example setup.

You can generate a new key using weblate/examples/generate-secret-key shipped with Weblate.

SECRET_KEY

Home directory

[2][2][2] 2.1 [2][2]: This is no longer required, Weblate now stores all its data in DATA_DIR.

The home directory for the user running Weblate should exist and be writable by this user. This is especially needed if you want to use SSH to access private repositories, but Git might need to access this directory as well (depending on the Git version you use).

You can change the directory used by Weblate in settings.py, for example to set it to configuration directory under the Weblate tree:

os.environ["HOME"] = os.path.join(BASE_DIR, "configuration")

22: On Linux, and other UNIX like systems, the path to user's home directory is defined in /etc/passwd. Many distributions default to a non-writable directory for users used for serving web content (such as apache, www-data or wwwrun), so you either have to run Weblate under a different user, or change this setting.

??:

222222222

Template loading

It is recommended to use a cached template loader for Django. It caches parsed templates and avoids the need to do parsing with every single request. You can configure it using the following snippet (the loaders setting is important here):



??:

django.template.loaders.cached.Loader

Running maintenance tasks

For optimal performance, it is good idea to run some maintenance tasks in the background. This is now automatically done by *Background tasks using Celery* and covers following tasks:

Configuration health check (hourly).

Committing pending changes (hourly), see *Lazy commits* and *commit_pending*.

Updating component alerts (daily).

Update remote branches (nightly), see AUTO_UPDATE.

Translation memory backup to JSON (daily), see *dump_memory*.

Fulltext and database maintenance tasks (daily and weekly tasks), see *cleanuptrans*.

2222 3.2 222: Since version 3.2, the default way of executing these tasks is using Celery and Weblate already comes with proper configuration, see *Background tasks using Celery*.

System locales and encoding

The system locales should be configured to UTF-8 capable ones. On most Linux distributions this is the default setting. In case it is not the case on your system, please change locales to UTF-8 variant.

For example by editing /etc/default/locale and setting there LANG="C.UTF-8".

In some cases the individual services have separate configuration for locales. For example when using Apache you might want to set it in /etc/apache2/envvars:

```
export LANG='en_US.UTF-8'
export LC_ALL='en_US.UTF-8'
```

Using custom certificate authority

Weblate does verify SSL certificates during HTTP requests. In case you are using custom certificate authority which is not trusted in default bundles, you will have to add its certificate as trusted.

The preferred approach is to do this at system level, please check your distro documentation for more details (for example on debian this can be done by placing the CA certificate into /usr/local/share/cacertificates/ and running update-ca-certificates).

Once this is done, system tools will trust the certificate and this includes Git.

For Python code, you will need to configure requests to use system CA bundle instead of the one shipped with it. This can be achieved by placing following snippet to settings.py (the path is Debian specific):

```
import os
```

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"

Compressing client assets

Weblate comes with a bunch of JavaScript and CSS files. For performance reasons it is good to compress them before sending to a client. In default configuration this is done on the fly at cost of little overhead. On big installations, it is recommended to enable offline compression mode. This needs to be done in the configuration and the compression has to be triggered on every Weblate upgrade.

The configuration switch is simple by enabling django.conf.settings.COMPRESS_OFFLINE and configuring django.conf.settings.COMPRESS_OFFLINE_CONTEXT (the latter is already included in the example configuration):

COMPRESS_OFFLINE = True

On each deploy you need to compress the files to match current version:

weblate compress

???: The official Docker image has this feature already enabled.

??:

Common Deployment Scenarios? Serving static files

Running server

You will need several services to run Weblate, the recommended setup consists of:

Database server (see *Weblate* ??????????)

Cache server (see *Enable caching*)

Frontend web server for static files and SSL termination (see Serving static files)

WSGI server for dynamic content (see Sample configuration for NGINX and uWSGI)

Celery for executing background tasks (see Background tasks using Celery)

There are some dependencies between the services, for example cache and database should be running when starting up Celery or uwsgi processes.

In most cases, you will run all services on single (virtual) server, but in case your installation is heavy loaded, you can split up the services. The only limitation on this is that Celery and Wsgi servers need access to *DATA_DIR*.

The WSGI process has to be executed under the same user the Celery process, otherwise files in the *DATA_DIR* will be stored with mixed ownership, leading to runtime issues.

Running web server

Running Weblate is not different from running any other Django based program. Django is usually executed as uWSGI or fcgi (see examples for different webservers below).

For testing purposes, you can use the built-in web server in Django:

weblate runserver

DO NOT USE THIS SERVER IN A PRODUCTION SETTING. It has not gone through security audits or performance tests. See also Django documentation on runserver.

The Django built-in server serves static files only with *DEBUG* enabled as it is intended for development only. For production use, please see wsgi setups in *Sample configuration for NGINX and uWSGI*, *Sample configuration for Apache, Sample configuration for Apache and Gunicorn*, and *Serving static files*.

Serving static files

2222 2.4 222: Prior to version 2.4, Weblate didn't properly use the Django static files framework and the setup was more complex.

Django needs to collect its static files in a single directory. To do so, execute weblate collectstatic --- noinput. This will copy the static files into a directory specified by the STATIC_ROOT setting (this defaults to a static directory inside *DATA_DIR*).

It is recommended to serve static files directly from your web server, you should use that for the following paths:

Serves static files for Weblate and the admin interface (from defined by STATIC_ROOT).

Used for user media uploads (e.g. screenshots).

Should be rewritten to rewrite a rule to serve /static/favicon.ico.

??:

Compressing client assets? Deploying Django? Deploying static files

Content security policy

The default Weblate configuration enables weblate.middleware.SecurityMiddleware middleware which sets security related HTTP headers like Content-Security-Policy or X-XSS-Protection. These are by default set up to work with Weblate and its configuration, but this might need customization for your environment.

??:

CSP_SCRIPT_SRC12CSP_IMG_SRC12CSP_CONNECT_SRC12CSP_STYLE_SRC12CSP_FONT_SRC

Sample configuration for NGINX and uWSGI

To run production webserver, use the wsgi wrapper installed with Weblate (in virtual env case it is installed as \sim /weblate-env/lib/python3.7/site-packages/weblate/wsgi.py). Don't forget to set the Python search path to your virtualenv as well (for example using virtualenv = /home/user/weblate-env in uWSGI).

The following configuration runs Weblate as uWSGI under the NGINX webserver.

Configuration for NGINX (also available as 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...
\hookrightarrowyour 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;
    }
}
```

Configuration for uWSGI (also available as 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]
             = python3
plugins
             = true
master
protocol
             = uwsgi
             = 127.0.0.1:8080
socket
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
            = 8192
buffer-size
# 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
```

??:

#

How to use Django with uWSGI

Sample configuration for Apache

It is recommended to use prefork MPM when using WSGI with Weblate.

The following configuration runs Weblate as WSGI, you need to have enabled mod_wsgi (available as weblate/examples/apache.conf):

```
#
 VirtualHost for Weblate
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/
⇔weblate-env
# and DATA DIR is set to /home/weblate/data, please adjust paths to match_
\hookrightarrowyour setup.
<VirtualHost *:80>
   ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org
    # DATA_DIR/static/favicon.ico
   Alias /favicon.ico /home/weblate/data/static/favicon.ico
    # DATA_DIR/static/
   Alias /static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>
```

(????????)

```
# DATA_DIR/media/
   Alias /media/ /home/weblate/data/media/
   <Directory /home/weblate/data/media/>
       Require all granted
   </Directory>
   # Path to your Weblate virtualenv
   WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env_
→user=weblate
   WSGIProcessGroup weblate
   WSGIApplicationGroup %{GLOBAL}
   WSGIScriptAlias / /home/weblate/weblate-env/lib/python3.7/site-
WSGIPassAuthorization On
   <Directory /home/weblate/weblate-env/lib/python3.7/site-packages/</pre>
→weblate/>
       <Files wsgi.py>
       Require all granted
       </Files>
   </Directory>
</VirtualHost>
```

Weblate requires Python 3, so please make sure you are running Python 3 variant of the modwsgi. Usually it is available as a separate package, for example libapache2-mod-wsgi-py3.

??:

System locales and encoding? How to use Django with Apache and mod_wsgi

Sample configuration for Apache and Gunicorn

The following configuration runs Weblate in Gunicorn and Apache 2.4 (available as weblate/examples/ apache.gunicorn.conf):

```
VirtualHost for Weblate using qunicorn 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_
\hookrightarrowyour 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/
    <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

Running Weblate under path

It is recommended to use prefork MPM when using WSGI with Weblate.

A sample Apache configuration to serve Weblate under /weblate. Again using mod_wsgi (also available as 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...
\hookrightarrowyour 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
    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/</pre>
→weblate/>
        <Files wsgi.py>
```

```
Require all granted
</Files>
</Directory>
```

</VirtualHost>

Additionally, you will have to adjust weblate/settings.py:

URL_PREFIX = "/weblate"

Background tasks using Celery

?????? 3.2 ????.

Weblate uses Celery to process background tasks. A typical setup using Redis as a backend looks like this:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

??:

Redis broker configuration in Celery

For development, you might want to use eager configuration, which does process all tasks in place, but this will have performance impact on Weblate:

```
CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True
```

You should also start the Celery worker to process the tasks and start scheduled tasks, this can be done directly on the command line (which is mostly useful when debugging or developing):

./weblate/examples/celery start
./weblate/examples/celery stop

The Celery process has to be executed under the same user as the WSGI process, otherwise files in the *DATA_DIR* will be stored with mixed ownership, leading to runtime issues.

Running Celery as system service

Most likely you will want to run Celery as a daemon and that is covered by Daemonization. For the most common Linux setup using systemd, you can use the example files shipped in the examples folder listed below.

Systemd unit to be placed as /etc/systemd/system/celery-weblate.service:

```
[Unit]
Description=Celery Service (Weblate)
After=network.target
[Service]
Type=forking
User=weblate
Group=weblate
Group=weblate
EnvironmentFile=/etc/default/celery-weblate
WorkingDirectory=/home/weblate
RuntimeDirectory=/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

Environment configuration to be placed as /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 concurency 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"
```

Logrotate configuration to be placed as /etc/logrotate.d/celery:

/var/log/celery/*.log {
 weekly
 missingok
 rotate 12
 compress
 notifempty
}

Periodic tasks using Celery beat

Weblate comes with built-in setup for scheduled tasks. You can however define additional tasks in settings.py, for example see *Lazy commits*.

The tasks are supposed to be executed by Celery beats daemon. In case it is not working properly, it might not be running or its database was corrupted. Check the Celery startup logs in such case to figure out root cause.

Monitoring Celery status

You can use *celery_queues* to see current length of Celery task queues. In case the queue will get too long, you will also get configuration error in the admin interface.

The Celery errors are by default only logged into Celery log and are not visible to user. In case you want to have overview on such failures, it is recommended to configure *Collecting error reports*.

??:

```
Configuration and defaults??Workers Guide??Daemonization??Monitoring and Management Guide??celery_queues
```

Monitoring Weblate

Weblate provides the /healthz/ URL to be used in simple health checks, for example using Kubernetes.

Collecting error reports

Weblate, as any other software, can fail. In order to collect useful failure states we recommend to use third party services to collect such information. This is especially useful in case of failing Celery tasks, which would otherwise only report error to the logs and you won't get notified on them. Weblate has support for the following services:

Sentry

Weblate has built-in support for Sentry. To use it, it's enough to set SENTRY_DSN in the settings.py:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

Weblate has built-in support for Rollbar. To use it, it's enough to follow instructions for Rollbar notifier for Python. In short, you need to adjust 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": "master",
    "root": "/absolute/path/to/code/root",
}
```

Everything else is integrated automatically, you will now collect both server and client side errors.

Migrating Weblate to another server

Migrating Weblate to another server should be pretty easy, however it stores data in few locations which you should migrate carefully. The best approach is to stop Weblate for the migration.

Migrating database

Depending on your database backend, you might have several options to migrate the database. The most straightforward one is to dump the database on one server and import it on the new one. Alternatively you can use replication in case your database supports it.

The best approach is to use database native tools, as they are usually the most effective (e.g. **mysqldump** or **pg_dump**). If you want to migrate between different databases, the only option might be to use Django management to dump and import the database:

```
# Export current data
weblate dumpdata > /tmp/weblate.dump
# Import dump
weblate loaddata /tmp/weblate.dump
```

Migrating VCS repositories

The VCS repositories stored under *DATA_DIR* need to be migrated as well. You can simply copy them or use **rsync** to do the migration more effectively.

Other notes

Don't forget to move other services Weblate might have been using like Redis, Cron jobs or custom authentication backends.

Weblate ???

Installing using Docker Installing on OpenShift Installing on Kubernetes

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

Bitnami Weblate ????

Weblate Cloudron ??????



YunoHost ? Weblate



yunohost app install https://github.com/YunoHost-Apps/weblate_ynh

Upgrading Weblate

Docker image upgrades

The official Docker image (see *Installing using Docker*) has all upgrade steps integrated. There are no manual step besides pulling latest version.

Generic upgrade instructions

Before upgrading, please check the current <u>2222222</u> as they might have changed. Once all requirements are installed or updated, please adjust your settings.py to match changes in the configuration (consult settings_example.py for correct values).

Always check *Version specific instructions* before upgrade. In case you are skipping some versions, please follow instructions for all versions you are skipping in the upgrade. Sometimes it's better to upgrade to some intermediate version to ensure a smooth migration. Upgrading across multiple releases should work, but is not as well tested as single version upgrades.

?!?: 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 *Backing up and moving 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 Running server and Serving static files):

weblate collectstatic -- noinput

6.Compress JavaScript and CSS files (optional, see Compressing client assets):

weblate compress

7.If you are running version from Git, you should also regenerate locale files every time you are upgrading. You can do this by invoking:

weblate compilemessages

8. Verify that your setup is sane (see also *Production setup*):

weblate check --deploy

9. Restart celery worker (see Background tasks using 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 Generic upgrade instructions in order to perform update.

Notable configuration or dependencies changes:

There are several changes in settings_example.py, most notable middleware changes, please adjust your settings accordingly.

There are new file formats, you might want to include them in case you modified the WEBLATE_FORMATS.

There are new quality checks, you might want to include them in case you modified the CHECK_LIST.

There 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.

??:

Generic upgrade instructions

Upgrade from 4.1 to 4.2

Please follow Generic upgrade instructions 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 CEL-ERY_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.

??:

Generic upgrade instructions

Upgrade from 4.2 to 4.3

Please follow Generic upgrade instructions 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* [2][2][2][2][2] 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 SES-SION_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*.

Changed in 4.3.1: The Celery configuration was changed to add memory queue. Please adjust your startup scripts and CELERY_TASK_ROUTES setting.

Changed in 4.3.2: The post_update method of addons now takes extra skip_push parameter.

??:

Generic upgrade instructions

Upgrade from 4.3 to 4.4

Please follow Generic upgrade instructions 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 2 MariaDB.

Changed in 4.4.1: *Monolingual gettext* now uses both msgid and msgctxt when present. This will change IDs of translation strings in such files. Please make sure you commit pending changes in such files prior upgrading.

Changed in 4.4.1: Increased minimal required version of translate-toolkit to address several file format issues.

Generic upgrade instructions

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 dabatase 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 ??????????

```
# 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",
         # Set to empty string for default
        "PORT": "",
         # Additional database options
        "OPTIONS": {
             # In case of using an older MySQL server, which has MyISAM as.
\hookrightarrowa 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...
\rightarrowerror:
             "connect_timeout": 28800,
        },
    },
    "postgresql": {
        # Database engine
        "ENGINE": "django.db.backends.postgresgl",
        # 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:

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*.

Backing up and moving Weblate

Automated backup using BorgBackup

?????? 3.9 ????.

Weblate has built-in support for creating service backups using BorgBackup. Borg creates space-effective encrypted backups which can be safely stored in the cloud. The backups can be controlled in the management interface on the *Backups* tab.

22222 4.4.1 222: Both PostgreSQL and MySQL/MariaDB databases are included in the automated backups.

The backups using Borg are incremental and Weblate is configured to keep following backups:

14 daily backups

8 weekly backups

6 monthly backups

Weblate Dashboard	Projects 🗸 🛛 Languages	← Checks ~				~ +	Add 🗕 🍓 🗕	•••
Manage / Backups Backup process triggered								
Backup process enggered								
Weblate status Backups	Translation memory	Performance report	SSH keys	Alerts	Repositories	Users	Appearance	
Tools Billing								
Backup service: /tmp/tmpd6i	6hug9weblate							0
Backup service credentials							Dec. 15, 202	20
Backup repository	/tmp/tmpd6i6hug9web	late [
Passphrase qVnji6y*fsRcY0mt50p7&bo3(^^VamUTBtG\$i0I3ACcbwXqjWo C The passphrase is used to encrypt the backups and is necessary to restore them.								
SSH key Download private key								
	The private key is need	ed to access the remote ba	ckup repository.					
Deleted the oldest backups	s						Dec. 15, 202	20
Backup performed							Dec. 15, 202	20
Repository initialization							Dec. 15, 202	20
Turn off Perform b	ackup Delete							
Activate support package								G
The support packages include p	riority e-mail support, or clo	oud backups of your Weblat	te installation.					
Activation token								
Disco anton the activation to be								
Please enter the activation toke	en obcained when making tr	le subscription.						
Activate Purchase	support package							
Add backup service								C
Backup repository URL								
Use /path/to/repo for local backups or user@host:/path/to/repo for remote SSH backups.								
Add								

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

Borg encryption key

BorgBackup creates encrypted backups and without a passphrase you will not be able to restore the backup. The passphrase is generated when adding new backup service and you should copy it and keep it in a secure place.

In case you are using *Weblate provisioned backup storage*, please backup your private SSH key as well — it is used to access your backups.

??:

borg init

Weblate provisioned backup storage

The easiest approach to backup your Weblate instance is to purchase backup service at weblate.org. The process of activating can be performed in few steps:

1.Purchase backup service on https://weblate.org/support/#backup.

2. Enter obtained key in the management interface, see Integrating support.

3.Weblate will connect to the cloud service and obtain access information for the backups.

4. Turn on the new backup configuration on the Backups tab.

5.Backup Borg credentials in order to be able to restore the backups, see *Borg encryption key*.

The manual step of turning on is there for your safety. Without your consent no data is sent to the backup repository obtained through the registration process.

Using custom backup storage

You can also use your own storage for the backups. SSH can be used to store backups on the remote destination, the target server needs to have BorgBackup installed.

??:

General in the Borg documentation

Local filesystem

When running Weblate in Docker, please make sure that the backup location is exposed as a volume from the Weblate container. Otherwise the backups would be discarded by Docker on container restart.

One option is to place backups in existing volume. For example choose /app/data/borgbackup. This is existing volume in the container.

You can also add new container for the backups in the Docker compose file and use for example /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 will not be able to write the backups there.

Remote backups using SSH are supported. The SSH server needs to have BorgBackup installed. Weblate connects to the server using SSH key, please make sure the Weblate SSH key is accepted by the server (see *Weblate SSH 202*).

Weblate provisioned backup storage provides you automated remote backups.

Restoring from BorgBackup

- 1.Restore access to your backup repository and prepare your backup passphrase.
- 2.List backup existing on the server using borg list REPOSITORY.
- 3. Restore the desired backup to current directory using borg extract REPOSITORY:: ARCHIVE.
- 4.Restore the database from the SQL dump placed in the backup directory in the Weblate data dir (see *Dumped data for backups*).
- 5.Copy Weblate configuration (backups/settings.py, see *Dumped data for backups*) to the correct location, see [2][2][2].
- 6.Copy the whole restored data dir to location configured by DATA_DIR.

The Borg session might look like:

```
$ borg list /tmp/xxx
Enter passphrase for key /tmp/xxx:
2019-09-26T14:56:08 Thu, 2019-09-26 14:56:08_
→[de0e0f13643635d5090e9896bdaceb92a023050749ad3f3350e788f1a65576a5]
$ borg extract /tmp/xxx::2019-09-26T14:56:08
Enter passphrase for key /tmp/xxx:
```

??:

borg list?borg extract

Manual backup

Depending on what you want to save, back up the type data Weblate stores in each respective place.

```
The set of the set of
```

```
SILENCED_SYSTEM_CHECKS.append("weblate.1028")
```

Database

The actual storage location depends on your database setup.

The database is the most important storage. Set up regular backups of your database, without it all your translation setup will be gone.

Native database backup

The recommended approach is to do dump of the database using database native tools such as **pg_dump** or **mysql-dump**. It usually performs better than Django backup and restores complete tables with all data.

You can restore this backup in newer Weblate release, it will perform any necessary migrations when running in migrate. Please consult *Upgrading Weblate* on more detailed information how to perform upgrade between versions.

