WEB-T Universal plugin and Translation Hub

Contents

Introduction 5
How it works 5
Translation Hub setup 6
Minimal system requirements 6
Supported software versions 6
Kubernetes cluster 6
Running WEB-T Translation Hub locally in a dev environment 6
MySQL Databases 7
Setup WEB-T databases on MySQL 8.0 server 7
Prepare empty WEB-T databases (DB) and users for databases: 7
Build WEB-T container images and push containers to your container registry ... 7
Installing the chart 8
Using the chart from GitHub repo 8
Uninstalling the chart 11
Translation Hub overview 12
Configure the translation provider 13
Website translator integration 14
My integrations 15
Integration (Basic) 16
Create new integration. 16
Include the generated JavaScript code in your website 17
Try an example 18
Translate the example page 18
Integration (Advanced) 19
Set up language selector 19
Customise the translation progress bar 21
Integrate into an already multilingual website 22
Full configuration options list 23
Edit existing integration 27
Delete integration 28
Manage translations 29
Edit translation 29

Confirm translation 30
Delete translations 31
Delete translations having a certain target language 31
Delete translations having a certain URL 31
Delete all translations 32
Filter translations 32
Filter by status or translation type 32
Filter by content type 32
Filter by text 32
Filter by URL 33
SEO tags 33
FAQ 34
How do I integrate Website translator into my website? 34
Can I customise the language selector look? 34
Do I have to use the default language selector? 34
Can I customise the translation notice? 34
Can I use machine translation alongside existing human translation? 34
I want to keep some parts untranslated / Some of my content is in another language. 35
I want to explicitly mark the content that should be translated. 35
I want the translation progress bar to not be visible to website visitors. 36
How can I edit the translations? 36
Does the plugin translate images and media content? 36
Will the edit I made in one page affect the translations of the whole website? 36
Will the translation edits I made in my test environment be used in production? 37
I want to add or change languages, or MT engines. 37
How to update translations after changing translation engine? 37
Troubleshooting 38
I can’t find some of my page content in the Translation editor 38
How to enable debugging for the plugin? 38
The language selector is not displayed 38
My website is not being translated 38
Translation doesn’t work with eTranslation provider 38
Language of the Website translator’s UI keeps changing 39

Website translation is slow / Translation makes the page load slower 39
Translation timed out 39
The Website translator doesn’t look right 39

Introduction

WEB-T Universal is solution for website content translation. It consists of two parts: a client-side plugin that is embedded in your website, and a server-side Translation Hub that provides translation caching and post-editing.

How it works

The plugin translates the currently rendered webpage content into the selected language. All translations are saved into a database, allowing you to edit them via Translation editor on the Translation Hub.

Set up the Translation Hub:

1. Set up and prepare the Kubernetes environment.
2. Download deployment HELM and configure values.
3. Deploy.
4. Make available to the outside.

Obtain eTranslation or WEB-T compatible translation provider authorisation information:

- Get MT provider access.
- Use this access information on the Translation Hub.

Create an integration and embed the JavaScript plugin on your website:

1. Create a Website Translator integration on the Translation Hub.
2. Embed the generated JavaScript code in your website.

Translate your website and improve the automated translations:

1. Translate your website by going through all pages and selecting every target language from the menu.
2. Use Translation editor to improve translated texts. The corrections will instantly appear on your website.
3. When new content is added in the original language, re-translate the edited page by selecting every target language from the menu.

The first time the content of any page is loaded it will take some time to translate it. The translations will be saved into a database and will load faster the next time.

Please note: Websites using WEB-T plugin with eTranslation integration enabled must be published and accessible online from eTranslation service. Translation with eTranslation will not work otherwise.

Translation Hub setup

The toolkit consists of several microservices; each component is deployable as a docker container.

Components:

- Frontend (frontend) – Application responsible for all GUI and its operations. It uses Angular, which means this is a Single Page Application
- Translation Hub API (websiteTranslationService) – service for translation management
- Configuration service (configurationService) – service for configurations
- MT API Integrator (mtApiIntegrator) – service for MT services
- eTranslation Provider (eTranslationProxy) – service for eTranslation implementation
- Redis – DB for eTranslation
- MySQL – DB for APIs

Minimal system requirements

- At least a 2-core x86/x64 CPU (Central Processor Unit)
- 5 GB of RAM (Random Access Memory) or more
- 30 GB of storage or more (for DB)
- Internet access

Supported software versions

- Kubernetes cluster version 1.25+
- Helm 3.5+
- MySQL server 8.0

Kubernetes cluster

You will need kubectl (instructions) and Helm (instructions) installed.

Running WEB-T Translation Hub locally in a dev environment

For testing and development purposes, an easy option to get Kubernetes running is to install Rancher Desktop. Once you have it installed, you will also need to set up an ingress controller. Rancher uses *Traefik* as the default one, so disable it first by unchecking Enable Traefik from the Kubernetes Settings page. Then deploy the NGINX ingress controller:

```
helm upgrade --install ingress-nginx ingress-nginx \
 --repo https://kubernetes.github.io/ingress-nginx \
 --namespace ingress-nginx --create-namespace
```

MySQL Databases

Setup WEB-T databases on MySQL 8.0 server

Please note: use separate on-premise or cloud hosted MySQL server or run MySQL server in Kubernetes. MySQL server must be reachable from Kubernetes Pods

Prepare empty WEB-T databases (DB) and users for databases:

Adjust DB, DB usernames and DB user passwords as needed; you must match the database names in your Helm chart values.yaml file.

```
-- For configurationService component

CREATE DATABASE IF NOT EXISTS `webt_configuration` CHARACTER SET utf8 COLLATE utf8_unicode_ci;

CREATE USER IF NOT EXISTS 'webt_configuration_user'@'%' IDENTIFIED WITH mysql_native_password BY 'webt_configuration_user_pass';

GRANT ALL PRIVILEGES ON `webt_configuration`.* TO 'webt_configuration_user'@'%';


-- For websiteTranslationService component

CREATE DATABASE IF NOT EXISTS `webt_website_translation` CHARACTER SET utf8 COLLATE utf8_unicode_ci;

CREATE USER IF NOT EXISTS 'webt_website_translation_user'@'%' IDENTIFIED WITH mysql_native_password BY 'webt_website_translation_user_pass';

GRANT ALL PRIVILEGES ON `webt_website_translation`.* TO 'webt_website_translation_user'@'%';
```

Build WEB-T container images and push containers to your container registry

Read about building docker container images here

Component	**Source code repository**	**Corresponding Helm chart yaml**
configurationService	UP-TranslationHub-ConfigurationService	configuration-service.yaml
eTranslationProxy	UP-TranslationHub-eTranslationProvider	translation-proxy-e-translation.yaml
frontend	UP-TranslationHub-Frontend	frontend.yaml

mtApiIntegrator	UP-TranslationHub-MT-API-Integrator	mt-api-integrator.yaml
websiteTranslationService	UP-TranslationHub-API	website-translation-service.yaml

Installing the chart

Using the chart from GitHub repo

1. Clone this repository

```
git clone https://code.europa.eu/web-t/web-t-translation-hub-containerable/deployment

cd deployment/translation-hub
```

2. Make sure to update **values.yaml** with corresponding values or create new values file.

The following table lists the configurable parameters of the Translation Hub chart. The current default values can be found in the **values.yaml** file.

Parameters	**Description**
basicAuth.enabled	Enable use of basic authentication for the frontend access – set "true" or "false"
basicAuth.credentials	[Optional] Set credentials for basic authentication
configurationService.dbConnectionString	Configuration Service Database connection string
configurationService.image.name	Configuration Service Image name
configurationService.image.pullSecret	Configuration Service Image pull secret

configurationService.image.repository	Configuration Service Image repository
configurationService.image.tag	Configuration Service Image tag
eTranslationProxy.image.name	ETranslation Proxy Image name
eTranslationProxy.image.pullSecret	ETranslation Proxy Image pull secret
eTranslationProxy.image.repository	ETranslation Proxy Image repository
eTranslationProxy.image.tag	ETranslation Proxy Image tag
frontend.image.name	Frontend Image name
frontend.image.pullSecret	Frontend Image pull secret
frontend.image.repository	Frontend Image repository
frontend.image.tag	Frontend Image tag
global.domain	Translation Hub domain name
global.eTranslation.email	eTranslation translation recipient email – for testing only, leave empty after testing
global.eTranslation.timeout	eTranslation timeout
ingress.tlsSecret	Ingress TLS secret name
mtApiIntegrator.image.name	MT API Integrator Image name
mtApiIntegrator.image.pullSecret	MT API Integrator Image pull secret

mtApiIntegrator.image.repository	MT API Integrator Image repository
mtApiIntegrator.image.tag	MT API Integrator Image tag
websiteTranslationService.dbConnectionString	Website translation service Database connection string
websiteTranslationService.image.name	Website translation service Image name
websiteTranslationService.image.pullSecret	Website translation service Image pull secret
websiteTranslationService.image.repository	Website translation service Image repository
websiteTranslationService.image.tag	Website translation service Image tag

Optional: Configuration for Restricting Access to translation hub website (Basic Authentication):

Credential preparation for Basic Authentication configuration:
If **basicAuth.enabled** is set to **true** then user credentials must be configured in **values.yaml** file.
Example of how to prepare Basic Auth configuration for **web-t-user** user with password **web-t-pwd**:

```
#  In Linux terminal execute command:

htpasswd -nb web-t-user web-t-pwd
```

Command output example:

```
web-t-user:$apr1$j2PQtFMu$7uEKTHLBdxwAbYKvgdfmh/
```

```
# Next, copy part of previous output – the part after username and colon symbol (e.g., part after "web-t-user:") and paste that to second command to get the base64-encoded password hash to use as a password:

printf '$apr1$j2PQtFMu$7uEKTHLBdxwAbYKvgdfmh/' | base64
```

Command output example, use this for password configuration:
JGFwcjEkajJQUXRGTXUkN3VFS1RITEJkeHdBYllLdmdkZm1oLw==

Update configuration in **values.yaml** file:

```
basicAuth:

  enabled: true

  credentials:

    - username: web-t-user

      password: JGFwcjEkajJQUXRGTXUkN3VFS1RITEJkeHdBYllLdmdkZm1oLw==

   # you can add more users if necessary by adding these lines with user prepared
user credentials:

    - username: user2      # to add another user

      password: *pwd2*  # tanother users pwd
```

3. Install the chart with the release name **translation-hub in web-t** namespace.

```
helm install --create-namespace -n web-t translation-hub . -f values.yaml
```

Please note: eTranslation is an online machine translation service provided by the European Commission (EC). eTranslation web service is asynchronous. This means that the client sends a translation request and is notified later when the text snippet is translated. Make sure that the translation hub API domain is correctly exposed to the internet and can receive traffic from eTranslation servers.