Django database backup

Alternatively you can backup database using Django's dumpdata command. That way the backup is database agnostic and can be used in case you want to change database backend.

Prior to restoring you need to be running exactly same Weblate version as was used when doing backups. 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.

Once this is done, some entries will be already 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 management shell (see *Invoking management commands*):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

????

If you have enough backup space, simply backup the whole *DATA_DIR*. This is safe bet even if it includes some files you don't want. The following sections describe in detail what you should back up and what you can skip.

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 *Background tasks using Celery*). 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 by default saved as plain text, but they can also be compressed or entirely skipped by 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 and you do not have to backup the repositories on the Weblate side. They can be cloned again from the upstream locations 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.

User uploaded files

Stored in DATA_DIR /media.

You should back up user uploaded files (e.g. Visual context for strings).

Celery tasks

The Celery tasks queue might contain some info, but is usually not needed for a backup. At most you will lose updates that have not yet been processed to translation memory. It is recommended to perform the fulltext or repository updates upon restoring anyhow, so there is no problem in losing these.

??:

Background tasks using Celery

Command line for manual backup

Using a cron job, you can set up a bash command to be executed on a daily basis, for instance:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).
→xz backups vcs ssh home media fonts secret
```

The string between quotes after XZ_OPT allows you to choose your xz options, for instance the amount of memory used for compression; see https://linux.die.net/man/1/xz

You can adjust the list of folders and files to your needs. For instance, to avoid saving the translation memory (in backups folder), you could use:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).

$\dotsz backups/database.sql backups/settings.py vcs ssh home media fonts_

$\dotsecret
```

Restoring manual backup

1.Restore all data you have backed up.

2.Update all repositories using updategit.

weblate updategit --all

Moving a Weblate installation

Relocate your installation to a different system by following the backup and restore instructions above.

Upgrading from Python 2 to Python 3 Migrating from other databases to PostgreSQL

??

??????

The default setup for Weblate is to use python-social-auth, a form on the website to handle registration of new users. After confirming their e-mail a new user can contribute or authenticate by using one of the third party services.

You can also turn off registration of new users using *REGISTRATION_OPEN*.

Authentication backends

The built-in solution of Django is used for authentication, including various social options to do so. Using it means you can import the user database of other Django-based projects (see *Migrating from Pootle*).

Django can additionally be set up to authenticate against other means too.

??:

Authentication settings describes how to configure authentication in the official Docker image.

Social authentication

Thanks to Welcome to Python Social Auth's documentation!, Weblate support authentication using many third party services such as GitLab, Ubuntu, Fedora, etc.

Please check their documentation for generic configuration instructions in Django Framework.

By default, Weblate relies on third-party authentication services to provide a validated e-mail address. If some of the services you want to use don't support this, please enforce e-mail validation on the Weblate side by configuring FORCE_EMAIL_VALIDATION for them. For example:

SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True

??:

Pipeline

Enabling individual backends is quite easy, it's just a matter of adding an entry to the AUTHENTICA-TION_BACKENDS setting and possibly adding keys needed for a given authentication method. Please note that some backends do not provide user e-mail by default, you have to request it explicitly, otherwise Weblate will not be able to properly credit contributions users make.

??:

Python Social Auth backend

OpenID authentication

For OpenID-based services it's usually just a matter of enabling them. The following section enables OpenID authentication for OpenSUSE, Fedora and Ubuntu:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.suse.OpenSUSEOpenId",
    "social_core.backends.ubuntu.UbuntuOpenId",
    "social_core.backends.fedora.FedoraOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

??:

OpenID

GitHub authentication

You need to register an OAuth application on GitHub and then tell Weblate all its secrets:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.github.GithubOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)
# Social auth backends setup
SOCIAL AUTH GITHUB KEY = "GitHub Client ID"
```

(?????????)

```
SOCIAL_AUTH_GITHUB_SECRET = "GitHub Client Secret"
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]
```

The GitHub should be configured to have callback URL as https://example.com/accounts/complete/github/.

Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

??:

GitHub

Bitbucket authentication

You need to register an application on Bitbucket and then tell Weblate all its secrets:

```
# 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
```

Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

??:

Bitbucket

Google OAuth 2

To use Google OAuth 2, you need to register an application on <https://console.developers.google.com/> and enable the Google+ API.

The redirect URL is https://WEBLATE SERVER/accounts/complete/google-oauth2/

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)
# Social auth backends setup
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = "Client ID"
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = "Client secret"
```

Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

??:

Google

Facebook OAuth 2

As per usual with OAuth 2 services, you need to register your application with Facebook. Once this is done, you can set up Weblate to use it:

The redirect URL is https://WEBLATE SERVER/accounts/complete/facebook/

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.facebook.FacebookOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)
# Social auth backends setup
SOCIAL_AUTH_FACEBOOK_KEY = "key"
SOCIAL_AUTH_FACEBOOK_SECRET = "secret"
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
```

Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

??:

Facebook

GitLab OAuth 2

For using GitLab OAuth 2, you need to register an application on <https://gitlab.com/profile/applications>.

The redirect URL is https://WEBLATE SERVER/accounts/complete/gitlab/ and ensure you mark the *read_user* scope.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.gitlab.GitLabOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)
# Social auth backends setup
SOCIAL_AUTH_GITLAB_KEY = "Application ID"
SOCIAL_AUTH_GITLAB_SECRET = "Secret"
SOCIAL_AUTH_GITLAB_SECRET = "Secret"
SOCIAL_AUTH_GITLAB_SCOPE = ["read_user"]
# If you are using your own GitLab
# SOCIAL_AUTH_GITLAB_API_URL = 'https://gitlab.example.com/'
```

Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

??:

GitLab

Microsoft Azure Active Directory

Weblate can be configured to use common or specific tenants for authentication.

```
The redirect URL is https://WEBLATE SERVER/accounts/complete/azuread-oauth2/ for com-
mon and https://WEBLATE SERVER/accounts/complete/azuread-tenant-oauth2/ for tenant-
specific authentication.
```

```
# 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 = ""
```

Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

??:

Microsoft Azure Active Directory

Slack

For using Slack OAuth 2, you need to register an application on <https://api.slack.com/apps>.

```
The redirect URL is 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 = ""
```

Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

??:

Slack

Turning off password authentication

E-mail and password authentication can be turned off by removing social_core.backends.email. EmailAuth from AUTHENTICATION_BACKENDS. Always keep weblate.accounts.auth. WeblateUserBackend there, it is needed for core Weblate functionality.

You can still use password authentication for the admin interface, for users you manually create there. Just navigate to /admin/.

For example authentication using only the openSUSE Open ID provider can be achieved using the following:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

Password authentication

The default settings.py comes with a reasonable set of AUTH_PASSWORD_VALIDATORS:

Passwords can't be too similar to your other personal info.

Passwords must contain at least 10 characters.

Passwords can't be a commonly used password.

Passwords can't be entirely numeric.

Passwords can't consist of a single character or only whitespace.

Passwords can't match a password you have used in the past.

You can customize this setting to match your password policy.

Additionally you can also install django-zxcvbn-password which gives quite realistic estimates of password difficulty and allows rejecting passwords below a certain threshold.

SAML authentication

2????? 4.1.1 ????.

Please follow the Python Social Auth instructions for configuration. Notable differences:

Weblate supports single IDP which has to be called weblate in SOCIAL_AUTH_SAML_ENABLED_IDPS.

The SAML XML metadata URL is /accounts/metadata/saml/.

Following settings are automatically filled in: SOCIAL_AUTH_SAML_SP_ENTITY_ID, SO-CIAL_AUTH_SAML_TECHNICAL_CONTACT, SOCIAL_AUTH_SAML_SUPPORT_CONTACT

Example configuration:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.saml.SAMLAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)
# Social auth backends setup
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbnl+ev0peYzxFyF5sQA==",
        "attr_name": "username",
        "attr_email": "email",
```

(?????????)

}

??:

Configuring SAML in Docker?SAML

LDAP authentication

LDAP authentication can be best achieved using the *django-auth-ldap* package. You can install it via usual means:

```
# Using PyPI
pip install django-auth-ldap>=1.3.0
# Using apt-get
apt-get install python-django-auth-ldap
```

With django-auth-ldap older than 1.3.0 the <u>2222222222222</u> will not work properly for newly created users.

There are some incompatibilities in the Python LDAP 3.1.0 module, which might prevent you from using that version. If you get error AttributeError: 'module' object has no attribute '_trace_level', downgrading python-ldap to 3.0.0 might help.

Once you have the package installed, you can hook it into the Django authentication:

```
# Add LDAP backed, keep Django one if you want to be able to sign in
# even without LDAP for admin account
AUTHENTICATION_BACKENDS = (
    "django_auth_ldap.backend.LDAPBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)
# LDAP server address
AUTH_LDAP_SERVER_URI = "ldaps://ldap.example.net"
# DN to use for authentication
AUTH_LDAP_USER_DN_TEMPLATE = "cn=%(user)s,o=Example"
# Depending on your LDAP server, you might use a different DN
# like:
# AUTH_LDAP_USER_DN_TEMPLATE = 'ou=users,dc=example,dc=com'
 List of attributes to import from LDAP upon sign in
# Weblate stores full name of the user in the full_name attribute
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "name",
# Use the following if your LDAP server does not have full name
    # Weblate will merge them later
    # 'first_name': 'givenName',
# 'last_name': 'sn',
    # Email is required for Weblate (used in VCS commits)
"email": "mail",
}
# Hide the registration form
REGISTRATION OPEN = False
```

You should remove 'social_core.backends.email.EmailAuth' from the AUTHENTICA-TION_BACKENDS setting, otherwise users will be able to set their password in Weblate, and authenticate using that. Keeping 'weblate.accounts.auth.WeblateUserBackend' is still needed in order to make permissions and facilitate anonymous users. It will also allow you to sign in using a local admin account, if you have created it (e.g. by using createadmin).

Using bind password

If you can not use direct bind for authentication, you will need to use search, and provide a user to bind for the search. For example:

```
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)
 \rightarrow 
)
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 authentication

CAS authentication can be achieved using a package such as *django-cas-ng*.

Step one is disclosing the e-mail field of the user via CAS. This has to be configured on the CAS server itself, and requires you run at least CAS v2 since CAS v1 doesn't support attributes at all.

Step two is updating Weblate to use your CAS server and attributes.

To install django-cas-ng:

pip install django-cas-ng

Once you have the package installed you can hook it up to the Django authentication system by modifying the settings.py file:

```
# Add CAS backed, keep the Django one if you want to be able to sign in
# even without LDAP for the admin account
AUTHENTICATION_BACKENDS = (
    "django_cas_ng.backends.CASBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)
# CAS server address
CAS_SERVER_URL = "https://cas.example.net/cas/"
# Add django_cas_ng somewhere in the list of INSTALLED_APPS
INSTALLED_APPS = (..., "django_cas_ng")
```

Finally, a signal can be used to map the e-mail field to the user object. For this to work you have to import the signal from the *django-cas-ng* package and connect your code with this signal. Doing this in settings file can cause problems, therefore it's suggested to put it:

In your app config's <code>django.apps.AppConfig.ready()</code> method

In the project's urls.py file (when no models exist)

??:

Django CAS NG

Configuring third party Django authentication

Generally any Django authentication plugin should work with Weblate. Just follow the instructions for the plugin, just remember to keep the Weblate user backend installed.

??:

LDAP authentication?CAS authentication

Typically the installation will consist of adding an authentication backend to AUTHENTICATION_BACKENDS and installing an authentication app (if there is any) into INSTALLED_APPS:

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)
INSTALLED_APPS += (
    # Install authentication app here
)
```

???????

????

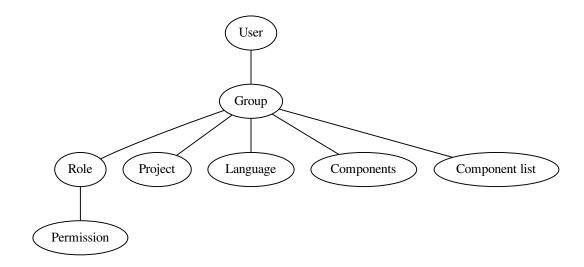
Weblate ???

??:

22222222

 <u>???????</u>?

Weblate Dashboard F	Projects 🗸 Languages 🗸 Checks 🗸		🔧 🕂 Add - 💮 - 🚥			
🛞 WeblateOrg / Settings						
Basic Access Workflow	Components					
Controlo de acesso 🗊						
🔘 🖬 Public	Protected	🔘 💄 Private	🔘 🔯 Custom			
Visible to everybody.	Visible to everybody.	Visible only to chosen users.	Permissions are not managed in			
Any authenticated user can	Only chosen users can contribute.	Only chosen users can contribute.	Weblate.			
contribute.			Only use this if you know what you are doing, enabling it might revoke your access to this project.			
You do not have permission to cha	nge project access control. Check your billing st a	atus				
	Powered by Weblate 4.4 About Weblate Leg	al Contact Documentation Donate to Weblat	te			
20222202222222222 20222222222222222222			222222222222222222222222222222222222222			



?????????

<u>? ? ? ? ? ?</u>	?!?!?!?!?!?	<u>???</u> ? ?????? <u>?</u> .	???????	???????????????????????????????????????	????	?????	????????	????	????????	??????	??????	???	?????
???????	???????	?????											
???????	???????	?????											
???????	???????	?											
???????	???????	????											
?????? 3	.11 ???.												
???????	???????	???????????????????????????????????????	?????	???????????????????????????????????????									
???????	??????	???? <i>!?!?</i> ?????	:										
Webla	ite Dashb	oard Projects -	Languages						,	~ +	Add 🗕 🍕	-	•••
💮 Weblate	eOrg / Manag	ge users											
Users													()
03013	Full		Last										U
Username		E-mail	login	Administration	Billing	Glossary	Languages	Memory	Screenshots	Sources	Translate	VCS	5
testuser	Weblate v Test	weblate@example.org	22 seconds ago										
Once all it	s nermissions	are removed, the user v	vill he remo	ved from the proi	ect								
once attra	5 permissions	are removed, ene user (fearion che proj									
Add a use	er					Invite ne	ew user						
User to a	dd					E-mail							
Diesse tw	e in an evictin	weblate account nam	ne or e-mail	address									
Please typ	e in an existin	g Weblate account nan	ne or e-mail	address.		Usernan	ne						
Please typ	e in an existin	ng Weblate account nan	ne or e-mail	address.				ontain lett	ers, numbers o	or the follo	owing charac	cters: (@.

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

Full name

Invite

??:

??????????

Administration

Glossary

Screenshots

Sources
222 Monolingual components 222222222222

Translate

Review 22222222222222222

 2/2/2/2/2
 12/2
 12/2/2/2
 12/2/2/2
 12/2/2/2
 12/2/2/2
 12/2/2
 12/2/2
 12/2/2
 12/2/2
 12/2/2
 12/2/2
 12/2/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 12/2
 <th12/2</th>
 12/2
 12/2

?????

[????????? [Administration, Edit source] [???????? [Administration] [2][2][2][2] [Administration, Edit source, Power user, Review strings, Translate] [??????? [Administration, Review strings] [?????? [Administration, Edit source, Power user] [2][2][2] [Administration, Edit source, Power user, Review strings, Translate] [22][2] [Add suggestion, Administration, Edit source, Power user, Review strings, Translate] [?????? [Administration] [2][2][2] [Administration, Edit source, Power user, Review strings, Translate] [22][22][22][22] [Administration, Manage languages, Power user] [????????? [Administration, Manage languages] 2222222222 [Administration, Manage languages] [22] [Administration, Edit source, Power user, Review strings, Translate] [22] [22] [22] [22] [Access repository, Administration, Manage repository, Power user] [22] [22] [22] [22] [22] [Administration, Manage repository] upstream 22222222222 [Access repository, Administration, Manage repository, Power user] 2222222222 [Administration, Manage repository] ???????? ????????? ????????????????

???????

?????????

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 the 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 *Supported file formats*.

?!?: 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

[2][2][2][2] 3.2 [2][2]: An interface for adding projects and components is included, and you no longer have to use *The Django admin interface*.

[2][2][2][2] 3.4 [2][2]: 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 [2][2]), 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:

W Weblate	Dashboard Projects - Languag	es 🗸 Checks 🗸			+ Add - 🍓 -	•••
💮 Your profile	/ Billing					
Billing plan		Ũ	Invoices			
Current plan	Basic plan (Active)		Invoice period	Invoice amount	Download invoice	
Monthly price	19 EUR		12/14/2020 - 12/16/2020	19.0 EUR	Not available	
Yearly price	199 EUR					
Strings limit	Used 0					
Languages limit	Used 0					
Last invoice	2020-12-14 - 2020-12-16					
Projects limit	Used 0 of 1					
Projects	No projects currently Add new tr assigned!	anslation project				
Terminat	e billing plan					

Powered by Weblate 4.4 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:

🔮 Weblate Dashboard Projects - Languages - Checks -	+ Add -	•	•••
S Create project			
Add new translation project			0
Project name 🕦			
WeblateOrg			
Display name			
URL slug 🕕			
weblateorg			
Name used in URLs and filenames.			
Project website 🛈			
https://weblate.org/			
Main website of translated project.			
Mailing list ①			
weblate@lists.cihar.com			
Mailing list for translators.			
Translation instructions ①			
https://weblate.org/contribute/			
You can use Markdown and mention users by @username.			10
Billing ()			
Weblate Test (Basic plan)			~
montace receivable promy			
Save			

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

After creating the project, you are taken directly to the project page:

()) Weblate	Dashboard	Projects 🗸	Languag	ges - Checks	•					+ Add -	- 🌒	•••
🛞 WeblateOrg											translated	100%
Components	Languages	Info	Search	Glossaries	Insights +	Files 🛨	Tools -	Manage 👻	Share 🗸		⊙ Watch	ing 🗸
Nothing to list he	re.											
Add new tran	slation comp	onent										

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

Creating a new translation component can be initiated via a single click there. The process of creating a component is multi-staged and automatically detects most translation parameters. There are several approaches to creating component:

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 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 -	+ Add -	•••
③ 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.		
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 4.4 About Weblate Legal Contact Documentation Donate to Weblate

On the next page, you are presented with a list of discovered translatable resources:

 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 P0 file, Filemask weblate/langdata/locale/*/LC_MESSAGES/django.po File format gettext P0 file, Filemask weblate/locale/*/LC_MESSAGES/django.po 	+ Add -	. - •
Choose translation files to import () Specify configuration manually File format Android String Resource, Filemask app/src/main/res/values-*/strings.xml File format gettext P0 file, Filemask weblate/langdata/locale/*/LC_MESSAGES/django.po		
 Specify configuration manually File format Android String Resource, Filemask app/src/main/res/values-*/strings.xml File format gettext P0 file, Filemask weblate/langdata/locale/*/LC_MESSAGES/django.po 		0
 File format Android String Resource, Filemask app/src/main/res/values-*/strings.xml File format gettext P0 file, Filemask weblate/langdata/locale/*/LC_MESSAGES/django.po 		
File format gettext P0 file, Filemask weblate/langdata/locale/*/LC_MESSAGES/django.po		
File format gettext P0 file, Filemask weblate/locale/*/LC_MESSAGES/django.po		
○ File format gettext PO file,Filemask weblate/locale/*/LC_MESSAGES/djangojs.po		
Continue		
○ File format gettext PO file,Filemask weblate/locale/*/LC_MESSAGES/djangojs.po		+ Add +

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

As a last step, you review the translation component info and fill in optional details:

Create component		
Detected license as N	IT, please check whether it is correct.	
Add new translatio	n component	(
Project 🛈		
WeblateOrg		`
Component name (
Language names		
Display name		
URL slug 🕕		
language-names		
Name used in URLs a	nd filenames.	
Version control sys	tem 🕜	
Git		,
Version control syste merge requests.	m to use to access your repository containing translations. You can also choose additional integration with third party providers to submit	
Source code reposi	tory ()	
https://github.con	/WeblateOrg/demo.git	
URL of a repository,	use weblate://project/component to share it with other component.	
Repository branch	٥	
Repository branch to	translate	
Repository push UF	LO	
URL of a nush reneri	tory, pushing is turned off if empty.	
Push branch 🕧	ony paoning is connection in competi-	
Branch for pushing c	hanges, leave empty to use repository branch	
Repository browse	r ()	
https://github.com	/WeblateOrg/demo/blob/{{branch}}/{{filename}}#L{{line}}	
	owser, use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders.	
File format 🕕		
gettext PO file		`
Filemask 🕕		
weblate/langdata/	locale/*/LC_MESSAGES/django.po	
Path of files to trans	late relative to repository root, use * instead of language code, for example: po/*.po or locale/*/LC_MESSAGES/django.po.	
Monolingual base l	anguage file 🕖	
5 1		
	ion base file, containing all strings and their source; it is recommended for monolingual translation formats.	
☑ Edit base file Whether users will be	e able to edit the base file for monolingual translations.	
Intermediate langu	age rice ()	
Filename of interme	diate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.	
Template for new t		
weblate/langdata/		
	for creating new translations. For gettext choose .pot file.	
Translation license		
	U ic License v3.0 or later	,
Adding new transla		
Create new langua		`
	sts for creating new translations.	
Language code sty		
Default based on t		`
	code used to generate the filename for translations created by Weblate.	
Language filter 🛈		
Language filter う ^(cs he hu)\$		
Language filter ① ^(cs he hu)\$ Regular expression u	sed to filter translation files when scanning for filemask.	
Language filter () ^(cs he hu)\$ Regular expression u Source language ()		
Language filter ① ^(cs he hu)\$ Regular expression u Source language ① English		`
Language filter ① ^(cs he hu)\$ Regular expression u Source language ① English		`
Language filter () ^(cs he hu)\$ Regular expression u Source language () English Language used for sc		`

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

 Image: The Diango admin interface
 Image: Project configuration

 Image: Component configuration
 Image: Component configuration

Project configuration

??:

/devel/integration

These basic attributes set up and inform translators of a project:

????????

Verbose project name, used to display the project name.

Project name suitable for URLs.

???????? Web ????

URL where translators can find more info about the project.

????????

Mailing list where translators can discuss or comment translations.

??????????

URL to more site with more detailed instructions for translators.

Set Language-Team header

Whether Weblate should manage the Language-Team header (this is a GNU gettext only feature right now).

Whether to use shared translation memory, see 2222222 for more details.

Whether to contribute to shared translation memory, see **2222222** for more details.

???????

Configure per project access control, see <u>2222222222222</u> for more details. Default value can be changed by *DEFAULT_ACCESS_CONTROL*.

??????

Enable review workflow for translations, see 2222222.

??????????

Enable review workflow for source strings, see 22222.

???????

Whether unauthenticated **2222** are to be used for this repository.

??:

?????

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.

??:

Parsing language codes

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.

663

/devel/integration

You can find some examples of typical configurations in the Supported file formats.

12: 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 repository used to pull changes.

??:

See 222222222 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* [2][2] 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 <u>22222222</u> 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 *Template markup*.

For example on GitHub, use something like: https://github.com/WeblateOrg/hello/blob/
{{branch}}/{{filename}}#L{{line}}

In case your paths are relative to different folder, you might want to strip leading directory by parentdir filter (see *Template markup*): https://github.com/WeblateOrg/hello/blob/{{branch}}/ {{filename|parentdir}}#L{{line}}

????????????? URL

??????????

Which branch to checkout from the VCS, and where to look for translations.

????? push

Branch for pushing changes, leave empty to use 222222222.

This is currently only supported for Git, GitLab and GitHub, it is ignored for other VCS integrations.

File mask

Mask of files to translate, including path. It should include one "*" replacing language code (see *Language definitions* for info on how this is processed). In case your repository contains more than one translation file (e.g. more gettext domains), you need to create a component for each of them.

For example po/*.po or locale/*/LC_MESSAGES/django.po.

In case your filename contains special characters such as [,], these need to be escaped as [[] or []].

??:

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

Base file containing string definitions for Monolingual components.

??:

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

??????????

Whether to allow editing the base file for Monolingual components.

?????????

Intermediate language file for *Monolingual components*. In most cases this is a translation file provided by developers and is used when creating actual source strings.

??:

Base file used to generate new translations, e.g. .pot file with gettext.

Imany 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.

??:

???????

Translation file format, see also Supported file formats.

Email address used for reporting upstream bugs. This address will also receive notification about any source string comments made in Weblate.

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*.

???????

Whether translation suggestions are accepted for this component.

??????

Turns on vote casting for suggestions, see 2222222.

????????

Automatically accept voted suggestions, see 222222.

????

?????

List of checks which can not be ignored, see ??????.

?????????

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.

??:

adding-translation.

????????

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 *Template markup*.

???????

Name of the committer used for Weblate commits, the author will always be the real translator. On some VCSs this might be not supported.

Default value can be changed by *DEFAULT_COMMITER_NAME*.

Email of committer used for Weblate commits, the author will always be the real translator. On some VCSs this might be not supported. The default value can be changed in *DEFAULT_COMMITER_EMAIL*.

Whether committed changes should be automatically pushed to the upstream repository. When enabled, the push is initiated once Weblate commits changes to its internal repository (see *Lazy commits*). To actually enable pushing *Repository push URL* has to be configured as well.

Sets how old changes (in hours) are to get before they are committed by background task or *commit_pending* management command. All changes in a component are committed once there is at least one older than this period.

Default value can be changed by *COMMIT_PENDING_HOURS*.

?????????

Enables locking the component on repository error (failed pull, push or merge). Locking in this situation avoids adding another conflict which would have to be resolved manually.

The component will be automatically unlocked once there are no repository errors left.

?????

Language used for source strings. Change this if you are translating from something else than English.

In case you are translating bilingual files from English, but want to be able to do fixes in the English translation as well, you might want to choose *English (Developer)* as a source language to avoid conflict between name of the source language and existing translation.

????????

Regular expression used to filter the translation when scanning for filemask. This can be used to limit the list of languages managed by Weblate.

??: You need to list language codes as they appear in the filename.

Some examples of filtering:

Filter description?????Selected languages only^(cs|de|es)\$Exclude languages^(?!(it|fr)\$).+\$Filter two letter codes only^..\$Exclude non language files ^(?!(blank)\$).+\$Include all files (default)^[^.]+\$

????????

Regular expression used to determine the variants of a string, see variants.

? Most of the fields can be edited by project owners or managers, in the Weblate interface.

??:

Does Weblate support other VCSes than Git and Mercurial? alerts

????

Restricted access

By default the component is visible to anybody who has access to the project, even if the person can not perform any changes in the component. This makes it easier to keep translation consistency within the project.

Enable this in case you want to grant access to this component explicitly - the project level permissions will not apply and you will have to specify component or component list level permission in order to grant access.

Default value can be changed by *DEFAULT_RESTRICTED_COMPONENT*.

This applies to project managers as well - please make sure you will not loose access to the component after toggling the status.

Template markup

Currently it is used in:

Commit message formatting, see Component configuration

Several addons

Executing scripts from addon There following variables are available in the component templates: ????? ????????????? Translation URL ????????Translation stats, this has further attributes, examples below. Total strings count Count of strings needing review Percent of strings needing review Translated strings count Translated strings percent Number of strings with failing checks Percent of strings with failing checks Author of current commit, available only in the commit scope. Name of currently executed addon, available only in the addon commit message. The following variables are available in the repository browser or editor templates: current branch line in file filename, you can also strip leading parts using the parentdir filter, for example { {filename | parentdir } } You can combine them with filters:

{{ component|title }}

You can use conditions:

{% if stats.translated_percent > 80 %}Well translated!{% endif %}

There is additional tag available for replacing characters:

{% replace component "-" " " %}

You can combine it with filters:

{% replace component | capfirst "-" " %}

There are also additional filter to manipulate with filenames:

```
Directory of a file: {{ filename|dirname }}
File without extension: {{ filename|stripext }}
File in parent dir: {{ filename|parentdir }}
It can be used multiple times: {{ filename|parentdir|parentdir }}
```

...and other Django template features.

Importing speed

Fetching VCS repository and importing translations to Weblate can be a lengthy process, depending on size of your translations. Here are some tips:

Optimize configuration

The default configuration is useful for testing and debugging Weblate, while for a production setup, you should do some adjustments. Many of them have quite a big impact on performance. Please check *Production setup* for more details, especially:

Configure Celery for executing background tasks (see *Background tasks using Celery*)

Enable caching Use a powerful database engine Disable debug mode

Check resource limits

If you are importing huge translations or repositories, you might be hit by resource limitations of your server.

Check the amount of free memory, having translation files cached by the operating system will greatly improve performance.

Disk operations might be bottleneck if there is a lot of strings to process—the disk is pushed by both Weblate and the database.

Additional CPU cores might help improve performance of background tasks (see Background tasks using Celery).

Disable unneeded checks

Some quality checks can be quite expensive, and if not needed, can save you some time during import if omitted. See *CHECK_LIST* for info on configuration.

Automatic creation of components

In case your project has dozen of translation files (e.g. for different gettext domains, or parts of Android apps), you might want to import them automatically. This can either be achieved from the command line by using *import_project* or *import_json*, or by installing the [2022/2022/2022] addon.

To use the addon, you first need to create a component for one translation file (choose the one that is the least likely to be renamed or removed in future), and install the addon on this component.

For the management commands, you need to create a project which will contain all components and then run *import_project* or *import_json*.

??:

Language definitions

To present different translations properly, info about language name, text direction, plural definitions and language code is needed.

Parsing language codes

While parsing translations, Weblate attempts to map language code (usually the ISO 639-1 one) to any existing language object.

You can further adjust this mapping at project level by **2222**.

If no exact match can be found, an attempt will be made to best fit it into an existing language (e.g. ignoring the default country code for a given language—choosing cs instead of cs_CZ).

Should that also fail, a new language definition will be created using the defaults (left to right text direction, one plural). The automatically created language with code xx_Xx will be named as xx_XX (generated). You might want to change this in the admin interface later, (see *Changing language definitions*) and report it to the issue tracker (see *Weblate* (2020)), so that the proper definition can be added to the upcoming Weblate release.

122: In case you see something unwanted as a language, you might want to adjust **222**22222 to ignore such file when parsing translations.

??:

???????Adding new translations

Changing language definitions

You can change language definitions in the languages interface (/languages/ URL).

While editing, make sure all fields are correct (especially plurals and text direction), otherwise translators will be unable to properly edit those translations.

????????

Definitions for more than 550 languages are included in Weblate and the list is extended in every release. Whenever Weblate is upgraded (more specifically whenever weblate migrate is executed, see *Generic upgrade instructions*) the database of languages is updated to include all language definitions shipped in Weblate.

This feature can be disable using *UPDATE_LANGUAGES*. You can also enforce updating the database to match Weblate built-in data using *setuplang*.

In many cases it is not a good idea to use macro language code for a translation. The typical problematic case might be Kurdish language, which might be written in Arabic or Latin script, depending on actual variant. To get correct behavior in Weblate, it is recommended to use individual language codes only and avoid macro languages.

??:

Macrolanguages definition, List of macrolanguages

Language definitions

Each language consists of following fields:

?????

Code identifying the language. Weblate prefers two letter codes as defined by ISO 639-1, but uses ISO 639-2 or ISO 639-3 codes for languages that do not have two letter code. It can also support extended codes as defined by BCP 47.

??:

Parsing language codes? Adding new translations

???

Visible name of the language. The language names included in Weblate are also being localized depending on user interface language.

???????

Determines whether language is written right to left or left to right. This property is autodetected correctly for most of the languages.

Plural number

Number of plurals used in the language.

??????

Gettext compatible plural formula used to determine which plural form is used for given count.

222, GNU gettext utilities: Plural forms, Language Plural Rules by the Unicode Consortium

Adding new translations

2222 2.18 222: In versions prior to 2.18 the behaviour of adding new translations was file format specific.

Weblate can automatically start new translation for all of the file formats.

Some formats expect to start with an empty file and only translated strings to be included (for example *Android string resources*), while others expect to have all keys present (for example *GNU gettext*). In some situations this really doesn't depend on the format, but rather on the framework you use to handle the translation (for example with *JSON files*).

When you specify 2222222222 in *Component configuration*, Weblate will use this file to start new translations. Any exiting translations will be removed from the file when doing so.

When <u>2222222222222</u> is empty and the file format supports it, an empty file is created where new strings will be added once they are translated.

The 22222 2222 allows you to customize language code used in generated filenames:

Dependent on file format, for most of them POSIX is used.

Typically used by gettext and related tools, produces language codes like pt_BR.

POSIX style language code including the country code even when not necessary (for example cs_CZ).

Typically used on web platforms, produces language codes like pt-BR.

BCP style language code including the country code even when not necessary (for example cs-CZ).

Only used in Android apps, produces language codes like pt-rBR.

Used by Java—mostly BCP with legacy codes for Chinese.

Additionally, any mappings defined in **2222** are applied in reverse.

Weblate recognizes any of these when parsing translation files, the above settings only influences how new files are created.

??:

??????Parsing language codes

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:

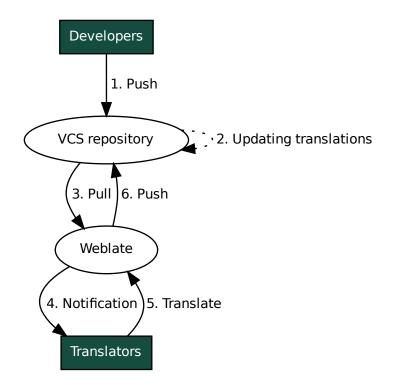
1.Developers make changes and push them to the VCS repository.

- 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*).



Updating repositories

Avoiding merge conflicts

The merge conflicts from Weblate arise when same file was changed both in Weblate and outside it. There are two approaches to deal with that - avoid edits outside Weblate or integrate Weblate into your updating process, so that it flushes changes prior to updating the files outside Weblate.

The first approach is easy with monolingual files - you can add new strings within Weblate and leave whole editing of the files there. For bilingual files, there is usually some kind of message extraction process to generate translatable files from the source code. In some cases this can be split into two parts - one for the extraction generates template (for example gettext POT is generated using **xgettext**) and then further process merges it into actual translations (the gettext PO files are updated using **msgmerge**). You can perform the second step within Weblate and it will make sure that all pending changes are included prior to this operation.

The second approach can be achieved by using API to force Weblate to push all pending changes and lock the translation while you are doing changes on your side.

The script for doing updates can look like this:

```
# Lock Weblate translation
wlc lock
# Push changes from Weblate to upstream repository
```

(?????????)

(???????????)

```
wlc push
# Pull changes from upstream repository to your local copy
git pull
# Update translation files, this example is for Django
./manage.py makemessages --keep-pot -a
git commit -m 'Locale updates' -- locale
# Push changes to upstream repository
git push
# Tell Weblate to pull changes (not needed if Weblate follows your repo
# automatically)
wlc pull
# Unlock translations
wlc unlock
```

If you have multiple components sharing same repository, you need to lock them all separately:

wlc lock foo/bar wlc lock foo/baz wlc lock foo/baj

The example uses *Weblate* ?????????, which needs configuration (API keys) to be able to control Weblate remotely. You can also achieve this using any HTTP client instead of wlc, e.g. curl, see API.

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:

📮 WeblateOrg / hello	O Unwatch ▼ 2 ★ Unstar 7 ÿ Fork 20
↔ Code (1) Pull requests 0	III Projects Insights Settings
Options	Webhooks / Add webhook
Collaborators & teams	We'll send a PosT request to the URL below with details of any subscribed events. You can also specify which
Branches	data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.
Webhooks	Payload URL *
Integrations & services	https://hosted.weblate.org/hooks/github/
Deploy keys	Content type
Alerts	application/x-www-form-urlencoded \$
	Secret
	By default, we verify SSL certificates when delivering payloads.
	Which events would you like to trigger this webhook?
	● Just the push event.
	Send me everything.
	O Let me select individual events.
	We will deliver event details when this hook is triggered.
	Add webhook

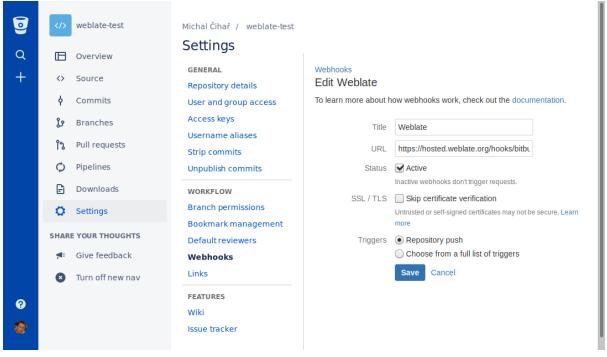
For the payload URL, append /hooks/github/ to your Weblate URL, for example for the Hosted Weblate service, this is https://hosted.weblate.org/hooks/github/.

You can leave other values at default settings (Weblate can handle both content types and consumes just the *push* event).

??:

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/).

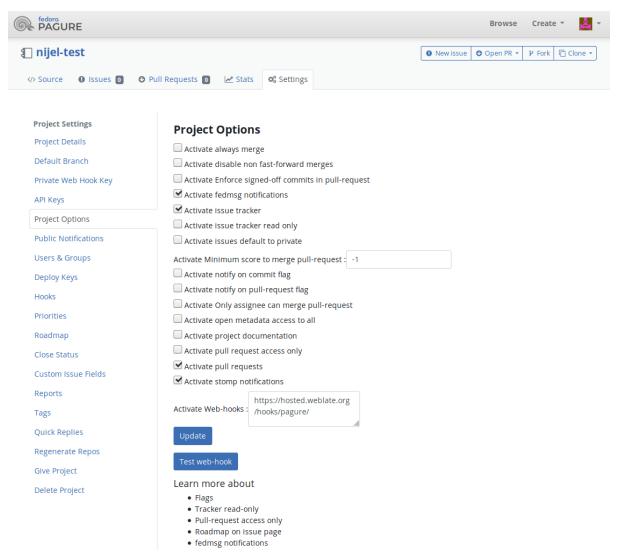


??:

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/).

????? 3.3 ????.



??:

Automatically receiving changes from Azure Repos

22222 3.8 222.

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*.

??:

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*.

??:

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 2222222

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

The push options differ based on the <u>22222222</u> used, more details are found in that chapter.

Overall, following options are available with Git, GitHub and GitLab:

Desired setup	<u>????????????</u>	<u>???????????</u> UR	L <u>??????</u> push
No push	Git	empty	empty
Push directly	Git	SSHURL	empty
???????? push	Git	SSH URL	Branch name
GitHub pull request from fork	GitHub	empty	empty
GitHub pull request from branch	GitHub	$SSH URL^1$	Branch name
GitLab merge request from fork	GitLab	empty	empty
GitLab merge request from branc	hGitLab	SSH URL ¹	Branch name
????????? Pagure ???? ???????	Pagure	empty	empty
??????????????????????????????????????	Pagure	SSH URL ¹	Branch name

22: You can also enable automatic pushing of changes after Weblate commits, this can be done in 22/22/22/22/22/22.

??:

See 22/22/22/22/22 for setting up SSH keys, and *Lazy commits* for info about when Weblate decides to commit changes.

Can be empty in case 2222222222222 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.

Q Search for people or teams

Weblate push user

People and teams that can dismiss reviews.

Organization and repository administrators These members can always dismiss.

V weblate

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.

API

Lazy commits

The behaviour of Weblate is to group commits from the same author into one commit if possible. This greatly reduces the number of commits, however you might need to explicitly tell it to do the commits in case you want to get the VCS repository in sync, e.g. for merge (this is by default allowed for the *Managers* group, see <u>2222222</u>).

The changes in this mode are committed once any of the following conditions are fulfilled:

Somebody else changes an already changed string.

A merge from upstream occurs.

An explicit commit is requested.

If you want to commit changes more frequently and without checking of age, you can schedule a regular task to perform a commit:

```
CELERY_BEAT_SCHEDULE = {
    # Unconditionally commit all changes every 2 minutes
    "commit": {
        "task": "weblate.trans.tasks.commit_pending",
        # Ommiting hours will honor per component settings,
        # otherwise components with no changes older than this
        # won't be committed
        "kwargs": {"hours": 0},
        # How frequently to execute the job in seconds
        "schedule": 120,
    }
}
```

Processing repository with scripts

The way to customize how Weblate interacts with the repository is 2222. Consult *Executing scripts from addon* for info on how to execute external scripts through addons.

Keeping translations same across components

Once you have multiple translation components, you might want to ensure that the same strings have same translation. This can be achieved at several levels.

Translation propagation

With translation propagation enabled (what is the default, see *Component configuration*), all new translations are automatically done in all components with matching strings. Such translations are properly credited to currently translating user in all components.

The translation propagation requires the key to be match for monolingual translation formats, so keep that in mind when creating translation keys.

Consistency check

The <u>[2][2][2][2]</u> 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 2222) or make it run automatically on repository update using addon (see 2222).

Licensing translations

You can specify which license translations are contributed under. This is especially important to do if translations are open to the public, to stipulate what they can be used for.

You should specify *Component configuration* license info. You should avoid requiring a contributor license agreement, though it is possible.

License info

Upon specifying license info (license name and URL), this info is shown in the translation info section of the respective *Component configuration*.

Usually this is best place to post licensing info if no explicit consent is required. If your project or translation is not libre you most probably need prior consent.

????????

If you specify a contributor license agreement, only users who have agreed to it will be able to contribute. This is a clearly visible step when accessing the translation:

Ø ₩€	eblate 🛛	Dashboard	Projec		anguages 🗸	Checks					٩	+ Add -	⊕
🔗 Web	blateOrg / l	Language r	names										translated
Contri	ibution to th	nis translat	ion require	es you to a	agree with a	contributo	r agreeme	ent.			Viewo	ontributor a	agreement
Transl	lations	Info A	Alerts	Search	Glossari	es Ins	ights •	Files 🕶	Tools 🗸	Manage -	Share -		
	Language			Transl	ated	Untrans	lated	Untra	anslated words	Checks	Sugg	estions	Comments
/	Czech 坐 🚹	GPL-3.0			~								
	Hebrew 坐 🛽	GPL-3.0			~								
/	Hungarian 🖞	GPL-3.	D		81%		4			5			
	English 🛥 🌮	9 📒 GPL-3.0			~								

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

The entered text is formatted into paragraphs and external links can be included. HTML markup can not be used.

User licenses

Any user can review all translation licenses of all public projects on the instance from their profile:

() Weblate Dashboard Projects → Languages → Checks →	🔧 🕂 Add - 🌍 - 🚥
Your profile	
Languages Preferences Notifications Account Profile Licen	ses Audit log API access
Licenses	
Please pay attention to the licensing info, as this specifies how translations can be use By registering you agree to use your name and e-mail in the commits, and provide your You have agreed to the following as a contributor: • WeblateOrg/Language names	
Licenses for individual translations	
GNU General Public License v3.0 or later GPL30 3 WeblateOrg/Djangojs WeblateOrg/Django WeblateOrg/Language names MIT License III 3 WeblateOrg/Android	

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

?????

??????

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).

You can combine these with **22222** into one of the following setups:

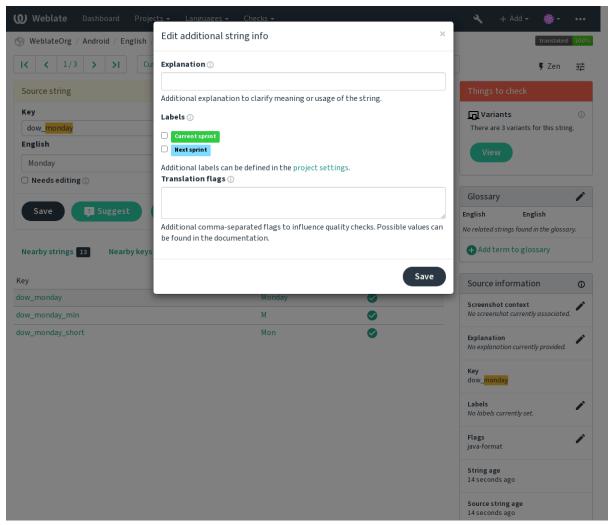
Users suggest and vote for suggestions and a limited group controls what is accepted. - Turn on voting. - Turn off automatic acceptance. - Don't let users save translations.

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.

W Weblate	Dashboard Projects - Languages - Che	cks 🗸		عر	+ Add -	• -	••
🛞 WeblateOrg	g / Django / Czech / Translate					translate	ed 9
K K 11	L/26 > > All strings -	Position and pri	ority -	-1		🖡 Zen	
Translation			e	Glossa	rv		
Explanation				English	Czech		
	r automatic translation tool			machine translation	strojov n překlad	Web	lateOr
English				project	projekt	Web	ateOr
	ranslation via machine translation uses active machine : and applies them in this project.	translation engines to get the best possible	Ū	🕂 Add	term to glo:	sary	
Czech		Clone source		Source	informatio		0
	ý překlad prostřednictvím strojového překladu používá 1ožných překladů a použije je na tento projekt.	aktivní enginy strojového překladu pro získání			hot context		
Needs edit		169	/1570		nshot current	r associated	1.
				Explana			/
Save	Suggest Skip			Help tex	t for automati	c translatio	n toc
Nearby string	s 26 Comments Automatic suggestions	Other languages History		Labels	s currently set.		/
Nearby string	S 20 Automatic suggestions	other tanguages history			s currently set.		
Context	English	Czech	State	Flags No flags	currently set.		/
	Files Automatic translation	Soubory Automatický překlad			tringlocatio		
	Automatic translation Add new translation string	Automaticky preklad Add new translation string	<u> </u>	weblate	/templates/tr	anslation.h1	:ml:2.
	-	-	A	String a 2 second			
	Translation status	Stav překladu	O	Sources	string age		
	%(count)s word Other components	%(count)s slovo Další součásti	 ✓ ✓ 	3 secon			
	Translation file	Soubor s překladem	0	Transla weblate	tion file /locale/cs/LC	MESSAGES	/dia
	Download	Stáhnout		go.po, s			
	Browse all translation changes	Procházet všechny změny v překladu.					
	Automatic translation takes existing translations in	Automatický překlad použije stávající					
	this project and applies them to the current component. It can be used to push translations to a	překlady v projektu na tuto součást. Může být užitečný pro sloučení překladů z jiné	-				
	different branch, to fix inconsistent translations or to translate a new component using translation	větve, opravu nekonzistentních překladů nebo překlad nové součásti pomocí					
	memory.	překladové paměti.					
	Automatic translation via machine translation uses active machine translation engines to get the best	Automatický překlad prostřednictvím strojového překladu používá aktivní enginy	0				
	possible translations and applies them in this project.	strojového překladu pro získání nejlepších možných překladů a použije je na tento					
	You can add new translation string bara, it will	projekt. Zda můžata přidat pový řatězas k překladu					
	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.	\checkmark				
	The uploaded file will be merged with the current translation. In case you want to overwrite already translated strings, don't forget to enable it.	Nahraný soubor bude sloučen se stávajícími překlady. Pokud chcete přepsat již přeložené řetězce, nezapomeňte to povolit.	0				
	The uploaded file will be merged with the current translation.	Nahraný soubor bude sloučen se stávajícími překlady.	0				
	The fulltext search might not work properly as the fulltext index for this translation is not yet up to	Fulltextové vyhledávání nemusí fungovat správě, protože fulltextový index pro tento	0				
	date. Review	překlad ještě není plně zpracován. Kontrola					
	Review translations touched by other users.	Zkontrolovat překlady od ostatních	S				
	Start review	uživatelů. Začít kontrolu					
	Percent	Procenta	0				
	Total	Celkem	\checkmark				
	Failing check	Neúspěšných kontrol	0				
	Last activity Last change	Poslední aktivita Poslední změna	 ⊘ 				
	Last author	Poslední autor	0				
Question for a mathematics- based	What is %s?	Kolik to je?					
CAPTCHA, the %s is an arithmetic problem							
	The string uses three dots () instead of an ellipsis character (\ldots)		D				

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

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 ????.

[2][2][2] 3.3 [2][2]: 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.

??:

2222

??

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:

Weblate Dashi	board Projects -	Languages - Chec	ks -				٩	+ Add -	•	•••
💮 WeblateOrg / Djang	go / Czech / Transla	te							translat	ed <mark>96%</mark>
IK K 11/26	> >I All strin	ngs 🔻		Position and pr	iority -	ΞĮ			🖡 Zen	幸
Translation					Θ		Glossa	ry		
Explanation						E	nglish	Czech		
Help text for automa	tic translation tool						nachine ranslation	strojový 1 překlad		lateOrg
English						р	roject	projekt	Web	lateOrg
	n via machine translat lies them in this projec		translation engines to ge	t the best possible	Ū		🕂 Add t	term to glos	sary	
Czech			🕒 Clone source 🛤				Source	informatio	on	0
, ,	prostřednictvím strojo překladů a použije je na		aktivní enginy strojového	překladu pro získán	Í		Screenst	hot context		
Needs editing i				16	9/1570		Searce string scio, end by			
Save 🔅 Save		Skip tomatic suggestions	Other languages	History			Explanat	anna ana tseografe. Anna Anna Garant for a filineal acada athig	c translatic	n tool
Translation memory		Search								
Translation	Source	Origin	Similarity				Labels No labels	currently set.		
Automatický překlad prostřednictvím strojového překladu	Automatic translation via machine	Weblate (<u>WeblateOrg/Django</u>) Weblate Translation	Сору	Copy and s	ave		Flags No flags (currently set.		/
používá aktivní enginy strojového překladu pro získání nejlepších	translation uses active machine translation	Memory (Project: weblateorg/django) Weblate Translation						t ring locatio /templates/tr		tml:212
možných překladů a použije je na tento projekt.	engines to get the best possible translations and	Memory (Shared: weblateorg/django)					String as 10 secon			
	applies them in this project.						Source s 10 secon	t ring age Ids ago		
							Translat weblate/ go.po, st	locale/cs/LC_	MESSAGE	6/djan

Powered by Weblate 4.4 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:

	Dashboard Projects - Lang	guages 👻 Checks 👻		🔧 🕂 Add 🕶 🍈 🕶 📢			
👌 WeblateOrg /	Django / Screenshots / Automa	tictranslation					
Screenshot has be	een uploaded, you can now assign it	: to source strings.					
Assigned source	strings						
ource string	Context	Location	Assigned screenshots	Actions			
lo source strings a	are currently assigned!						
Screenshot is sho	wn to add visual context for all list	ed source strings.					
Assign source st	rings						
ource string	Context	Location	Assigned screenshots	Actions			
lo new matching s	source strings found.						
Source string sea	rch Search			Automatically recognize			
Image							
	Source string						
	Hello, world!d						
	One						
	Orangutan has %d banana.∉	1					
	Other Orangutan has %d bananas.	.					
	Try Weblate at <http: demo<="" td=""><td>.weblate.org/>!↩</td><td></td><td></td></http:>	.weblate.org/>!↩					
	Thank you for using Weblate.						
	Screenshot is shown to add v	isual context for all listed	source strings.				
Edit screenshot							
Screenshot name	e						
Automatic trans	slation						
Change: Choose File No	shots/screenshot.png o file chosen NG images up to 2000x2000 pixels.						
Upload JPEG or P							
Upload JPEG or P							
Save	ils						
Save Screenshot deta	ils		now				
Save Screenshot deta reated	ils		now testuser				
Save Screenshot deta created Uploaded by	ils		40b				
			testuser				
Save Screenshot deta irreated Iploaded by anguage Delete screensho		ed source strings.	testuser				

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to 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/>. # 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 ??<u>???????????????</u>AUTOFIX_LIST Python

??????????

placeholders:"special:value":"other value", regex:.*

22222 Markdown 22222222

UNIX 22222 DOS 222222222 \n 22222 \r\n22

2222 HTML 22222222222222 22222 HTML?

22222 XML 2222222222222222 XML 22 222 XML 22222222 []: replacements:%s:"John Doe" [] 2JavaScript 222222222222 2Python 22222222222222

2XML 222222222222

?????

?????? 3.11 ????.

???????

?????? 3.7 ????.

Weblate

Weblate font management tool in *Fonts* under the *Manage* menu of your translation project provides interface to upload and manage fonts. TrueType or OpenType fonts can be uploaded, set up font-groups and use those in the check.

@ Weblate Dashboard Projects - Languages -	• Checks •	ع	+ Add -	•••
🛞 WeblateOrg / Font groups / default-font				
Font group				
Name	default-font			
Default font	Source Sans Pro Bold			
Japanese language overri	Droid Sans Fallback Regular	Remo	ove	
Korean language overri	Droid Sans Fallback Regular	Remo	ove	
Delete				
Add language override				
Language				
				~
Font				
				~
Save				
Edit font group				
Font group name				
default-font				
Identifier you will use in checks to select this font group. Av	oid whitespaces and special characters.			
Default font				
Source Sans Pro Bold				~
Default font is used unless per language override matches.				
Save				

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

Weblate Dashboar	rd Projects -	Languages -	Checks -	ع	+ Add -	••	•••
🛞 WeblateOrg / Fonts							
Font groups Fonts							
Group name	Default font		Language overrides				
default-font	Source Sans Pro I	Bold	Japanese: Droid Sans Fallback Regular Korean: Droid Sans Fallback Regular		Edit		
Add font group							
Font group name							
-	ecks to select this fo	ont group. Avoid	vhitespaces and special characters.				
Default font							~
Default font is used unless	per language overrie	de matches.					
Save							

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

W Weblate	Dashboard	Projects 🗸	Languages 🗸	Checks -		ع	+ Add -	•••
💮 WeblateOrg	/ Fonts / Dro	id Sans Fallbac	k Regular					
Font								
Font family					Droid Sans Fallback			
Font style					Regular			
File size					3939852			
Created					now			
Uploaded by					testuser			
Used in groups								
Delete								

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

Weblate Dashboard Projects - Languages - Chec		٩	+ Add -	۰	•••
🛞 WeblateOrg / Fonts					
Font groups Fonts					
Font family	Font style				
Droid Sans Fallback	Regular Edit				
Source Sans Pro	Bold				
Add font					
Font file Choose File No file chosen OpenType and TrueType fonts are supported. Upload					

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to 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/>.

(???????????)

```
"""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
```

```
# 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
```

(?????????)

(???????????)

```
def check_target_unit(self, sources, targets, unit):
    if self.is_language(unit, ("cs",)):
        return targets[1] == targets[2]
    return False

def check_single(self, source, target, unit):
    """We don't check target strings here."""
    return False
```

????

Built-in support for several machine translation services and can be turned on by the administrator using $MT_SERVICES$ for each one. They come subject to their terms of use, so ensure you are allowed to use them how you want.

The source language can be configured at Project configuration.

amaGama

Special installation of *tmserver* run by the authors of Virtaal.

Turn on this service by adding weblate.machinery.tmserver.AmagamaTranslation to ${\it MT_SERVICES}.$

??:

Installing amaGama? Amagama? amaGama Translation Memory

Apertium

A libre software machine translation platform providing translations to a limited set of languages.

The recommended way to use Apertium is to run your own Apertium-APy server.

Turn on this service by adding weblate.machinery.apertium.ApertiumAPYTranslation to *MT_SERVICES* and set *MT_APERTIUM_APY*.

??:

MT_APERTIUM_APY, Apertium website, Apertium APy documentation

AWS

?????? 3.1 ????.

Amazon Translate is a neural machine translation service for translating text to and from English across a breadth of supported languages.

1. Turn on this service by adding weblate.machinery.aws.AWSTranslation to *MT_SERVICES*.

2.Install the *boto3* module.

3.Configure Weblate.

??:

MT_AWS_REGION^[]*MT_AWS_ACCESS_KEY_ID*^[]*MT_AWS_SECRET_ACCESS_KEY* Amazon Translate Documentation

Baidu API machine translation

?????? 3.2 ????.

Machine translation service provided by Baidu.

This service uses an API and you need to obtain an ID and API key from Baidu to use it.

Turn on this service by adding weblate.machinery.baidu.BaiduTranslation to *MT_SERVICES* and set *MT_BAIDU_ID* and *MT_BAIDU_SECRET*.

??:

MT_BAIDU_ID?MT_BAIDU_SECRET Baidu Translate API

DeepL

?????? 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*.

The set of the 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

Machine translation service provided by Google.

This service uses the Google Translation API, and you need to obtain an API key and turn on billing in the Google API console.

To turn on this service, add weblate.machinery.google.GoogleTranslation to *MT_SERVICES* and set *MT_GOOGLE_KEY*.

??:

*MT_GOOGLE_KEY***?**Google translate documentation

Google Translate API V3 (Advanced)

Machine translation service provided by Google Cloud services.

This service differs from the former one in how it authenticates. To enable service, add weblate.machinery. 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*.

??:

 $\label{eq:mt_google_credentials} \ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{PROJECT}}\ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{LOCATION}}\ensuremath{\underline{GOOgle}}\ensuremath{\underline{CREDENTIALS}}\ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{PROJECT}}\ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{LOCATION}}\ensuremath{\underline{GOOgle}}\ensuremath{\underline{CREDENTIALS}}\ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{LOCATION}}\ensuremath{\underline{GOOgle}}\ensuremath{\underline{CREDENTIALS}}\ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{LOCATION}}\ensuremath{\underline{GOOgle}}\ensuremath{\underline{CREDENTIALS}\ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{LOCATION}}\ensuremath{\underline{GOOgle}}\ensuremath{\underline{CREDENTIALS}\ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{CREDENTIALS}\ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{LOCATION}}\ensuremath{\underline{GOOgle}}\ensuremath{\underline{CREDENTIALS}\ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{CREDENTIALS}\ensuremath{\underline{MT}}\ensuremath{\underline{GOOGLE}}\ensuremath{\underline{CREDENTIALS}\ensuremath{\underline{MT}}\ens$

Microsoft Cognitive Services Translator

?????? 2.10 ????.

Machine translation service provided by Microsoft in Azure portal as a one of Cognitive Services.

Weblate implements Translator API V3.

To enable this service, add weblate.machinery.microsoft.MicrosoftCognitiveTranslation to *MT_SERVICES* and set *MT_MICROSOFT_COGNITIVE_KEY*.

Translator Text API V2

The key you use with Translator API V2 can be used with API 3.

Translator Text API V3

You need to register at Azure portal and use the key you obtain there. With new Azure keys, you also need to set *MT_MICROSOFT_REGION* to locale of your service.

??:

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 allows you to programmatically access the terminology, definitions and user interface (UI) strings available in the Language Portal through a web service.

Turn this service on by adding weblate.machinery.microsoftterminology. MicrosoftTerminologyService to MT_SERVICES.

??:

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 ????.

Machine translation service provided by Netease.

This service uses an API, and you need to obtain key and secret from NetEase.

Turn on this service by adding weblate.machinery.youdao.NeteaseSightTranslation to MT_SERVICES and set MT_NETEASE_KEY and MT_NETEASE_SECRET.

??:

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 ?? Installing amaGama? Amagama ?? Amagama Translation Memory

Yandex Translate

Machine translation service provided by Yandex.

This service uses a Translation API, and you need to obtain an API key from Yandex.

Turn on this service by adding weblate.machinery.yandex.YandexTranslation to *MT_SERVICES*, and set *MT_YANDEX_KEY*.

??:

MT_YANDEX_KEY ?? Yandex Translate API ?? Powered by Yandex. Translate

Youdao Zhiyun API machine translation

?????? 3.2 ????.

Machine translation service provided by Youdao.

This service uses an API, and you need to obtain an ID and an API key from Youdao.

Turn on this service by adding weblate.machinery.youdao.YoudaoTranslation to *MT_SERVICES* and set *MT_YOUDAO_ID* and *MT_YOUDAO_SECRET*.

??:

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 $\it MT_SERVICES.$

Weblate Translation Memory

?????? 2.20 ????.

The 222222 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

Machine translation service provided by SAP.

You need to have a SAP account (and enabled the SAP Translation Hub in the SAP Cloud Platform) to use this service.

Turn on this service by adding weblate.machinery.saptranslationhub.SAPTranslationHub to *MT_SERVICES* and set the appropriate access to either sandbox or the productive API.

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*.

??:

```
MT_SAP_BASE_URL<sup>®</sup>MT_SAP_SANDBOX_APIKEY<sup>®</sup>MT_SAP_USERNAME<sup>®</sup>MT_SAP_PASSWORD<sup>®</sup>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
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
\hookrightarrow": text}
```

You can list own class in MT_SERVICES and Weblate will start using that.

????

?????? 2.19 ????.

₩eblate Dashboard Projects - Languages - Checks -	ચ	+ Add •	. -	•••
SeblateOrg / Language names / Addons				
Installed addons				0
There are no addons currently installed.				
Available addons				Û
$oldsymbol{ar{\lambda}}_{oldsymbol{A}}$ Automatic translation \odot		nstall		
ズA Language consistency ⊕ project w	ide I	nstall		
Q Component discovery () repository w	ide I	nstall		
🗱 Bulk edit 🛈		nstall		
I Statistics generator ①		nstall		
🔹 Contributors in comment 🛈		nstall		
Customize gettext output 🕜		nstall		
🖨 Generate MO files 🛈		nstall		
Update PO files to match POT (msgmerge) ①		nstall		
Squash Git commits 🛈 repository w	ide I	nstall		
Stale comment removal 🛈 project w	ide I	nstall		
Stale suggestion removal () project w	ide I	nstall		
Some addons will ask for additional configuration during installation.				

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

???????

????

 22222
 3.9
 222

 22222
 22222
 22222

 22222
 22222
 22222

 22222
 22222
 22222

 22222
 22222
 22222

 222
 22222
 22222

 222
 22222
 22222

 222
 2222
 2222

 22
 2222
 2222

 22
 2222
 2222

 22
 2222
 2222

 22
 222
 222

 23
 4
 4

JavaScript ???? CDN

?????? 4.2 ????.

cdn-addon-config?weblate-cdn?cdn-addon-extract?cdn-addon-html

?????????

Does Weblate update translation files besides translations?

Does Weblate update translation files besides translations?

???????

???: 22222222222 *2222* 22222222

 Project
 <t

Configure addon			
Please review and confirm the	e matched components.		
Component Ma	tched files		
Following components would be o			
we	blate/locale/cs/LC_MESSA(blate/locale/hu/LC_MESSA(blate/locale/he/LC_MESSA(GES/djangojs.po (hu)	
we	blate/locale/hu/LC_MESSA(blate/locale/he/LC_MESSA(blate/locale/cs/LC_MESSA(GES/django.po (he)	
☐ I confirm the above matches lool	correct		
Regular expression to match trans	lation files against		
weblate/locale/(?P <language>[^/]*</language>)/LC_MESSAGES/(?P <compo< td=""><td>nent>[^/]*)\.po</td><td></td></compo<>	nent>[^/]*)\.po	
File format			
gettext PO file			
Customize the component name			
{{ component title }}			
Define the monolingual base filena	me		
Leave empty for bilingual translation	files.		
Define the base file for new transla	ations		
weblate/locale/{{ component }}.pot			
Filename of file used for creating new	translations. For gettext ch	oose .pot file.	
Language filter			
^(cs he hu)\$			
Regular expression to filter translation	on files against when scannin	ng for filemask.	
Clone addons from the main com	ponent to the newly create	ed ones	
Remove components for inexista	nt files		
The regular expression to match tran	slation files has to contain t	wo named groups to match component and la	nguage, some examples:
Regular expression		Example matched files	Description
(?P <language>[^/.]*)/(?P<compor< th=""><th>ment>[^/]*)\.po</th><th>cs/application.po cs/website.po de/application.po de/website.po</th><th>One folder per language containing translation files for components.</th></compor<></language>	ment>[^/]*)\.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_ [^/]*)\.po</language>	MESSAGES/(?P <component></component>	<pre>locale/cs/LC_MESSAGES/application.po locale/cs/LC_MESSAGES/website.po locale/de/LC_MESSAGES/application.po locale/de/LC_MESSAGES/website.po</pre>	Usual structure for storing gettext PO files.
	\.(?P <language></language>	<pre>src/locale/application.cs.po src/locale/website.cs.po</pre>	Using both component and language name within filename.
<pre>src/locale/(?P<component>[^/]*) [^/.]*)\.po</component></pre>		<pre>src/locale/application.de.po src/locale/website.de.po</pre>	
	<component>[^/]*)/(?</component>		Using language in both path and filename.
<pre>[^/.]*)\.po locale/(?P<language>[^/.]*)/(?f</language></pre>		<pre>src/locale/website.de.po locale/cs/application/cs.po locale/cs/website/cs.po locale/de/application/de.po</pre>	Using language in both path and filename. Android resource strings, split into several files.

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

??: *Template markup*

????

????? 3.11 ????.

2:

?????? NOT has:label

??:

2222

???????? "????" ????????

?????? 3.1 ????.

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

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

?????????

locale/{{ language_code }}.json

```
{
  "language": "{{ language_code }}",
  "strings": "{{ stats.all }}",
  "translated": "{{ stats.translated }}",
  "last_changed": "{{ stats.last_changed }}",
  "last_author": "{{ stats.last_author }}",
}
```

??:

Template markup


```
# 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 ?????

gettext ??????????

2222222 gettext 2222222222222222 ???????????

??: ???????gettext	?	77	22222222222222222222222222222222no-wrap
???????????????????????????????????????			

LINGUAS ???????

MO ???????

22222 PO 222222222 MO 22222222222

POT ?????? PO ??????? (msgmerge)

?????????

??:

Does Weblate update translation files besides translations?

Git ???????????

?????? 3.4 ????. ?????????22222 3.5 222. ????? ?????? 4.1 ????. ???????Co-authored-by:

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

222222222222223JSON 222222222222

??????????

????????

RESX ??????

?????? 3.9 ????.

??:

Does Weblate update translation files besides translations?

22222 3.10.2 222. 22222222222 YAML 2222222222222

Writing addon

You can write your own addons too, all you need to do is subclass weblate.addons.base.BaseAddon, define the addon metadata and implement a callback which will do the processing.

??:

Executing scripts from addon

Addons can also be used to execute external scripts. This used to be integrated in Weblate, but now you have to write some code to wrap your script with an addon.

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 descrption 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"

The script is executed with the current directory set to the root of the VCS repository for any given component.

Additionally, the following environment variables are available:

WL_VCS Version control system used.

WL_REPO Upstream repository URL.

WL_PATH Absolute path to VCS repository.

WL_BRANCH ?????? 2.11 ????.

Repository branch configured in the current component.

WL_FILEMASK Filemask for current component.

WL_TEMPLATE Filename of template for monolingual translations (can be empty).

WL_NEW_BASE ??????? 2.14 ????.

Filename of the file used for creating new translations (can be empty).

WL_FILE_FORMAT Fileformat used in current component.

WL_LANGUAGE Language of currently processed translation (not available for component level hooks).

WL_PREVIOUS_HEAD Previous HEAD on update (available only when running post update hook).

WL_COMPONENT_SLUG 2222 3.9 222.

Component slug used to construct URL.

WL PROJECT SLUG 22.2.2.2.2.3.9 2.2.2.

Project slug used to construct URL.

WL COMPONENT_NAME 222222 3.9 2222.

Component name.

WL_PROJECT_NAME 22222 3.9 2222.

Project name.

WL_COMPONENT_URL 22222 3.9 2222.

Component URL.

WL_ENGAGE_URL ??????? 3.9 ????.

Project engage URL.

??:

Component configuration

Post update repository processing

Post update repository processing can be used to update translation files when the VCS upstream source changes. To achieve this, please remember that Weblate only sees files committed to the VCS, so you need to commit changes as a part of the script.

For example with Gulp you can do it using following code:

```
#! /bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Pre commit processing of translations

Use the commit script to automatically make changes to the translation before it is committed to the repository. It is passed as a single parameter consisting of the filename of a current translation.

?????

?????? 2.20 ????.

Weblate comes with a built-in translation memory consisting of the following:
Manually imported translation memory (see *User interface*).
Automatically stored translations performed in Weblate (depending on *Translation memory scopes*).
Automatically imported past translations.
Content in the translation memory can be applied one of two ways:
Manually, 22222 view while translating.
Automatically, by translating strings using 22222, or 22222 addon.
For installation tips, see *Weblate Translation Memory*, which is turned on by default.

Translation memory scopes

[2][2][2][2] 3.2 [2][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.

????????

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.

???: Translation memory in JSON can be imported into Weblate, TMX is provided for interoperability with other tools.

??:

Weblate Dashboard Project	cts + Languages +	Checks -			+ Add -	•••
🛞 testuser / Translation memory						
Translation memory status						(i)
Number of your entries	0 Downlo	ad as JSON	Download as TMX	Delete		
Total number of entries	0					
Import translation memory						
File Choose File No file chosen You can upload a TMX or JSON file.						
Upload						

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

Management interface

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

??

22222222 settings.py 2222222Django 22222222

??:

AKISMET_API_KEY

ANONYMOUS_USER_NAME

AUDITLOG_EXPIRY

AUTH_LOCK_ATTEMPTS

AUTO_UPDATE

2222212t222 2222222222 222222222222

AVATAR_URL_PREFIX

 Display="block-system: 1pt space-system: 1pt spt space-system: 1pt space-system: 1pt space-system

AUTH_TOKEN_VALID

AUTH_PASSWORD_DAYS

??????? 180 ?????

AUTOFIX_LIST

```
AUTOFIX_LIST = (
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
)
```

??:

BASE_DIR

BASIC_LANGUAGES

?????? 4.4 ????.

??

BASIC_LANGUAGES = {"cs", "it", "ja", "en"}

??:

Language definitions

CSP_SCRIPT_SRC?CSP_IMG_SRC?CSP_CONNECT_SRC?CSP_STYLE_SRC?CSP_FONT_SRC

??

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

??:

CHECK_LIST

??: check 222222222222 Python 22222222222222222

 $CHECK_LIST = ()$

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```

??: 22227**?**222222222222

COMMENT_CLEANUP_DAYS

?????? 3.6 ????.

COMMIT_PENDING_HOURS

?????? 2.10 ????.

??:

DATA_DIR

sudo chown www-data:www-data -R \$DATA_DIR

222222 \$BASE_DIR/data 2222
BASE_DIR Backing up and moving Weblate

DATABASE_BACKUP

Backing up and moving Weblate

DEFAULT_ACCESS_CONTROL

DEFAULT_RESTRICTED_COMPONENT

DEFAULT_ADD_MESSAGE?DEFAULT_ADDON_MESSAGE?DEFAULT_COMMIT_MESSAGE?DEFAULT_DEI

DEFAULT_ADDONS

??

```
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

DEFAULT_COMMITER_NAME

DEFAULT_LANGUAGE

DEFAULT_MERGE_STYLE

DEFAULT_TRANSLATION_PROPAGATION

22222 2.5 222.
2222222222222 True 222
22:
Component configuration 222222222222222

DEFAULT_PULL_MESSAGE

ENABLE AVATARS

??:

ENABLE HOOKS

??:

[?][?][?][?][?]

ENABLE HTTPS

?????????????????HSTS -

The HTTPS redirection might be problematic in some cases and you might hit issue with infinite redirection in case you are using a reverse proxy doing SSL termination which does not correctly pass protocol headers to Django. Please tweak your reverse proxy configuration to emit X-Forwarded-Proto or Forwarded headers or configure SE-CURE_PROXY_SSL_HEADER to let Django correctly detect the SSL status.

??:

SESSION_COOKIE_SECURE^[]CSRF_COOKIE_SECURE^[]SECURE_SSL_REDIRECT^[]SECURE_PROXY_SSL_HEADER Set correct site domain

ENABLE SHARING

GITLAB_CREDENTIALS

?????? 4.3 ????.

222: Weblate 2022 GitLab 20222020202020202020202020 GitLab 202202020 GITLAB_USERNAME 202

```
GITLAB_CREDENTIALS = {
    "gitlab.com": {
        "username": "weblate",
        "token": "your-api-token",
    "gitlab.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

GITLAB_USERNAME

GITLAB_TOKEN

22222 4.3 222.
222222222 API 22222 GitLab 222222 2222
22: *GITLAB CREDENTIALS [GitLab*] GitLab: Personal access token

GITHUB_CREDENTIALS

22222 4.3 222. GitHub 2222222222222

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "github.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

GITHUB_USERNAME

GITHUB_TOKEN

GOOGLE_ANALYTICS_ID

HIDE_REPO_CREDENTIALS

HIDE_VERSION

?????? 4.3.1 ????.

IP_BEHIND_REVERSE_PROXY

?????? 2.14 ????.

??:

IP_PROXY_HEADER

?????? 2.14 ????.

??:

IP_PROXY_OFFSET

??????? 0 ????

??:

LEGAL_URL

?????? 3.5 ????.

Weblate ????????????????? URL?

??

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

 22222
 4.3
 2222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22222
 22

??

LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}

LICENSE_FILTER = set()

??:

alerts

LICENSE_REQUIRED

LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

??: ?????? True ????

LOCALIZE_CDN_URL ???? LOCALIZE_CDN_PATH

222: Hosted Weblate 222https://weblate-cdn.com/ 22222

??:

JavaScript ????? CDN

LOGIN_REQUIRED_URLS

22222222 URL 22222Weblate 2222222222222222

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

??:

REQUIRE_LOGIN

LOGIN_REQUIRED_URLS_EXCEPTIONS

```
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_URL

MATOMO_URL

???: 222222Matomo 22 22222222222222

??

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

??:

MATOMO_SITE_ID

MT_SERVICES

```
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.weblatetm.WeblateTranslation",
    "weblate.machinery.saptranslationhub.SAPTranslationHub",
    "weblate.memory.machine.WeblateMemory",
}
```

??:

<u>?????</u>?????

MT_APERTIUM_APY

Apertium-APy 22222 URL? https://wiki.apertium.org/wiki/Apertium-apy

MT_AWS_ACCESS_KEY_ID

MT_AWS_SECRET_ACCESS_KEY

22222222 API 2222 22: AWS2222222222

MT_AWS_REGION

MT_BAIDU_ID

MT_BAIDU_SECRET

MT_DEEPL_API_VERSION

?????? 4.1.1 ????.

??:

MT_DEEPL_KEY

MT_GOOGLE_KEY

Google Translate API v2 2 API 22 register at https://cloud.google.com/translate/docs 22 22 22: *Google Translate* 22 22 22 22 22

MT_GOOGLE_CREDENTIALS

MT_GOOGLE_PROJECT

MT_GOOGLE_LOCATION

Google Translate API V3 (Advanced)

MT_MICROSOFT_BASE_URL

MT_MICROSOFT_COGNITIVE_KEY

MT_MICROSOFT_REGION

MT_MICROSOFT_ENDPOINT_URL

MT_MODERNMT_KEY

ModernMT 22222222 API 222 22: ModernMT MT_MODERNMT_URL

MT_MODERNMT_URL

MT_MYMEMORY_EMAIL

MT_MYMEMORY_KEY

MT_MYMEMORY_USER

 22222222222
 MyMemory
 2222
 ID
 22MT_MYMEMORY_KEY
 22222222222
 222222222222

 Image: MyMemory
 22222222222222
 MyMemory: API key generator

MT_NETEASE_KEY

MT_NETEASE_SECRET

NetEase Sight API 2 App secret 2 https://sight.youdao.com/ 2222 22: NetEase Sight API machine translation

MT_TMSERVER

MT_YANDEX_KEY

MT_YOUDAO_ID

Youdao Zhiyun API 2222222 ID 22https://ai.youdao.com/product-fanyi-text.s 2222 22: Youdao Zhiyun API machine translation

MT_YOUDAO_SECRET

MT_SAP_BASE_URL

MT_SAP_SANDBOX_APIKEY

 2222222
 API
 22

 22:
 SAP Translation Hub
 22
 22

MT_SAP_USERNAME

MT_SAP_PASSWORD

SAP 22222 22: SAP Translation Hub22222222222

MT_SAP_USE_MT

NEARBY_MESSAGES

PAGURE_CREDENTIALS

```
PAGURE_CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "pagure.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

PAGURE_USERNAME

 22222
 4.3.2
 222

 222222
 222222
 222222

 22:
 PAGURE_CREDENTIALS
 Pagure

PAGURE_TOKEN

22222 4.3.2 222.
22222222222 API 2222 Pagure 22222222 2222
22:
PAGURE_CREDENTIALS Pagure Pagure API

RATELIMIT_ATTEMPTS

RATELIMIT_WINDOW

 22222
 3.2
 222

 222222
 300
 225
 22222

 222
 200
 225
 22222

 222
 27
 ATTEMPTS
 RATELIMIT_LOCKOUT

RATELIMIT_LOCKOUT

REGISTRATION_ALLOW_BACKENDS

?????? 4.1 ????.

REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]

??:

REGISTRATION_OPEN???

REGISTRATION_CAPTCHA

REGISTRATION_EMAIL_MATCH

REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org\$"

REGISTRATION_OPEN

??:

REGISTRATION_ALLOW_BACKENDS ?? REGISTRATION_EMAIL_MATCH???

REPOSITORY_ALERT_THRESHOLD

?????? 4.0.2 ????.

??:

alerts

REQUIRE_LOGIN

?????? 4.1 ????.

??????	LUGIN_REQUI	RED_URLS	???????REST	????????????????????????????????????	API
???????????	????????????????????????????????????	??????			

22: 222 222222 222222 22222 2222 Docker 2222 WEBLATE_REQUIRE_LOGIN 2222222

SENTRY_DSN

[2][2][2] 3.9 [2][2]. *Collecting error reports* [2][2][2][2] Sentry DSN[2]
[2][2]:
Sentry [2] Django [2][2]

SESSION_COOKIE_AGE_AUTHENTICATED

SIMPLIFY_LANGUAGES

SITE_DOMAIN

2:

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"
# Local development with IP address and port
SITE DOMAIN = "127.0.0.1:8000"
```

222: Docker 22222222222 22222 WEBLATE_ALLOWED_HOSTS 222222

??:

```
Set correct site domain? Allowed hosts setup? Correctly configure HTTPS WE-BLATE_SITE_DOMAIN? ENABLE_HTTPS
```

SITE_TITLE

SPECIAL_CHARS

????????

SPECIAL_CHARS = ("\t", "\n", "...")

SINGLE_PROJECT

?????? 3.8 ????.

??

SINGLE_PROJECT = "test"

STATUS_URL

Weblate ???????????????!?! URL?

SUGGESTION_CLEANUP_DAYS

?????? 3.2.1 ????.

UPDATE_LANGUAGES

setuplang

URL_PREFIX

??

URL_PREFIX = "/translations"

VCS_BACKENDS

22222 VCS 222222222

VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)

??:

VCS_CLONE_DEPTH

?????? 3.10.2 ????.

 $VCS_CLONE_DEPTH = 0$

WEBLATE_ADDONS

```
WEBLATE_ADDONS = (
    # Built-in addons
    "weblate.addons.gettext.GenerateMoAddon",
    "weblate.addons.gettext.UpdateLinguasAddon",
    "weblate.addons.gettext.UpdateConfigureAddon",
    "weblate.addons.gettext.MsgmergeAddon",
    "weblate.addons.gettext.GettextCustomizeAddon",
    "weblate.addons.gettext.GettextAuthorComments",
    "weblate.addons.cleanup.CleanupAddon",
    "weblate.addons.consistency.LangaugeConsistencyAddon",
    "weblate.addons.discovery.DiscoveryAddon",
    "weblate.addons.flags.SourceEditAddon",
    "weblate.addons.flags.TargetEditAddon",
    "weblate.addons.flags.SameEditAddon",
    "weblate.addons.flags.BulkEditAddon",
    "weblate.addons.generate.GenerateFileAddon",
    "weblate.addons.json.JSONCustomizeAddon",
    "weblate.addons.properties.PropertiesSortAddon",
    "weblate.addons.git.GitSquashAddon",
    "weblate.addons.removal.RemoveComments",
    "weblate.addons.removal.RemoveSuggestions",
    "weblate.addons.resx.ResxUpdateAddon",
    "weblate.addons.autotranslate.AutoTranslateAddon",
    "weblate.addons.yaml.YAMLCustomizeAddon",
    "weblate.addons.cdn.CDNJSAddon",
    # Addon you want to include
    "weblate.addons.example.ExampleAddon",
)
```

??:

???????DEFAULT_ADDONS

WEBLATE_EXPORTERS

Supported file formats

WEBLATE_FORMATS

??:

Supported file formats

WEBLATE_GPG_IDENTITY

22222 3.1 222.

WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"

??:

#

Signing Git commits with GnuPG

???????