Uninstalling the chart

To uninstall/delete the translation-hub deployment from web-t namespace, run:

```
helm delete translation-hub --namespace web-t
```

Delete web-t namespace

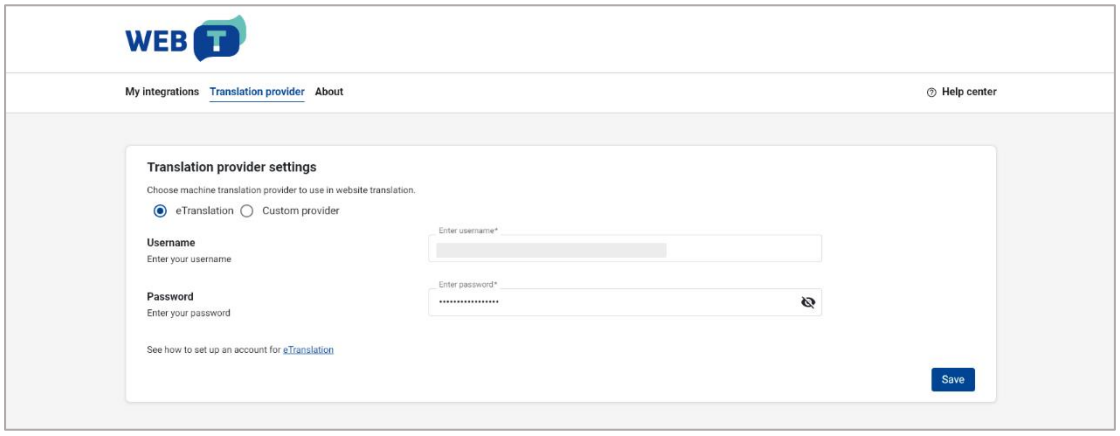
```
kubectl delete namespace web-t
```

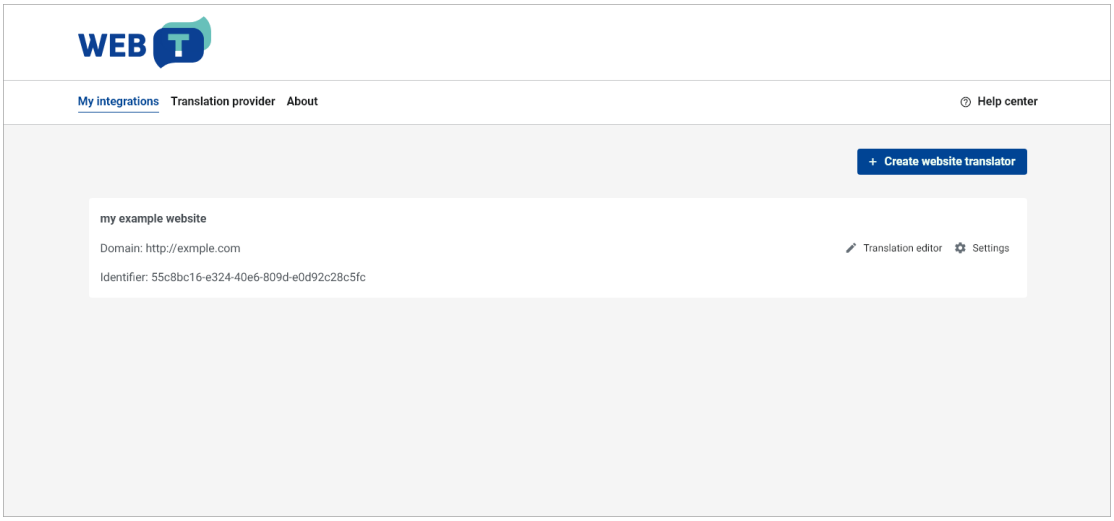
Translation Hub overview

After you have set up the **Translation Hub**, open the browser and navigate to the URL where it is located.

On the **Translation Hub**, you will find:

- **Translation Provider** – for setting the translation provider authorisation information (eTranslation API credentials or Base URL & API key).
- **My integrations** – for creating and managing your website integrations. Once you create an integration it appears on the "**My integrations**" list.
- **Translation editor** – for managing and editing translations (for each integration separately).
- **Settings** – for editing languages, machine translation engines and domain access (for each integration separately). This page also includes the generated JavaScript integration code.