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/>.

```
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_
\rightarrowerror:
            # "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.
# 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 = (
       GUAGES = (
("ar", "ăцָיָוֶצָּוֹ"),
("az", "Azərbaycan"),
("be", "Беларуская"),
("be@latin", "Biełaruskaja"),
("bg", "Български"),
("br", "Brezhoneg"),
("ca", "Català"),
("ca", "Català"),
("cs", "Čeština"),
("da", "Dansk"),
("de", "Deutsch"),
("en", "English"),
("el", "Eλληνικά"),
("en-gb", "English (United Kin
         ("en-gb", "English (United Kingdom)"),
        ("en-gb", "English (
("es", "Español"),
("fi", "Suomi"),
("fr", "Français"),
("gl", "Galego"),
("he", "עברית"),
("hu", "Magyar"),
("hu", "Hrvatski"),
("id", "Indonesia"),
        ("id", "Indonesia"),
("is", "Íslenska"),
("it", "Italiano"),
("ja", "202"),
("kab", "Taqbaylit"),
("kk", "Қазақ тілі"),
("ko", "202"),
("nb", "Norsk bokmål"),
("nb", "Norsk bokmål"),
("nl", "Nederlands"),
("pl", "Português"),
("pt-br", "Português br
         ("pt-br", "Português brasileiro"),
        ("pt-br", "Portugues b
("ru", "Pycский"),
("sk", "Slovenčina"),
("sl", "Slovenščina"),
("sq", "Shqip"),
("sr", "Српски"),
("sv", "Svenska"),
("tr", "Türkçe"),
("uk", "Українська"),
        ("zh-hans", "???"),
("zh-hant", "???"),
)
SITE_ID = 1
# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True
# If you set this to False, Django will not format dates, numbers and
# calendars according to the current locale.
USE L10N = True
# If you set this to False, Django will not use timezone-aware datetimes.
USE_TZ = True
# URL prefix to use, please see documentation for more details
URL_PREFIX = ""
# Absolute filesystem path to the directory that will hold user-uploaded...
\hookrightarrow 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",
            1,
             "loaders": _TEMPLATE_LOADERS,
        },
    }
1
# GitHub username for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None
GITHUB_TOKEN = None
# GitLab username 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.Bitbucket0Auth",
    # "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"
    "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
\hookrightarrow 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.

→NumericPasswordValidator"
},

    {"NAME": "weblate.accounts.password_validation.CharsPasswordValidator"}
\hookrightarrow,
    {"NAME": "weblate.accounts.password_validation.PastPasswordsValidator"}
\hookrightarrow,
    # Optional password strength validation by django-zxcvbn-password
    # {
    #
          "NAME": "zxcvbn_password.ZXCVBNValidator",
          "OPTIONS": {
    #
    #
              "min score": 3,
              "user_attributes": ("username", "email", "full_name")
    #
    #
          }
    # },
]
# Allow new user registrations
REGISTRATION_OPEN = True
# Shortcut for login required setting
REOUIRE 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",
1
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",
    # 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.
\hookrightarrowLOG_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
\hookrightarrow (message) s" },
        "simple": {"format": "[%(asctime)s: %(levelname)s/%(process)s]
\hookrightarrow (message) \bar{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": {
    "'" "TNFO
             "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"],
    """ "TNFO",
            "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_
\hookrightarrowLOGLEVEL},
        # Python Social Auth
        "social": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        # Django Authentication Using LDAP
        "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_
\hookrightarrowLOGLEVEL},
        # SAML IdP
        "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_
\hookrightarrowLOGLEVEL},
   },
}
# 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 referrer 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. \hookrightarrow check # Session cookie age (in seconds) SESSION_COOKIE_AGE = 1000 SESSION_COOKIE_AGE_AUTHENTICATED = 1209600 # Increase allowed upload size DATA_UPLOAD_MAX_MEMORY_SIZE = 5000000 # 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 # 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 = # 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.PerlFormatCheck",
      "weblate.checks.format.JavaScriptFormatCheck",
#
#
      "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",
#
 )
# 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.consistency.LangaugeConsistencyAddon",
#
      "weblate.addons.discovery.DiscoveryAddon",
#
#
      "weblate.addons.flags.SourceEditAddon",
      "weblate.addons.flags.TargetEditAddon"
#
      "weblate.addons.flags.SameEditAddon"
#
      "weblate.addons.flags.BulkEditAddon",
#
#
      "weblate.addons.generate.GenerateFileAddon",
#
      "weblate.addons.json.JSONCustomizeAddon",
      "weblate.addons.properties.PropertiesSortAddon",
#
      "weblate.addons.git.GitSquashAddon",
#
      "weblate.addons.removal.RemoveComments",
#
#
      "weblate.addons.removal.RemoveSuggestions",
      "weblate.addons.resx.ResxUpdateAddon"
#
#
      "weblate.addons.yaml.YAMLCustomizeAddon",
      "weblate.addons.cdn.CDNJSAddon",
#
      "weblate.addons.autotranslate.AutoTranslateAddon",
#
#
# 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_
\hookrightarrowBIDI": 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/$",
rf"{URL_PREFIX}/contact/$",
#
                                   # JavaScript localization
                                   # 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-
\hookrightarrow 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
SENTRY_DSN = None
AKISMET_API_KEY = None
```

Management commands

22: Running management commands under a different user than the one running your webserver can result in files getting wrong permissions, please check **2222 22222222** 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
```

??:

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).

??

auto_translate

weblate auto_translate <project> <component> <language>

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

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 222222).

--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.

??

??:

<u>?????</u>

celery_queues

weblate celery_queues

222223.72222Displays 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*.

??:

Running maintenance tasks? 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.

??:

Running maintenance tasks

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). 22222 2.9 222: Added parameters --username, --email, --name and --update.

dump_memory

weblate dump_memory
202222 2.20 2022.
Export a JSON file containing Weblate Translation Memory content.
20222222 Weblate 20222222

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

2????? 4.1 ????.

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>

?????? 2.7 ????.

Batch import of components based on JSON data.

The imported JSON file structure pretty much corresponds to the component object (see *GET /api/components/(string:project)/(string:component)/*). You have to include the name and filemask fields.

--project PROJECT Specifies where the components will be imported from.

--main-component COMPONENT

Use the given VCS repository from this component for all of them.

--ignore

Skip (already) imported components.

--update

Update (already) imported components.

[2][2][2][2] 2.9 [2][2]: The parameters --ignore and --update are there to deal with already imported components. Example of JSON file:

```
[
  {
    "slug": "po",
    "name": "Gettext PO",
    "file_format": "po",
    "filemask": "po/*.po",
    "new_lang": "none"
  },
  {
    "name": "Android",
    "filemask": "android/values-*/strings.xml",
    "template": "android/values/strings.xml",
    "repo": "weblate://test/test",
    "file_format": "aresource"
  }
]
```

??:

import_memory

import_memory

weblate import_memory <file>

?????? 2.20 ????.

Imports a TMX or JSON file into the Weblate translation memory.

--language-map LANGMAP Allows mapping languages in the TMX to the Weblate translation memory. The language codes are mapped after normalization usually done by Weblate.

--language-map en_US: en will for example import all en_US strings as en ones.

This can be useful in case your TMX file locales happen not to match what you use in Weblate.

??:

import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

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 Supported file formats), 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-
    il18N.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

?????? 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:

??:

2222

list_languages

weblate list_languages <locale>

Lists supported languages in MediaWiki markup - language codes, English names and localized names. This is used to generate https://wiki.l10n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF.

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.

222: 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

22.2.2.2.3.0 2.2.2.

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.

22: Weblate pushes changes automatically if <u>2222222222222222</u> 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

??:

222222

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.

22: Usually it is better to configure hooks in the repository to trigger **2022**, instead of regular polling by updategit.

????

[2][2][2][2] 4.0 [2][2]: In prior releases this feature was called whiteboard messages.

Provide info to your translators by posting announcements, site-wide, per project, component, or language.

Announce the purpose, deadlines, status, or specify targets for translation.

The users will receive notification on the announcements for watched projects (unless they opt out).

This can be useful for various things from announcing the purpose of the website to specifying targets for translations. The announcements can posted on each level in the *Manage* menu, using *Post announcement*:

🛛 Weblate	Dashboard	Projects 🗸	Language	s - Checks				s	L	+ Add -	•	•••
🖗 WeblateOrg											translate	d 90%
Translations wil	l be used only if t	they reach 6	0%.									×
Components	Languages	Info	Search	Glossaries	Insights +	Files +	Tools +	Manage -	Share +	Ø	lot watch	ing 🗸
Post announce	ment											0
Message												
You can use Mark Category	down and menti	on users by	@username.									
Info (light blu	ie)											~
Category defines	s color used for t	he message	2.									
Expiry date												
mm/dd/yyyy												
The message wil	ll be not shown a	after this dat	te. Use it to a	nnounce string	g freeze and tra	anslation dea	adline for ne	kt release.				
🗹 Notify users												
The message is s	shown for all tra	nslations w	ithin the proj	ect, until its g	iven expiry, or p	ermanently	until it is del	leted.				
Add												

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

285

It can be also added using the admin interface:

Weblate administration		WELCOME, WEBLATE TEST, RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT				
Home - Weblate translations	Announcements Add Announcement					
Add Announcement						
Required fields are marked in	n bold.					
Message:	Translations will be used only if they reach 60%.					
Project:	WeblateOrg 🗸 🤌 +					
Component:	v / +					
Language:	v /+					
Category:	Info (light blue) Category defines color used for the message.					
Expiry date:	Today M The message will be not shown after this date. Use it to announce strin	g freeze and translation deadline for next release.				
✓ Notify users						
		Save and add another Save and continue editing SAVE				

The announcements are then shown based on their specified context: No context specified Shown on dashboard (landing page). Project specified Shown within the project, including all its components and translations. Component specified Shown for a given component and all its translations. Language specified

Shown on the language overview and all translations in that language.

This is how it looks on the language overview page:

Weblate Dashboard	Projects 🗸 🛛 Language	es + Checks +		عر	🛦 + Add -	۰۰۰ - چ
$\overleftarrow{x}_{\!\!\!A}$ Languages / Czech						
Czech translators rock!						×
Projects Information	History Activity	Glossaries Tools -				
Project	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
🗨 🛛 WeblateOrg 🔟 🔢	97%	1	12	3		

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

?????? 2.13 ????.

1. Define Automatic component list assignment with $^{\circ}$. * \$ as regular expression in both the project and the component fields, as shown on this image:

Weblate adminis	tration	WELCOME, WEBLATE TEST. RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD /	
Home - Weblate translations	-> Component lists -> Add Component list		
Add Component list Required fields are marked ir	n bold.		
Component list name:	All components Display name		
URL slug:	all-components Name used in URLs and filenames.		
Show on dashboard When enabled this component lis	t will be shown as a tab on the dashboard		
Components:	Available components Q Filter WeblateOrg/Django WeblateOrg/Language names Choose all O	Chosen components Chosen components Chosen comp	
AUTOMATIC COMPONENT LI	Hold down "Control", or "Command" on a Mac, to select more tha	an une.	
PROJECT REGULAR EXPRESSION	NØ	COMPONENT REGULAR EXPRESSION @ DELETE?	,
^.*\$		^.*\$	C
+ Add another Automatic con	nponent list assignment		
		Save and add another Save and continue editing Save and continue editing	SAVE

Optional Weblate modules

Several optional modules are available for your setup.

Git exporter

?????? 2.10 ????.

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/master/.

Repositories for publicly available projects can be cloned without authentication:

git clone 'https://example.org/git/weblate/master/'

Access to the repositories with restricted access (using <u>22222222222222222</u> or when <u>REQUIRE_LOGIN</u> is enabled) requires a API token which can be obtained in your <u>222222</u>:

git clone 'https://user:KEY@example.org/git/weblate/master/'

??

????? 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 **2**[2][2][2][2].

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

12: Legal documents for the Hosted Weblate service are available in this Git repository https://github.com/WeblateOrg/wllegal/tree/master/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 *EN*-*ABLE_AVATARS*.

Weblate currently supports:

Gravatar

??:

Spam protection

You can protect against suggestion spamming by unauthenticated users by using the akismet.com service.

1.Install the akismet Python module

2.Configure the Akismet API key.

This (among other things) relies on IP address of the client, please see 2222 22222222 for properly configuring that.

??:

22.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.AKISMET_API_KEY

Signing Git commits with GnuPG

?????? 3.1 ????.

All commits can be signed by the GnuPG key of the Weblate instance.

1. Turn on *WEBLATE_GPG_IDENTITY*. (Weblate will generate a GnuPG key when needed and will use it to sign all translation commits.)

This feature needs GnuPG 2.1 or newer installed.

You can find the key in the DATA_DIR and the public key is shown on the "About" page:

₩eblate Dashboard Projects - Languages - Checks -	Register	Sign in	•••
🔘 About Weblate / Weblate keys			
About Weblate Statistics Keys			
SSH key			(i)
SSH key not available.			
Commit signing			Ū
All commits made with Weblate are signed with the GPG key C0F6E5B4379EF46B1DF13D27CBFA89B076096E48, for which the correspondence of the second s	esponding public key	is found be	low.
BEGIN PGP PUBLIC KEY BLOCK			
mQGNBF/Yd8sBDAC6NWpo5LpLY5rB2VNIdgGoM2Ohv9F4xC+IDRI4u6i7uuhAhvl0			
d+6RiCaErdJMIFb2Z+4VxDxccX2AJyGC7n1asDdgPKIu7LQ3D4IxaRw/L9iRkDn5 BsZ4gm3j1g7q4b9+FUIejjFAfRfKDAxjnQ77NiT93nze7FIAaL5fmUPtjClePIOG			
oYn8XZAeOJpBwAfF1uj3dBJYuuCpnBIoN1GOf4pNyImIciKpusqAquZI7t7krmYE			
p6oR07uAz3t/3egbyJaRzr8ECBaSFgRqx6OT1XxvjZEXhb6EdP7NGJzaWWJ8TlQq			
BnGHBRVS/Hrn7bsqGhsP0/jIOBUD8E7xiuyYSWw5mE4DYU5OqCKoTg48t/uaOMG0			
7gAe+mBnGlo+zhLFWVhYNEA0dlkRkNBvbKkU2koOYKjD91ePiPjHf3sO7PArKwJi uAWJ0FFSeaheYn6ASP9AjiMcnJRG6QMtOEol4JcvwMQY3MrHWDfJVnenwRcbk44v			
hwK7PViZH2prB8sAEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBIeGFtcGxILmNvbT6J			
Ac4EEwEKADgWIQTA9uW0N570ax3xPSfL+omwdgluSAUCX9h3ywIbAwULCQgHAgYV			
CgkICwIEFgIDAQIeAQIXgAAKCRDL+omwdgluSBBDDAC16BEvUHhpBLS7GTQukbeD			
Imzz9hXbW3VZUWf+NgQv7TtWh+/iECb2Pt4vH9V2s7K3NvfNn3IsyFx4jnsI0CYA			
GETRRQygUt0Zxv3l0aTQbLpfEdOUiere2vmTMfgDcBV/+uhRFRsxL5rIAtBv85bT G8EJwGUMTvOhVhC7inIkO5wS9cu3vnVv9ofD+B86ind9BtSw5OJYeN08Clguvm2+			•

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

2. Alternatively you can also import existing keys into Weblate, just set HOME=\$DATA_DIR/home when invoking gpg.

??:

WEBLATE_GPG_IDENTITY

????

??	??	???	????????	???????????????????????????????????????
??	REGISTRATI	on5	300	600
????????????????????????????????????	MESSAGE	5	300	600
????????????????????????????????????	2000 LOGIN	5	300	600
??????????	SEARCH	6	60	60
??	TRANSLATE	30	60	600
????????????????????????????????????	GLOSSARY	30	60	600

22: 2222222222 2222222222 API 2222

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.

12: 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:

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

Branding appears in the following files:

/weblate.svgLogo shown in the navigation bar.

logo-*.pngWeb icons depending on screen resolution and web-browser.

favicon.icoWeb icon used by legacy browsers.

eblate-*.pngAvatars for bots or anonymous users. Some web-browsers use these as shortcut icons.

ail-logo.pngUsed 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.

??:

Managing static files (e.g. images, JavaScript, CSS) Serving static files


```
# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)
# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)
# Addons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

??:

Management interface

The management interface offer administration settings under the /manage/ URL. It is available for users signed in with admin privileges, accessible by using the wrench icon top right:

₩eblate Dashboard Projects - Languages - Checks -	ع	+ Add -)			
🔧 Manage						
Weblate status Backups Translation memory Performance report SSH keys Alerts Repositories	Users	s Appearan	ce			
Tools Billing						
Weblate support status			Ū			
Support status Community support						
Purchase support package Donate to Weblate						
Activate support package			Ū			
The support packages include priority e-mail support, or cloud backups of your Weblate installation.						
Please enter the activation token obtained when making the subscription.						
Activate Purchase support package						
Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate						

It includes basic overview of your Weblate:

The Django admin interface

Will be removed in the future, as its use is discouraged—most features can be managed directly in Weblate.

Here you can manage objects stored in the database, such as users, translations and other settings:

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 + Add	
venired entans	- Aud	🤌 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
GLOSSARIES		
Glossaries	+ 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		Ch.
Languages	+ Add	🤌 Change
WEBLATE TRANSLATIONS		
	+ Add	🥜 Change
Announcements		
Component lists	+ Add	Change
Components	+ Add	Change
Contributor agreements	+ Add	🤌 Change
Projects	+ Add	🥓 Change

Recent actions

My actions None available In the *Reports* section, you can check the status of your site, tweak it for *Production setup*, or manage SSH keys used to access 222222222222222.

Manage database objects under any of the sections. The most interesting one is probably *Weblate translations*, where you can manage translatable projects, see *Project configuration* and *Component configuration*.

Weblate languages holds language definitions, explained further in Language definitions.

Adding a project

Adding a project serves as container for all components. Usually you create one project for one piece of software, or book (See *Project configuration* for info on individual parameters):

Weblate adminis	stration	WELCOME, WEBLATE TEST. RETURN TO WEBLA	TE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT
Home - Weblate translation:	s - Projects - Add Project		
Add Project			
Required fields are marked i	n bold.		
Project name:	WeblateOrg Display name		
URL slug:	weblateorg Name used in URLs and filenames.		
Project website:	https://weblate.org/ Main website of translated project.		
Mailing list:	weblate@lists.cihar.com Mailing list for translators.		
Translation instructions:	https://weblate.org/contribute/		
Set "Language-Team" hea Lets Weblate update the "Langu	ider age-Team' file header of your project.		
✓ Use shared translation m Uses the pool of shared translat			
Contribute to shared trans Contributes to the pool of share			
Controlo de acesso:	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 r	epository by remote hooks.		
Language aliases:	Comma-separated list of language code mappings, for example: en_GB	en,en_US;en	
		Save and add an	Nother Save and continue editing SAVE

??:

Project configuration

Bilingual components

Once you have added a project, translation components can be added to it. (See *Component configuration* for info regarding individual parameters):

Weblate adminis	tration инсцоихе инныте техт, нетики то извыште / осоционталиси / сныме вызвилио / нем оит
Home - Weblate translations	
Add Component Required fields are marked in	INFORT SPEED DO COMENTATION
Component name:	Long. Langsage names
	Display name
URL alog:	Regulage-names Neme used in USLs and Histories.
Project:	Weblanding 👻 🥖 +
Version control system:	CR V Wrision control system to use to access your reporting containing translations. You can also choose additional integration with third party providers to submit merge requests.
Source code repository:	Imput/Sylthub.com/Weblanchg/domo.gt/ UR: di a regionary use weblan: ()project/component: to share it with other component.
Repository push URL:	
Repository browser.	UR, of a push-repositivity pushing is seried of if empty https://ythab.com/weblatching/demos/blok/lybance/(//)filename()#1.
	(improviginational in measure of point induced grant and grant and grant of a grant and grant of a
Exported repository URL:	UID, of repository where users can fetch changes from Weblate
Source string bug reporting address:	Certail address for inports on entrors in source strongs, Lawe empty for no e-mails,
Repository branch:	
Push branch:	Repository branch to transfere
	Breach for pushing changes, leave enspry to use repository branch
Filemask:	webbins/targ/strat/scc.lw/%CC_MESEADES/dp Path of files to tousider relative to reporting visit, use * instead of longuage code, for example, por* po or locale/*/LC_MESEADES/dpinguage.
Monolingual base language file:	
dit base file	Fiferance of translation base file, containing all strings and their source; it is recommended for monological translation formats.
Whether users will be able to edi Intermediate language file:	
	Ference of Internetions translation file. In most cases this is a translation file provided by developers and is used when creating actual source attrings.
Template for new translations:	webbah langkita localedginga pot Filmame el His une for enustra y environmentations. For gettest choses pot His.
File format:	getted PO file v
Locked Locked component will not get a	ny tanalatan updatas.
Allow translation propagal Whether translation updates in o	Son Ene companents will cause automatic translation is this one
Turn on suggestions Whether to allow translation sup	gerion et al.
Suggestion voting Whether users can rote for sugg	#rfons,
Autoaccept suggestions:	0
Translation flags:	Automatically accept suggestions with this number of votes, use 0 to turn it off.
	Additional comma separated flags to influence quality checks. Ponsible values can be found in the documentation.
Enforced checks:	0
	Unit of sheeks which can not be ignored.
Translation license:	ONU General Public License x3.0 or later 🗸
Contributor agreement:	
	User anywervert which needs to be approved before a user can transfere this component.
Adding new translation	Create new larguage file 🔹 👻
Language code style:	Here to hardle requests for creating one translations. Default based on the file format
	Customize language code used to generate the Elename for translations created by Weblane.
Merge style:	Rebase Other whether Weblass should merge the uppresent repository or rebase charges onto is.
Commit message when translating:	Translated using Weblate (() language_name (i)) Currently translated at () states translated_percent ()/s (i) states translated () of () states uil () strings)
	Translation: () project_name ()v() component_name () Translate USL: () vf ()
Commit message when	You can use semplate language for versions into please consult the documentation for more strains. Added translation using Weblate (i) language_merre (i)
adding translation:	
Commit message when removing translation:	Deleted translation using Weblate ()] languaga.name (i)
	Vou can use template language for various info, please consult the documentation for more details.
Commit message when merging translation:	Marge branch () component_semole_branch () into WebBale.
	You can use template language for various info, please consult the documentation for more details.
Commit message when adden makes a change:	Update translation files
and a charge	Updated by '#addou_name ii' hook in Weblate. Translation: @project_tranes@ii' (or opponent_name ii) Translation: Uii: [10]
	- concentration over - II on (5)
	You can use template language for various info, please consult the documentation for more details.
	The care can enclose lenguage for version will be been executed for more stable.
	noreply@weblate.org
	s pushed sportware on every convent.
Age of changes to commit:	24 Time in hours after which any pending changes will be converted to the VCE.
Lock on error Whether the component should be	or locked on repeating errors.
	English
	Marchedex M
	Negular elphasision used to beservine variants of a string.
Priority:	Components with higher priority are offered first to translation.
Restricted component Restrict access to the component	It to only those explicitly gives permission.
	Sove and add acother Save and contrase edding SAVE

22: Component configuration Bilingual and monolingual formats

Monolingual components

For easier translation of these, provide a template file containing the mapping of message IDs to its respective source language (usually English). (See *Component configuration* for info regarding individual parameters):

in units of the lange of the la
in units of the lange of the la
and anon cannot request
com u
com y
www.er.atub.ender.to.for.s.atur.com.tencion.this.imposer.
entreet which needs to be approved before a user can transform this component.
ensert which needs to be approved before a user can translate this component.
remerkingsage file v
It based on the file format v
where Weblates should merger the operations repository or initiates changes onto it.
ly translated at ((stats.translated, percent))% ()(stats.translated () of ((stats.all)) strings)
alion: () project, name (),(i) component, name ()) 06-URL () wf ()
as template language for various lists, please consult the documentation for more details.
translation using Weblate (() language_name (i)
as template language for various info, please consult the documentation for more details.
d translation using Weblate ()] language_name ())
are template language for various info, please consult the documentation for more details.
branch '() component, semote, branch ()' into Weblate.
as template language for various info, please consult the documentation for more details.
tranaliston files d by 'il addon_name (i' hook in Weblate.
Source () project_name () () component_name () alse UBL: () wil ()
are template language for various info, please consult the documentation for more details.
a de la constante de
gwediata arg
@weblats org
gewitat og
gewither org
geedata ng aastar is ang saast aa salar is ang saast aastar is ang saast ang aastar ang ang aastar ang ang ang ang ang ang ang ang ang ang ang ang ang ang ang ang
gewither org
Gueddan og enner en
geneticating and the state product program will be substituted to the ST. and the same program will be same pr
genetia en general en ante en

22: *Component configuration Bilingual and monolingual formats*

Getting support for Weblate

Integrating support

????? 3.8 ????.