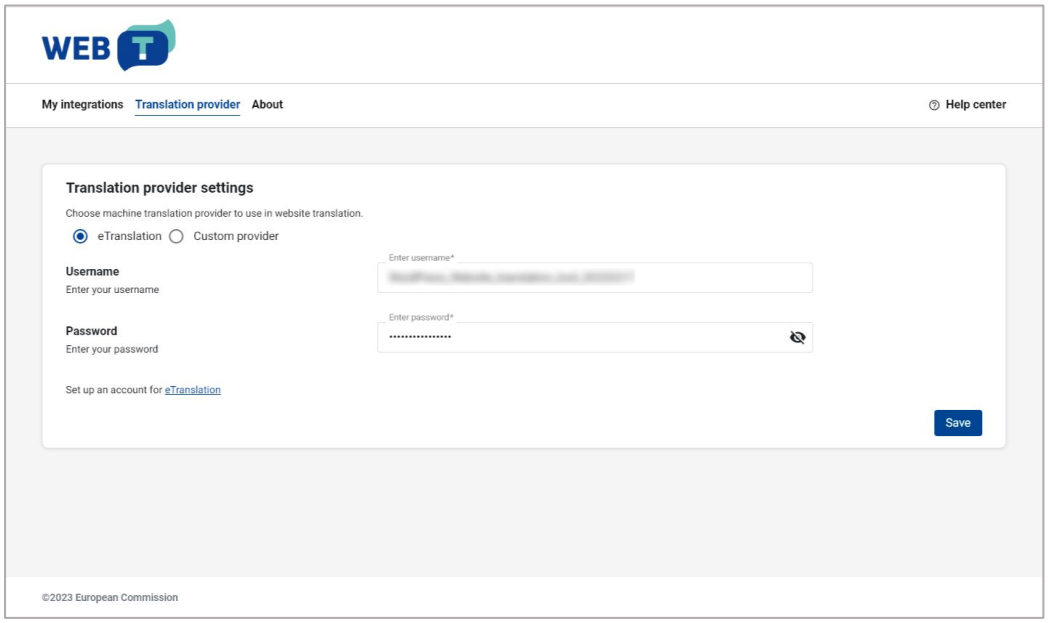


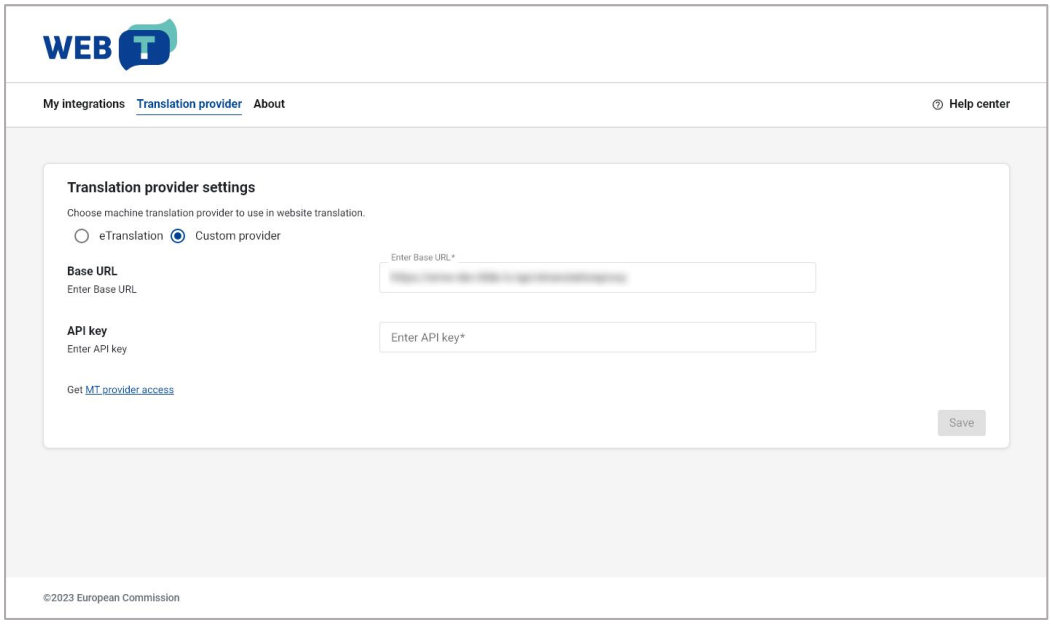
Configure the translation provider

Before you continue:
You must have an eTranslation or WEB-T compatible translation provider account.
You have obtained the eTranslation API credentials or Base URL & API key from your translation provider.

To configure the **MT Provider** on the **Translation Hub**:

1. On the Translation Hub, open **Translation provider**.
2. Select MT provider (eTranslation or Custom).
3. Enter the required data (eTranslation API credentials or Base URL & API key).
4. Click **Save**.





Website translator integration

Before you continue:
You have deployed the Translation Hub and have access to it.
You have configured the Translation provider.

Integration consists of 2 main steps:

1. **Creating an integration** on the **Translation Hub**.
2. **Including** the **generated JavaScript code** in your website.

After you have included the JavaScript code, the language selector will appear on your website, and you can start translating by selecting a language.

💡 For **most users** we recommend following the steps in **Integration (Basic)**.
For more **advanced** use cases, see **Integration (Advanced):**

- If you wish to display languages as buttons, see Configure the default language selector.
- If you wish to use a custom language selector, see Use a custom language selector.
- If your website is already multilingual, see Integrate into an already multilingual website.