Purchased support packages can optionally be integrated into your Weblate subscription management interface, from where you will find a link to it. Basic instance details about your installation are also reported back to Weblate this way.

Weblate Dashboard Projects -	Languages 🗸 Checks 🗸					م +	Add -	+	•••
🔌 Manage									
Weblate status Backups Translati	tion memory Performan	ce report SS	SH keys A	Alerts	Repositories	Users	Appeara	ance	
Tools Billing									
Weblate support status									(i)
Support status		Community sup	port						
Purchase support package Do	onate to Weblate								
Activate support package Ø							0		
The support packages include priority e-mail	il support, or cloud backups of	your Weblate ins	tallation.						
Activation token									
Please enter the activation token obtained w	when making the subscription	l.							
Activate Purchase support page	ackage								

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

Data submitted to the Weblate

URL where your Weblate instance is configured

Your site title

The Weblate version you are running

Tallies of some objects in your Weblate database (projects, components, languages, source strings and users)

The public SSH key of your instance

No other data is submitted.

Integration services

See if your support package is still valid Weblate provisioned backup storage

Purchased support packages are already activated upon purchase, and can be used without integrating them.

Legal documents

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 ???

Weblate ???

Weblate ???????

Weblate ??????????

Weblate ?????????

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2.virtualenv ?????

```
virtualenv .venv
.venv/bin/activate
```

pip install -e .

pip install -r requirements-dev.txt

4.??????????

weblate runserver

./weblate/examples/celery start

. scripts/test-database

./manage.py test

??:

Docker??????? Weblate ???

./rundev.sh

Dockerfile 2 docker-compose.yml 22 dev-docker 22222222222

./rundev.sh test --failfast weblate.trans

./rundev.sh logs

./rundev.sh stop

PyCharm ???? Weblate ????????

(B)		Weblate [/tmp/Weblate]/weblate/celery.py - PyCharm	^ _ O X
Eile Edit View Navig	gate <u>C</u> ode <u>R</u> efactor R <u>u</u> n <u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> el late) 💑 celery.py	·	± 5, 6, 5, ≣ Git: ✔ ✔ ③ 5 Q
ਜ਼ੂ 🔲 Project 👻	÷ + -	💑 celery.py 🗵	
Partial and a set of the se	es	<pre>@ celery.py × Package requirements 'celery/redis]>=4.2.0<4.5', 'kombulredis]>=4.6.0,1=4.6.4,1=4.6.5' are not s Ins def handle_task_failure.connect def handle_task_failure(exception=Hone, **knargs):</pre>	-
initi celery.py دواوي.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دواو.py دور دواو.py دور دور دور دور دور دور دور دور دور دور	y y vare.py	<pre>67 if not bool(os.environ.get('CELERY_WORKER_RUNNING', False)): 68 return 69 70 from weblate.utils.errors import init_error_collection 73</pre>	Project Interpreter 💎 Python 3.8 (weblate- 🥏 Python 3.8
Terminal @ Py	y ython Console III <u>6</u> : TODO	on for the following file(s): dev-docker/Dockerfile // Do not ask again (today 13:43) 1:1 LF UTF-8	Interpreter Settings Add Interpreter 4 spaces Git: master Python 3.8 🖕 👮 🎭



a de la companya de la compan	Add	Python Interpreter		^			
🐳 Virtualenv Environment	New environment	:					
🔾 Conda Environment	Location:	/tmp/Weblate/.venv					
ቅ System Interpreter	Base interpreter	🥏 /usr/bin/python3	•] [
🎼 Pipenv Environment	🗌 Inherit globa	🗌 Inherit global site-packages					
SSH Interpreter	🗌 Make availat	Make available to all projects					
V Vagrant	O Existing environm	ient					
Docker	Interpreter: <	No interpreter>	Ŧ] [
🖢 Docker Compose	🗌 Make availat	le to all projects					
			OK Car	ncel			
222222222222222222222222222222222222222	222222222222222	222222222: 22222222222	2??????????IDE ??????	???			

<u>e</u>	Sett	ings	^ X
Qr django X	Languages & Frameworks > D	jango @ For current project	Reset
Keymap	🗹 Enable Django Support		
▼ Editor	Django project root:	/tmp/Weblate/	_
▼ General	Settings:	weblate/settings_test.py	
Smart Keys	-		
HTML/CSS	Do not use Django test runner:		
Color Scheme	Manage.py tasks		
Django/Jinja2 Template	Manage script:	manage.py	<u>_</u>
Inspections @	Environment variables:		
Live Templates	Environment variables:		Ξ
Intentions	Folder pattern to track files:	migrations	
Build, Execution, Deployment	Forder pattern to track mes:	Separate several names with colon. Glob-style wildcards are s	upported
▼ Console @	Show structure		
Django Console 👘	Show structure		
Languages & Frameworks			
Django 🖻			
		OK Cancel Apply	Help

[🕘 Unnamed 🔻] 🕨	ð	G	4		i.	Git:	×	~	G	5	QZ
Edit Configurations													
Save 'Test: weblate.formats.tests.test_exporters.CSVExporterTest' Configuration													
Unnamed													
Test: weblate.formats.tests.test_exporters.CSVExporterTest													

		Run/Debug C	Configurations		^		
+ - 恒 🖌 🔺 🖬 🕸	2	<u>N</u> ame: Unnamed		Share through VCS	Allow parallel_ru		
Add New Configuration							
📀 App Engine server		Configuration Logs					
🚡 Attach to Node.js/Chrome		Heat	localhost		Port: 8000		
B Behave	ests.xwiki.test_	Host:	localhost		Port: 8000		
🖡 Compound	ests.xwiki.test_	Additional options:					
ق Django server	ests.test_format						
🚯 Django tests	ests.test_format	N	http://localhost:8000/				
🚽 Docker 🛛 🕨 🕨	ests.IntegrationT						
실 Firefox Remote			Start JavaScript debugger	automatically when debugging			
👅 Flask server		Custom run command:					
🖁 Grunt.js		Test server					
🖠 Gulp.js							
🔚 HTTP Request		🗹 No reload					
oript Debug		 Environment 					
🎽 Jest							
🔋 Lettuce		Environment variables:	IONUNBUFFERED=1;DJANG	D_SETTINGS_MODULE=weblate	settings_test		
🖬 npm		Python interpreter:	Project Default (Python 3	.8 (weblate)) ~/Github/weblate/.v	env/bin/python 🔻		
NW.js			•••••••••••••••••••••••••••••••••••••••				
Protractor		Interpreter options:			с ^я		
🎠 Pyramid server							
🔁 Python		Working directory:					
🗿 Python docs 🛛 🕨 🕨							
Python Remote Debug		Add content roots to PY					
🗞 Python tests 🛛 🕨 🕨		Add source roots to PYT	HONPATH				
🖗 React Native							
Shell Script		 <u>B</u>efore launch: Activate to 	ool window				
뷶 tox		+ - /					
		• • •					
			There are no tasks to ru	in before launch			
				OK Cancel	Apply Help		
				Cancel	негр		

22: Be careful with the property called *No reload*: if you check it, the server live reloads won't happened when you modify files. This allows the existing debugger breakpoints to persist as these would be discarded on reload.

Weblate ???? ???

?????????

pre-commit run --all

Debugging Weblate

Bugs can behave as application crashes or as misbehavior. You are welcome to collect info on any such issue and submit it to the issue tracker.

????????

Turning on debug mode will make the exceptions show in the browser. This is useful to debug issues in the web interface, but not suitable for production environment as it has performance consequences and might leak private data.

??:

Disable debug mode

Weblate logs

Weblate can produce detailed logs of what is going in the background. In the default configuration it uses syslog and that makes the log appear either in /var/log/messages or /var/log/syslog (depending on your syslog daemon configuration).

The Celery process (see *Background tasks using Celery*) usually produces own logs as well. The example system-wide setups log to several files under /var/log/celery/.

Docker containers log to their output (as usual in the Docker world), so you can look at the logs using dockercompose logs.

??:

??????? contains LOGGING configuration.

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 Background tasks using Celery

Check Celery queue status either in *Management interface* 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 *Management interface* under the *Tools* tab.

These send e-mail directly, so this verifies that your SMTP configuration is correct (see <u>22222222</u>). 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 *Collecting error reports*.

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 *Collecting error reports* 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 *Collecting error reports* and get pinpointed and detailed tracebacks in the error collection tool.

Weblate ???

????????

?????

accounts addons api Django REST framework ???????? API? auth ?????????? billing checks fonts formats gitexport lang legal

machinery memory screenshots trans utils vcs wladmin 22222222 Django 222222

????????

class weblate.addons.base.**BaseAddon**(*storage=None*)

configure (settings) 2222222222

daily(component)

get_settings_form(user, **kwargs) 22222222222222222222

post_commit (component)

pre_commit (translation, author)

stay_on_create = False

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.

```
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 ???????

???????

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.

????????

????

??:

Translation topic in the Django documentation

????

Weblate ?????

??????????

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.

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 COEE 5A32 9C27 B313 42B7 511D. You can also get the PGP key from Keybase.

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
./ci/run-setup
```

Local testing

To run a testsuite locally, use:

DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test

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.
```

??:

See Testing in Django for more info on running and writing tests for Django.

?????????

Weblate ???????

	s/weblate-memory.schema.js	son	
? ??	[2][2] [2][2][2][2][2][2][2][2][2]		
212	?	<u>?????????</u>	
	22222		
	category	<u>[2][2][2][2][2][2][2][2]</u> [2][2][2][2][2][2][2][2][2][2][2][2][2][
	8.	1 = [?]?[?]?[?]2 = [?]?[1000 [?]?[?]?[?]?[?]20000000 [?]?	0000 ??? =
		22222222222000000 22] = ??????
		?	[2][2]
		? ??	1
		<u>?????</u> ?	0 1
	origin	<u>[][][][][][][][]</u> [][][]	1
	or igni		
		?	[2][2][2]
		?	test
		??????	
	source	<u>???</u>	ାରାହାର
		? ?	<u>????</u> Hello
		minLength	1
		22222	1
	source_language	[2][2][2][2][2]	
		ISO 639-1 / ISO 639-2 / IE	
		?	[2][2][2]
		? ?!?!?!?	en ^[^]+\$
		22222	[]+\$
	target	<u>[2][2][2][2][2][2][2][2][2][2][2][2][2][</u>	
	8	?	[?]?]?]
		2	Ahoj
		minLength	1
	toward law awara	2221212 21212121212	
	target_language	ISO 639-1 / ISO 639-2 / IE	TF BCP 47
		?	2222 2222
		?	CS
		2222	^[^]+\$
		??????	
	??????????	False	

??

??:

22.2.2.2.2.dump_memory?import_memory

https://weblate.org/so	chemas/weblate-userdat	a.schema.json
?	[?]?]?]?]?]	·
??????		
basic	[?][?]	
	?	[?]?]?]?]?]
	?????	
	username	Username
		?
		?
		??????
	full_name	<u>??</u> ????
		?
		?

<u>???</u> ???

?????
Weblate ????

?????????

	Table 5 - ??????????????????????????????????					
	email	22222 222222				
	Cinan	?	[2][2][2]			
		?	noreply@example.com			
	data isinad	?????? ??????				
	date_joined		[2][2][2]			
		?	2019-11-18T18:53:54.862Z			
01	രത്രത്രത	??????				
profile	<u>2 2 2 2 2 2</u> 2	[2][2][2][2][2]				
	22222					
	language	<u>?</u> ??				
		? ?	[2][2][2] CS			
		2222	^.*\$			
		??????				
	suggested	<u>?????????????</u> ???????????????????????	ାରା			
		? ?	<u>?</u> 1			
		??????	0			
	translated	<u>????????????</u> D	তাতা			
		? ?	212 24			
		??????	0			
	uploaded	<u> </u>				
		? ?	<u>[2][2]</u> 1			
		??????	0			
	hide_completed		12 2 2 2 2 2 2			
		? ?	boolean False			
		??????	True			
	secondary_in_zen	Zen ????????????????????????????????????				
		? ?	boolean True			
		??????	True			
			121212121212			
	hide_source_seconda	?	boolean False			
		??????	True			
	editor_link	<u>?!?!?!?</u> <u>?!?!?</u>	ାରାରାର			
		? ?	[2][2][2]			
			^.*\$			
	translate_mode	[<u>]]]]]]]</u> []	<u>?</u> [?]			
		?	0			
	-		0			
	zen_mode	Zen [?]?]?]?] [?]	<u>?</u> [?]			
		?	0			
			0			
	special_chars	[<u>]]]</u>]]] []	21212			
		?				
		222	^.*\$			
	dashboard_view	?!?!?!?!?!?!?!?!?!?!?!?!?!?!?!?!?!?!?!	7			
		?				
		2	1			
		?!?!?!?!?!?!?!?!?!?!?!?!?!?!?!?!?!?!?!	0 <u>?!?</u>			
	dashboard_compone		null			
			?			

	Tal	ble 5 – ???????? ? ?????	?	null
	languages	2222 2 22222 222	2 212 2121212121	20
	secondary_languages		2 2 2[2][2] 2[2][2][2]	[2][2][2] cs ^.*\$
		2 21212122 212	2 2 2 2 2 2 2 2 2 2	[2][2][2] sk
		22/2/2/2/2/2/2/2/2/ 2 22(2)2/2	2121212 21212122 212	^.*\$
		22	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	272 weblate ^.*\$
auditlog	[2][2][2] [2] [2][2][2][2][2] [2][2]	<i>[2][2]</i> [2] [2] [2][2][2][2]	[2][2][2][2][2]	
		address	<i>IP</i> [2][2][2] [2] [2][2][2] [2][2][2][2] [2][2][2][2]	222 127.0.0.1 ^.*\$
		user_agent	2022 22220 2 2 2022	22/2 PC / Linux / Firefox 70.0 ^.*\$
		timestamp	21212122 212121212121 21 21 21 21 21 21	2019-11- 18T18:58:30.845Z
		activity	2121212 21212122 21 <u>21212</u> 21 21 21 21 21 21	^.*\$ <u>21212</u> 2121212
22			212122 212121212	^.*\$

шш

??:

?????????dumpuserdata

Weblate ?????

Weblate has two month release cycle for releases (x.y). These are usually followed by a bunch of bugfix releases to fix issues which slip into them (x.y.z).

The change in the major version indicates that the upgrade process can not skip this version - you always have to upgrade to x.0 before upgrading to higher x.y releases.

??:

Upgrading Weblate

Release planning

The features for upcoming releases are collected using GitHub milestones, you can see our roadmap at https://github.com/WeblateOrg/weblate/milestones>.

Release process

1../scripts/list-translated-languages 2222222222222

2../scripts/prepare-release 222222222222

- 5.Docker ?????????

Weblate git ???????? push ??????? push?

Weblate ????

??????????

????????

???????? Web ????

??????????

???

???

?????

Copyright (C) 2012 - 2021 Michal Čihař <michal@cihar.com>

????

Weblate 4.4.2

Released on January 14th 2021. Fixed corruption of one distributed MO file.

Weblate 4.4.1

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

Weblate 4.3.1

Weblate 4.3

Released on October 15th 2020. Include user stats in the API. Fixed component ordering on paginated pages. Define source language for a glossary. Rewritten support for GitHub and GitLab pull requests. Fixed configuration of enforced checks. Improve documentation about built-in backups. Moved source language attribute from project to a component. 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. 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

Weblate 4.2.1

Released on August 21st 2020. Fixed saving plurals for some locales in Android resources. Fixed crash in the cleanup addon for some XLIFF files. Allow setting up localization CDN in Docker image.

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. Removed leading dot from JSON unit keys. 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 22. CDN. The Weblate domain is now configured in the settings, see *SITE_DOMAIN*.

Weblate 4.1.1

Released on June 19th 2020.
Fixed changing autofix or addons configuration in Docker.
Fixed possible crash in "About" page.
Improved installation of byte-compiled locale files.
Fixed adding words to glossary.
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 authentication.

Weblate 4.1

Released on June 15th 2020.

Added support for creating new translations with included country code.

Added support for searching source strings with screenshot.

Extended info available in the stats insights.

Improved search editing on "Translate" pages.

Improve handling of concurrent repository updates.

Include source language in project creation form.

Include changes count in credits.

Fixed UI language selection in some cases.

Allow to whitelist registration methods with registrations closed.

Improved lookup of related terms in glossary.

Improved translation memory matches.

Group same machinery results.

Add direct link to edit screenshot from translate page.

Improved removal confirmation dialog.

Include templates in ZIP download.

Add support for Markdown and notification configuration in announcements.

Extended details in check listings.

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.

Renamed shapings to more generic name variants.

Added new quality checks: 2222222, 222222, 2222222.

Reintroduced support for wiping translation memory.

Fixed option to ignore source checks.

Added support for configuring different branch for pushing changes.

Weblate 4.0.4

Released on May 07th 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 02nd 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 22. Improved comments extraction from monolingual po files. Renamed whiteboard messages to announcements. Fixed occasional problems with registration mails. Improved LINGUAS update addon to handle more syntax variants. Fixed editing monolingual XLIFF source file. Added support for exact matching in **2?**. Add support for source upload on bilingual translations. Added support for intermediate language from developers.

Added support for source strings review.

Extended download options for platform wide translation memory.

Weblate 3.x series

Weblate 3.11.3

Released on March 11th 2020. Fixed searching for fields with certain priority. Fixed predefined query for recently added strings. Fixed searching returning duplicate matches. Gmail [202020202020] Fixed reverting changes from the history. Added links to events in digest notifications. Fixed email for account removal confirmation.

Weblate 3.11.2

Released on February 22nd 2020. 222222222 Fixed some strings wrongly reported as having no words.

Weblate 3.11.1

Released on February 20th 2020. Documented Celery setup changes. Improved filename validation on component creation. Fixed minimal versions of some dependencies. Fixed adding groups with certain Django versions. Fixed manual pushing to upstream repository. Improved glossary matching.

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 <u>222222</u>. Increased default validity of confirmation links. Improved Matomo integration. Fixed <u>22222</u> to correctly handle source string change. Extended automatic updates configuration by <u>AUTO_UPDATE</u>. 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. Fixed searching on systems with non-English locales. Improved repository matching for GitHub and Bitbucket hooks. Fixed data migration on some Python 2.7 installations. Allow configuration of Git shallow cloning. Improved background notification processing. Fixed broken form submission when navigating back in web browser. New addon to configure YAML formatting. Fixed same plurals check to not fire on single plural form languages. Fixed regex search on some fields.

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.
2022 2020202020202020
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 read only strings. Added support for Markdown in comments. Allow placing translation instruction text in project info. Add copy to clipboard for secondary languages. Improved support for Mercurial. Improved Git repository fetching performance. Add search lookup for age of string. Show source language for all translations. Show context for nearby strings. Added support for notifications on repository operations. Improved translation listings. Extended search capabilities. Added support for automatic translation strings marked for editing. Avoid sending duplicate notifications for linked component alerts. Improve default merge request message. Better indicate string state in Zen mode. Added support for more languages in Yandex Translate. Improved look of notification e-mails. Provide choice for translation license.

Weblate 3.9.1

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

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.
Make Microsoft Terminology service compatible with current Zeep releases.
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. 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. Rewritten Microsoft Terminology machine translation backend. Weblate now uses Celery to offload some processing. Improved search capabilities and added regular expression search. Added support for Youdao Zhiyun API machine translation. Added support for Baidu API machine translation. Integrated maintenance and cleanup tasks using Celery. Improved performance of loading translations by almost 25%. Removed support for merging headers on upload. Removed support for custom commit messages. Configurable editing mode (zen/full). Added support for error reporting to Sentry. Added support for automated daily update of repositories. Added support for creating projects and components by users. Built in translation memory now automatically stores translations done. Users and projects can import their existing translation memories. Better management of related strings for screenshots. Added support for checking Java MessageFormat. See 3.2 milestone on GitHub for detailed list of addressed issues.

Weblate 3.1.1

Released on July 27th 2018. Fix testsuite failure on some setups.

Weblate 3.1

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 built in translation memory. 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. Improved built in machine translation service. Added support for SAP Translation Hub service. Added support for Microsoft Terminology service. Removed support for advertisement in notification e-mails. 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. Locales updates.

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 direct link to edit secondary language translation. 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 2.12

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. 2022/2022/2022/2022/2022/20 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 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.

Highlight secondary language if not showing source.

Record suggestion deletion in history.

Improved UX of languages selection in profile.

Fixed showing whiteboard messages for component.

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.

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.

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. Added option to specify source language. Improved support for XLIFF files. Extended list of options for import_project. Improved targeting for whiteboard messages. Support for automatic translation across projects. Optimized fulltext search index. Added management command for auto translation. Added placeables highlighting. Added keyboard shortcuts for placeables, checks and machine translations. Improved translation locking. Added quality check for AngularJS interpolation. Added extensive group based ACLs. Clarified terminology on strings needing action and not translated strings. Support for Python 3. Dropped support for Django 1.7. Dropped dependency on msginit for creating new gettext PO files. Added configurable dashboard views. Improved notifications on parse errors. Added option to import components with duplicate name to import_project. Improved support for translating PHP files. Added XLIFF export for dictionary. Added XLIFF and gettext PO export for all translations. Documentation improvements. Added support for configurable automatic group assignments.

Improved adding of new translations.

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. Allow to show secondary language in Zen mode. Support for hiding source string in favor of secondary language.

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

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 ttkit loader. Added sitemaps to allow easier access by crawlers. Provide direct links to string in notification e-mails or feeds. Various improvements to admin interface. Provide hints for production setup in admin interface. Added per language widgets and engage page. Improved translation locking handling. Show code snippets for widgets in more variants. Indicate failing checks or fuzzy strings in progress bars. More options for formatting commit message. Fixed error handling with machine translation services. Improved automatic translation locking behaviour. Support for showing changes from previous source string. Added support for substring search. Various quality checks improvements. Support for per project ACL. Basic code coverage by unit tests.

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. **347** 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.

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 ttkit). 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.

349

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. Displays secondary languages while translating. 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,108 wlc.config,109 wlc.main,109

/ ANY /, 67 /api GET /api/, 69 /api/addons GET /api/addons/,96 GET /api/addons/(int:id)/,96 PUT /api/addons/(int:id)/,97 DELETE /api/addons/(int:id)/,97 PATCH /api/addons/(int:id)/,97/api/changes GET /api/changes/,94 GET /api/changes/(int:id)/,94/api/component-lists GET /api/component-lists/,97 GET /api/component-lists/(str:slug)/,97 POST /api/component-lists/(str:slug)/components/,98 PUT /api/component-lists/(str:slug)/,97 DELETE /api/component-lists/(str:slug)/,97 DELETE /api/component-lists/(str:slug)/components/(str:component_slug),98 PATCH /api/component-lists/(str:slug)/,97/api/components GET /api/components/,81 GET /api/components/(string:project)/(string:component)/,81 GET /api/components/(string:project)/(string:component)/changes/,84 GET /api/components/(string:project)/(string:component)/lock/,85 GET /api/components/(string:project)/(string:component)/monolingual_base/,87 GET /api/components/(string:project)/(string:component)/new_template/,87 GET /api/components/(string:project)/(string:component)/repository/,86 GET /api/components/(string:project)/(string:component)/screenshots/,85 GET /api/components/(string:project)/(string:component)/statistics/,88 GET /api/components/(string:project)/(string:component)/translations/,87 POST /api/components/(string:project)/(string:component)/addons/,96 POST /api/components/(string:project)/(string:component)/lock/,85 POST /api/components/(string:project)/(string:component)/repository/,86 POST /api/components/(string:project)/(string:component)/translations/,87 PUT /api/components/(string:project)/(string:component)/,84 DELETE /api/components/(string:project)/(string:component)/,84 PATCH /api/components/(string:project)/(string:component)/,83/api/glossary GET /api/glossary/,98 GET /api/glossary/(int:id)/,98 GET /api/glossary/(int:id)/projects/,99 GET /api/glossary/(int:id)/terms/,100 GET /api/glossary/(int:id)/terms/(int:term_id)/,100 POST /api/glossary/(int:id)/projects/,100 POST /api/glossary/(int:id)/terms/,100 PUT /api/glossary/(int:id)/,99 PUT /api/glossary/(int:id)/terms/(int:term_id)/,100 DELETE /api/glossary/(int:id)/,99 DELETE /api/glossary/(int:id)/projects/,100