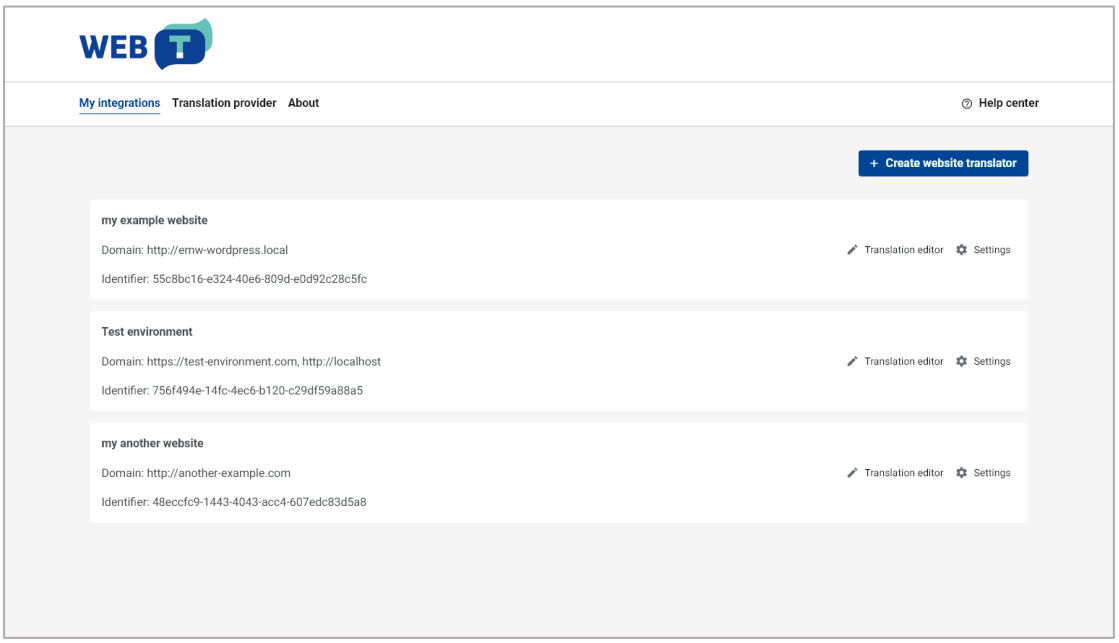
In this section:

- My integrations
- Integration (Basic)
  - Create new integration
  - Include the generated JavaScript code in your website
  - Try an example
  - Translate the example page
- Integration (Advanced)
  - Set up language selector
    - Configure the default language selector
    - Use a custom language selector
  - Customise the translation progress bar
    - Change display position
    - Hide the translation progress bar
  - Integrate into an already multilingual website
  - Full configuration options list
- Edit existing integration
- Delete integration

My integrations

“My integrations” lists all of the website integrations you have created on the Translation Hub. From this view you can:

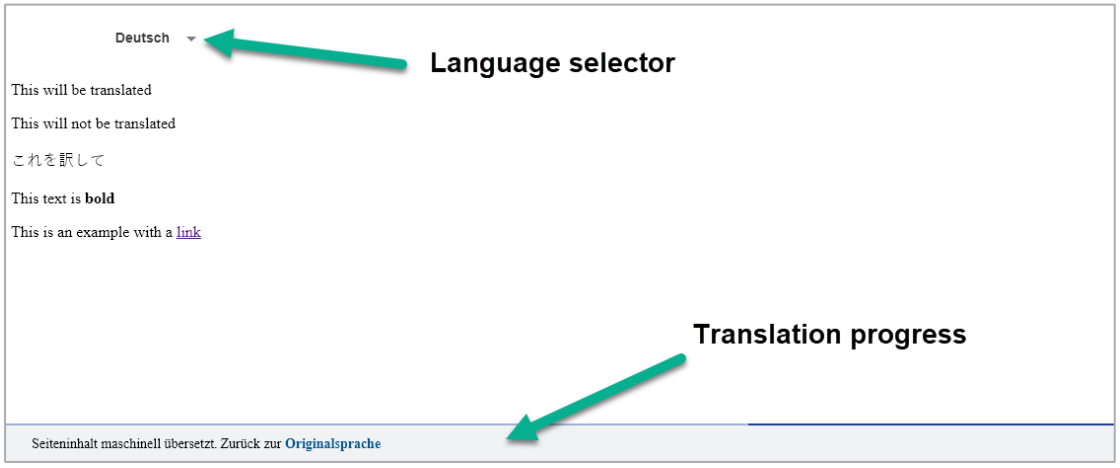
- create new integrations,
- access the Translation editor for a specific integration,
- access the Settings of a specific integration,
- delete integration by opening the Settings of a specific integration.



Integration (Basic)

Following the basic integration steps, you will add the default language selector to your website. This section includes an HTML example that you can try.

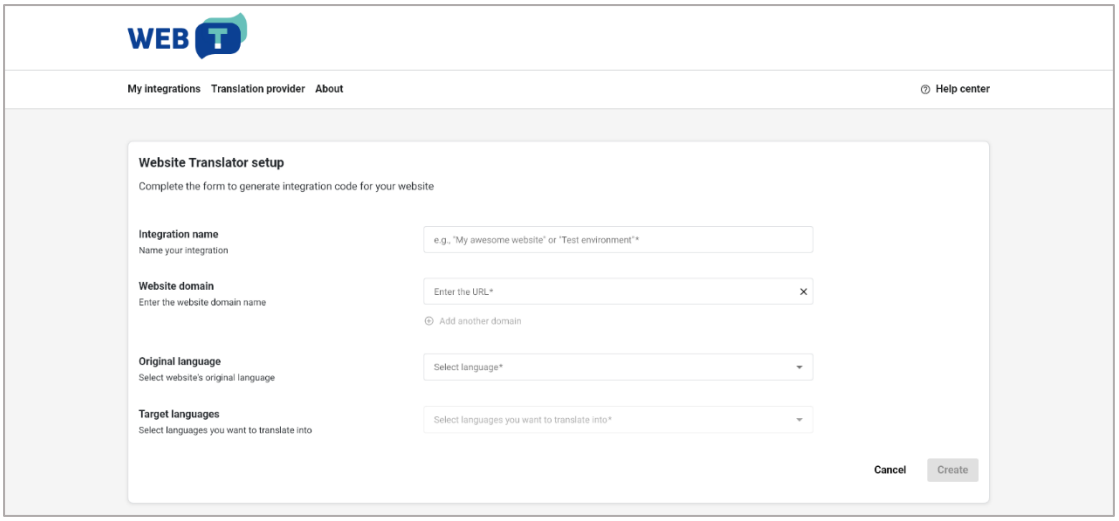
The language selector will appear as a dropdown. It will load the available languages automatically and start translation on selecting. The translations will appear on the **Translation Hub** in **Translation editor** where you can edit them.