```
353
```

```
DELETE /api/glossary/(int:id)/terms/(int:term_id)/,100
PATCH /api/glossary/(int:id)/,99
PATCH /api/glossary/(int:id)/terms/(int:term_id)/, 100/api/groups
GET /api/groups/,72
GET /api/groups/(int:id)/,72
POST /api/groups/,72
POST /api/groups/(int:id)/componentlists/,74
POST /api/groups/(int:id)/components/,73
POST /api/groups/(int:id)/languages/,74
POST /api/groups/(int:id)/projects/,74
POST /api/groups/(int:id)/roles/,73
PUT /api/groups/(int:id)/,73
DELETE /api/groups/(int:id)/,73
DELETE /api/groups/(int:id)/componentlists/(int:component_list_id),74
DELETE /api/groups/(int:id)/components/(int:component_id),73
DELETE /api/groups/(int:id)/languages/(string:language_code),74
DELETE /api/groups/(int:id)/projects/(int:project_id),74
PATCH /api/groups/(int:id)/, 73 /api/languages
GET /api/languages/,75
GET /api/languages/(string:language)/,75
GET /api/languages/(string:language)/statistics/,76
POST /api/languages/,75
PUT /api/languages/(string:language)/,76
DELETE /api/languages/(string:language)/,76
PATCH /api/languages/(string:language)/,76/api/projects
GET /api/projects/,77
GET /api/projects/(string:project)/,77
GET /api/projects/(string:project)/changes/,78
GET /api/projects/(string:project)/components/,79
GET /api/projects/(string:project)/languages/,81
GET /api/projects/(string:project)/repository/,78
GET /api/projects/(string:project)/statistics/,81
POST /api/projects/,77
POST /api/projects/(string:project)/components/,79
POST /api/projects/(string:project)/repository/,78
PUT /api/projects/(string:project)/,78
DELETE /api/projects/(string:project)/,78
PATCH /api/projects/(string:project)/,77/api/roles
GET /api/roles/,74
GET /api/roles/(int:id)/,74
POST /api/roles/,74
PUT /api/roles/(int:id)/,75
DELETE /api/roles/(int:id)/,75
PATCH /api/roles/(int:id)/,75/api/screenshots
GET /api/screenshots/,95
GET /api/screenshots/(int:id)/,95
```

354

```
GET /api/screenshots/(int:id)/file/,95
POST /api/screenshots/,95
POST /api/screenshots/(int:id)/file/,95
POST /api/screenshots/(int:id)/units/,95
PUT /api/screenshots/(int:id)/,96
DELETE /api/screenshots/(int:id)/,96
DELETE /api/screenshots/(int:id)/units/(int:unit_id),95
PATCH /api/screenshots/(int:id)/,96/api/tasks
GET /api/tasks/,101
GET /api/tasks/(str:uuid)/, 101 /api/translations
GET /api/translations/,88
GET /api/translations/(string:project)/(string:component)/(string:language)/,
88
GET /api/translations/(string:project)/(string:component)/(string:language)/changes/,
90
GET /api/translations/(string:project)/(string:component)/(string:language)/file/,
91
GET /api/translations/(string:project)/(string:component)/(string:language)/repository/,
92
GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/,
92
GET /api/translations/(string:project)/(string:component)/(string:language)/units/,
90
POST /api/translations/(string:project)/(string:component)/(string:language)/autotransla
91
POST /api/translations/(string:project)/(string:component)/(string:language)/file/,
91
POST /api/translations/(string:project)/(string:component)/(string:language)/repository/
92
POST /api/translations/(string:project)/(string:component)/(string:language)/units/,
91
DELETE /api/translations/(string:project)/(string:component)/(string:language)/,
90 /api/units
GET /api/units/,93
GET /api/units/(int:id)/,93
PUT /api/units/(int:id)/,93
DELETE /api/units/(int:id)/,94
PATCH /api/units/(int:id)/,93/api/users
GET /api/users/,70
GET /api/users/(str:username)/,70
GET /api/users/(str:username)/notifications/,71
GET /api/users/(str:username)/notifications/(int:subscription_id)/,71
GET /api/users/(str:username)/statistics/,71
POST /api/users/,70
POST /api/users/(str:username)/groups/,71
POST /api/users/(str:username)/notifications/,71
PUT /api/users/(str:username)/,70
PUT /api/users/(str:username)/notifications/(int:subscription_id)/,71
DELETE /api/users/(str:username)/,71
```

```
355
```

```
DELETE /api/users/(str:username)/notifications/(int:subscription_id)/,72
PATCH /api/users/(str:username)/,70
PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 72 /ex-
ports
GET /exports/rss/, 104
GET /exports/rss/(string:project)/,104
GET /exports/rss/(string:project)/(string:component)/,104
GET /exports/rss/(string:project)/(string:component)/(string:language)/,104
GET /exports/rss/language/(string:language)/,104
GET /exports/stats/(string:project)/(string:component)/, 103 /hooks
GET /hooks/update/(string:project)/,101
GET /hooks/update/(string:project)/(string:component)/, 101
POST /hooks/azure/, 102
POST /hooks/bitbucket/,101
POST /hooks/gitea/, 102
POST /hooks/gitee/,102
POST /hooks/github/, 101
POST /hooks/gitlab/, 101
POST /hooks/pagure/, 102
```

.XML resource file file format, 56 --base-file-template TEMPLATE import_project 222222222222,280 --convert wlc ???????????,106 --input wlc ??????????,107 --language-map LANGMAP import_memory 22222222,280 --language-regex REGEX import_project 222222222, 280 --main-component import_project 222222222,280 --name-template TEMPLATE import_project 222222222,280 --new-base-template TEMPLATE import_project 22222222222,280 --no-projects-update setupgroups 22222222,284 --overwrite auto_translate 2222222222, 277 wlc 22222222222, 107 --source PROJECT/COMPONENT auto_translate 222222222,277 --update createadmin ???????????, 279 in-

```
357
```

add_suggestions weblate admin command, 276 add_suggestions 2222222222 -- author USER@EXAMPLE.COM, 276 ADMINS setting, 143 AKISMET_API_KEY setting, 239 ALLOWED_HOSTS setting, 143 Android file format, 52 ANONYMOUS_USER_NAME setting, 239 API, 67, 104, 108 Apple strings file format, 53 ARB file format, 55 AUDITLOG_EXPIRY setting, 239 AUTH_LOCK_ATTEMPTS setting, 239 AUTH_TOKEN_VALID setting, 240 auto_translate weblate admin command, 277 auto_translate 222222222222 -- add, 277 -- inconsistent, 277 -- mt MT, 277 -- overwrite, 277 -- source PROJECT/COMPONENT, 277 -- threshold THRESHOLD, 277 -- user USERNAME, 277 AUTO_UPDATE setting, 239 AUTOFIX_LIST setting, 240 AVATAR_URL_PREFIX setting, 240 BASE_DIR setting, 240 BaseAddon (weblate.addons.base ?????), 310 BASIC_LANGUAGES setting, 241 bilingual translation, 46 can_install() (weblate.addons.base.BaseAddon ?????????), 310 celery_queues weblate admin command, 277 changes wlc ???????????,106 CHECK LIST setting, 241 checkgit weblate admin command, 277 cleanup wlc ??????????,106 cleanuptrans weblate admin command, 278 Comma separated values file format, 56 Command (*wlc.main* 2222), 109 COMMENT_CLEANUP_DAYS setting, 242 commit wlc ??????????, 106 commit_pending weblate admin command, 278 COMMIT_PENDING_HOURS setting, 242 commitgit weblate admin command, 277 configure () (weblate.addons.base.BaseAddon 22.2.2.2), 310 createadmin weblate admin command, 278 278 -- password PASSWORD, 278 -- update, 278 -- username USERNAME, 278 CSP_CONNECT_SRC setting, 241 CSP_FONT_SRC setting, 241 CSP_IMG_SRC setting, 241 CSP_SCRIPT_SRC setting, 241

CSP_STYLE_SRC setting, 241 CSV file format, 56 daily() (weblate.addons.base.BaseAddon ???????), 310 DATA_DIR setting, 242 DATABASE_BACKUP setting, 242 DATABASES setting, 143 DEBUG setting, 144 DEFAULT_ACCESS_CONTROL setting, 242 DEFAULT_ADD_MESSAGE setting, 243 DEFAULT_ADDON_MESSAGE setting, 243 DEFAULT_ADDONS setting, 243 DEFAULT_COMMIT_MESSAGE setting, 243 DEFAULT_COMMITER_EMAIL setting, 243 DEFAULT_COMMITER_NAME setting, 244 DEFAULT_DELETE_MESSAGE setting, 243 DEFAULT_FROM_EMAIL setting, 144 DEFAULT_LANGUAGE setting, 244 DEFAULT_MERGE_MESSAGE setting, 243 DEFAULT MERGE STYLE setting, 244 DEFAULT_PULL_MESSAGE setting, 244 DEFAULT_RESTRICTED_COMPONENT setting, 243 DEFAULT TRANSLATION PROPAGATION setting, 244 download wlc ?????????, 106 DTD file format, 58 dump_memory weblate admin command, 278 dumpuserdata weblate admin command, 279

ENABLE_AVATARS setting, 244

ENABLE_HOOKS setting, 245

 $ENABLE_HTTPS$ setting, 245

ENABLE_SHARING setting, 245

file format .XML resource file, 56 Android, 52 Apple strings, 53 ARB, 55 Comma separated values, 56 CSV, 56 DTD, 58 gettext, 47 go-i18n, 55 GWT properties, 50 i18next, 54 INI translations, 51 Java properties, 50 Joomla translations, 51 JSON, 53 PHP strings, 53 PO, 47 Qt, 52 RC, 59 RESX, 56 Ruby YAML, 57 Ruby YAML Ain't Markup Language, 57 string resources, 52 TS, 52 XLIFF, 49 XML, 58 YAML, 57 YAML Ain't Markup Language, 57

get () (wlc. Weblate ??????), 108

get_add_form() (weblate.addons.base.BaseAddon ??????????), 310

get_settings_form() (weblate.addons.base.BaseAddon 22.2.2.2), 310

gettext file format, 47

GITHUB CREDENTIALS setting, 246

GITHUB_TOKEN setting, 246

GITHUB_USERNAME setting, 246

GITLAB_CREDENTIALS setting, 245

GITLAB_TOKEN setting, 246

GITLAB_USERNAME setting, 245

go-i18n file format, 55

GOOGLE_ANALYTICS_ID setting, 246

GWT properties file format, 50

HIDE_REPO_CREDENTIALS setting, 247 HIDE_VERSION setting, 247 i18next file format, 54 import_demo weblate admin command, 279 import_json weblate admin command, 279 import_json 2222222222 -- ignore, 279 -- main-component COMPONENT, 279 -- project PROJECT, 279 -- update, 279 import_memory weblate admin command, 280 import_project weblate admin command, 280 import_project 2222222222 -- base-file-template TEMPLATE, 280 -- file-format FORMAT, 280 -- language-regex REGEX, 280 -- license NAME, 280 -- license-url URL, 280 -- main-component, 280 -- name-template TEMPLATE, 280 -- new-base-template TEMPLATE, 280 --vcs NAME, 280 importuserdata weblate admin command, 282 importusers weblate admin command, 282 importusers ???????????? -- check, 282 INI translations file format, 51 install_addon weblate admin command, 282 install_addon 2222222222 -- addon ADDON, 282 -- configuration CONFIG, 282 -update, 282 IP_BEHIND_REVERSE_PROXY setting, 247 IP_PROXY_HEADER setting, 247 IP_PROXY_OFFSET setting, 247 iPad translation, 53 iPhone translation, 53 Java properties file format, 50 Joomla translations file format, 51 JSON file format, 53 LEGAL_URL setting, 248 LICENSE EXTRA setting, 248 LICENSE_FILTER setting, 248 LICENSE REQUIRED setting, 248 LIMIT TRANSLATION LENGTH BY SOURCE LENGTH setting, 249 list_languages weblate admin command, 282 list_translators weblate admin command, 282 list_translators ?????????? -- language-code, 282 list_versions weblate admin command, 283 list-components wlc ???????????,106 list-projects wlc ????????????,106 load() (wlc.config. WeblateConfig ??????), 109 loadpoweblate admin command, 283 loadpo ????????????? -- force, 283 -- lang LANGUAGE, 283 LOCALIZE_CDN_PATH setting, 249 LOCALIZE CDN URL setting, 249 lock wlc ??????????,106

```
360
```

lock_translation weblate admin command, 283 lock-status wlc ??????????,106 LOGIN_REQUIRED_URLS setting, 249 LOGIN_REQUIRED_URLS_EXCEPTIONS setting, 249 lswlc ??????????,106 MACHINE_TRANSLATION_SERVICES setting, 250 main() (wlc.main 22222), 109 MATOMO_SITE_ID setting, 250 MATOMO_URL setting, 250 monolingual translation, 46 move_language weblate admin command, 283 MT_APERTIUM_APY setting, 250 MT_AWS_ACCESS_KEY_ID setting, 251 MT_AWS_REGION setting, 251 MT_AWS_SECRET_ACCESS_KEY setting, 251 MT_BAIDU_ID setting, 251 MT_BAIDU_SECRET setting, 251 MT_DEEPL_API_VERSION setting, 251 MT DEEPL KEY setting, 251 MT_GOOGLE_CREDENTIALS setting, 252 MT_GOOGLE_KEY setting, 252 MT_GOOGLE_LOCATION setting, 252 MT_GOOGLE_PROJECT setting, 252 MT MICROSOFT BASE URL setting, 252 MT_MICROSOFT_COGNITIVE_KEY setting, 252 MT_MICROSOFT_ENDPOINT_URL setting, 253 MT_MICROSOFT_REGION setting, 252 MT_MODERNMT_KEY setting, 253 MT_MODERNMT_URL setting, 253 MT_MYMEMORY_EMAIL setting, 253 MT_MYMEMORY_KEY setting, 253 MT_MYMEMORY_USER setting, 253 MT_NETEASE_KEY setting, 253 MT_NETEASE_SECRET setting, 253 MT_SAP_BASE_URL setting, 254 MT_SAP_PASSWORD setting, 255 MT_SAP_SANDBOX_APIKEY setting, 254 MT_SAP_USE_MT setting, 255 MT_SAP_USERNAME setting, 254 MT_SERVICES setting, 250 MT_TMSERVER setting, 254 MT_YANDEX_KEY setting, 254 MT_YOUDAO_ID setting, 254 MT_YOUDAO_SECRET setting, 254 NEARBY_MESSAGES setting, 255 PAGURE_CREDENTIALS setting, 255 PAGURE_TOKEN setting, 255