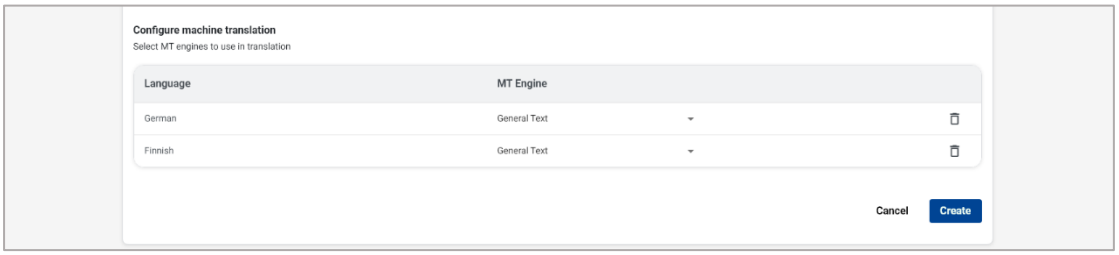


Create new integration

To create a new website integration:

1. On the **Translation Hub**, open **My integrations**.
2. Click **+Create website translator**.
3. Select **General setup**.
4. Fill in the form:
   4.1. Give your integration a name.
   4.2. Add your website domain URL. (https://example.com, https://www.example.com and https://sub.example.com should all be added separately by clicking **Add another domain**).
   4.3. Select your website's original language.
   4.4. Select the languages you want your website to be translated into.
   4.5. In the generated table, select an MT engine for each language.
5. Click **Save**.

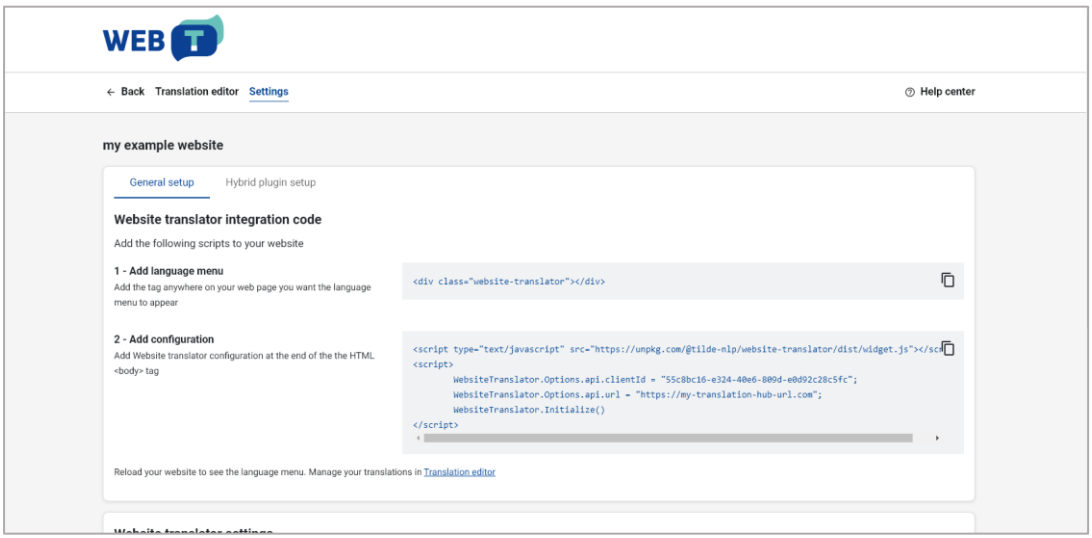




Include the generated JavaScript code in your website

To add the JavaScript plugin to your website:

1. On the **Translation Hub**, open the **Settings** of the specific integration.
2. Open the **General setup** tab.
3. Copy the language selector tag and insert it in your website code, where you want the language selector to appear.
4. Copy the website translation widget script and insert it at the end of the <body> tag.
5. The language selector should appear on your webpage. Reload the page if needed.



Try an example

To create and translate a test page:

1. Copy the example below.
2. Replace the WebsiteTranslator.Options.api.clientId value from the generated code.
3. Replace the WebsiteTranslator.Options.api.url value to your Translation Hub URL.
4. Save the file to your web solution.
5. Open the example page in your web browser.

```
<html lang="en">
<head>
   <title>My title</title>
   <meta property="og:site_name" content="My site name">
   <meta name="description" content="Some site description.">
</head>
<body>
   <div class="website-translator"></div>
   <p>This will be translated</p>
   <p translate="no">This will not be translated</p>
   <p lang='ja'>これを訳して</p>
   <p>This text is <b>bold</b></p>
   <p>This is an example with a <a href="">link</a></p>
</body>
<footer>
   <script      type="text/javascript"      src="https://unpkg.com/@tilde-
nlp/website-translator/dist/widget.js"></script>
   <script>
      // 👇 Change XXXXXXXXXX to your Client-ID
      WebsiteTranslator.Options.api.clientId = "XXXXXXXXXX";
      // 👇 Change XXXXXXXXXX to your Translation Hub URL
      WebsiteTranslator.Options.api.url = "XXXXXXXXXX";
      WebsiteTranslator.Initialize()
   </script>
</footer>
</html>
```

Translate the example page

Translated text segments get stored in a database. This means that translating the same page again will be a lot quicker and you can edit translations.

The translations will appear on the **Translation Hub** in **Translation editor** where you can edit them.

Integration (Advanced)

Before you continue:
You will need to have some JavaScript knowledge to follow the Advanced integration instructions.
You will need to have access to edit the JavaScript integration embedded into your website.

In this section:

- Set up language selector
  - Configure the default language selector
  - Use a custom language selector
- Customise the translation progress bar
  - Change display position
  - Hide the translation progress bar
- Integrate into an already multilingual website
- Full configuration options list

Set up language selector

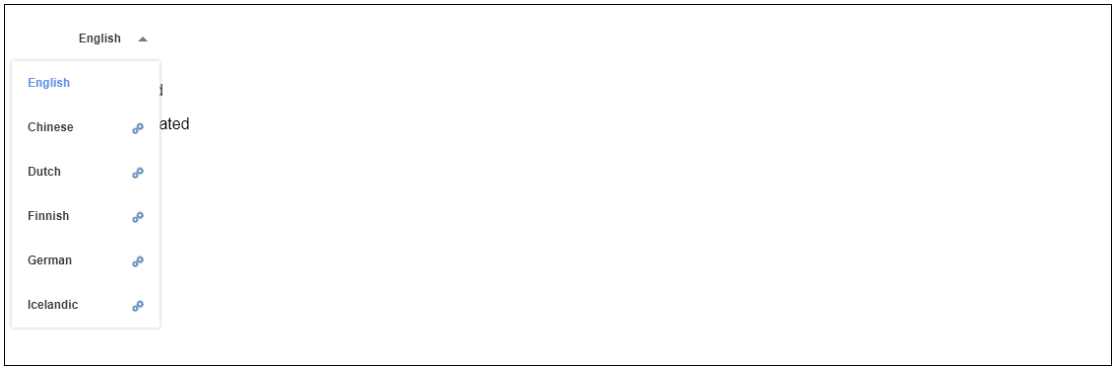
Configure the default language selector

Before you continue:
You have followed the steps of Basic integration. The JavaScript plugin is embedded into your website.

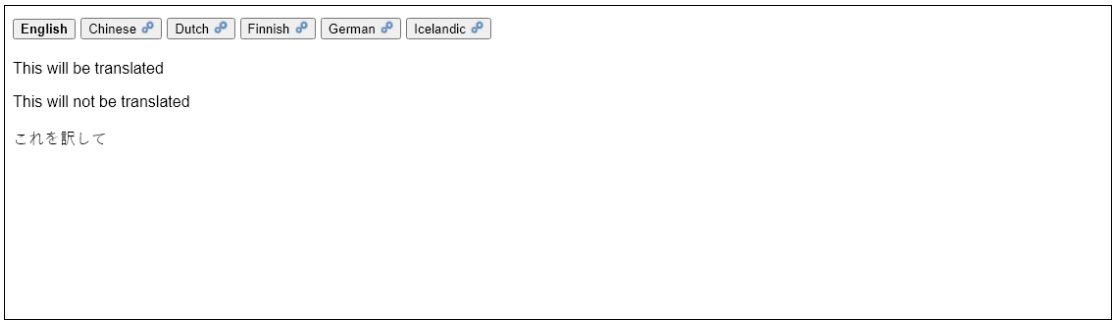
Language selector can be displayed as a dropdown or a list of buttons that can be styled by CSS to further match your style. It will load the available languages automatically and start translation on selecting.

Add an element with **class="website-translator"** to your page. See: Try an .

Setting **WebsiteTranslator.Options.ui.layout="menu"** will display the language selector as a dropdown menu. It is the default appearance.



The website translator's language selector can be displayed as a list of buttons setting **WebsiteTranslator.Options.ui.layout="list".**



That can be further styled by overwriting CSS as you wish:

Use a custom language selector

Before you continue:
You have created an integration on the Translation Hub.
The default language selector (dropdown or list of buttons) doesn't fit your needs.

Initialise

Initialises Website Translator – language select, fetches available MT systems. If there is lang parameter in URL, it will translate to the language that is provided in the parameter value. If there is no lang parameter and the current source language is the same as the last time Tilde Website Translator was used, then it will translate to the last translated target language.

Normally Initialise should be issued only once, but further calls will reinitialise language select. Consider this when your page changes DOM including language selector.

```
// Configure your personal access key
WebsiteTranslator.Options.api.clientId = 'x-xxxxxxxx-xxxx-xx'

// Initialize
await WebsiteTranslator.Initialize()
```

Start translation and wait until finished

Translate page to specific language. If you call Translate even if the previous translation is not finished, it will cancel the current translation, restore the web page in the original language and start a new translation in the specified language – same as to issue Cancel and restore translation and start new translation with Translate.

```
// You can call the next translation even when the previous one is not
finished
WebsiteTranslator.Translate("lt").then(function(translationFinish){
   // You can wait when page is translated
   Promise.all(translationFinish).then(function(){
      console.log("Translation is complete")
   })
})
```

When the page is translated, the web page url lang parameter is changed to translated language or added if it does not exist.

For example: https://example.com/ -> https://example.com/?lang=lt If page translation is restored, then the lang parameter is set to the source language.

Cancel translation

Restores the page in the Source language.

```
WebsiteTranslator.CancelAndRestore();
```

Translate with the given language

```
WebsiteTranslator.Translate("hu");
```

Get the current language

Returns the language code for the web page. Value can also be SourceLanguage. This can be used to programmatically read the web page language.

```
WebsiteTranslator.CurrentLanguage
```

Get the available languages and machine translation systems

Returns available languages and systems. Can be used to create the custom language selector or language selection using JavaScript translation API.

```
WebsiteTranslator.GetTargetLanguages()
```

Customise the translation progress bar

Translation progress bar can be displayed at the top or bottom of the page or hidden completely. You will need to edit the JavaScript integration code on your website to change it.

Change display position

To change the display position, you will have to set it from JavaScript.

Add the following line to the integration code:

```
WebsiteTranslator.Options.ui.toolbarPosition = “top”
```

or

```
WebsiteTranslator.Options.ui.toolbarPosition = “bottom”
```

Hide the translation progress bar

Website visitors won’t see the translation progress bar and be able to revert the page to the original language by clicking the restore link.

Add the following line to the integration code:

```
WebsiteTranslator.Options.ui.headless=true
```

Integrate into an already multilingual website

Use machine translation to make an already multilingual website available in more languages.

Specify the languages of your webpage that already have the translation embedded and for which machine translation should not be carried out. Configure the *thirdPartyTranslationLanguages* option, if we have German and French versions of the English pages:

```
WebsiteTranslator.Options
  .translation.thirdPartyTranslationLanguages = ['de', 'fr']
```

Full example:

```
<!DOCTYPE html>
<head>
   <script      type="text/javascript"      src="https://unpkg.com/@tilde-
nlp/website-translator/dist/widget.js"></script>
</head>
<body>
    <div class="website-translator"></div>
    <p>This is an example</p>
</body>

<footer>
    <script>
      // 👇 Change XXXXXXXXXX to your Client-ID
      WebsiteTranslator.Options.api.clientId = "XXXXXXXXXX";
     // 👇 Change XXXXXXXXXX to your Translation Hub URL
      WebsiteTranslator.Options.api.url = "XXXXXXXXXX";

// Specify the embedded languages of your webpage that already has a
translation
        var thirdPartyLanguages = ["de", "fr", "sv"];

WebsiteTranslator.Options.translation.thirdPartyTranslationLanguages =
thirdPartyLanguages;

        // Set the language of the webpage that is currently rendered.
Example shows how to get the language code from the document lang
attribute.
        WebsiteTranslator.Options.currentLanguage =
document.documentElement.getAttribute('lang')
```

```
        // Add custom logic to use embedded translations instead of
machine translation
        WebsiteTranslator.Options.translation.onLanguageSelected =
function (selectedLanguage) {
            return new Promise(function (resolve) {
                let translationHandled = false
                console.info('Language selected: ' + selectedLanguage)
                // Check if custom action must be performed to open the
embedded translation, bypassing the machine translation process
                if
(WebsiteTranslator.Options.translation.thirdPartyTranslationLanguages.inc
ludes(selectedLanguage)) {
                    translationHandled = true
                    if (WebsiteTranslator.Options.currentLanguage !==
selectedLanguage) {
                        console.info('Redirecting to embedded language
version of the page')
                        // Add your own logic on how to transform the URL
of the current language to the URL of the selected target language
                        // or other necessary actions.
                        // Note. This replacement logic is to transform
http://example.com/page_en.html to http://example.com/page_de.html
                        // and vice versa.
                        window.location.href =
window.location.href.replace(/page_\w+/, 'page_' + selectedLanguage)
                        // For example, for page
https://example.com/en/news to transform to https://example.com/de/news
                        // window.location.href =
window.location.href.replace(/example.com\/\w+/, "/example.com/" +
selectedLanguage)
                    }
                }
                resolve(translationHandled)
            })
        }

        // Initialize and run translations
        WebsiteTranslator.Initialize()
    </script>
</footer>
</html>
```

Full configuration options list

Section contains a full list of the plugin's configuration options, that you can set through JavaScript.

Language

The widget determines the target language automatically if the webpage uses lang parameter in URL query.
It supports ISO 639-1 language codes with optional ISO 3166-1 alpha-2 country codes. If the language is not found with the specified country code, then the widget checks if the language without the country code is available.

WebsiteTranslator.Options.currentLanguage

Type: String

Default: null

Current web page language (if the web page is already pre-translated). The language in which the website has been translated. When currentLanguage is not equal to the source language, selecting the language to translate, Website Translator will try to get the page in the language by calling translation.onLanguageSelected(language)

Debug mode

WebsiteTranslator.Options.debug

Type: Boolean
Default: false

Enable verbose logging in console.

Translation API options

WebsiteTranslator.Options.api.clientId

Type: String
Default: null

Authorisation key which allows the Website Translator to access the translation API Required.

Translation options

WebsiteTranslator.Options.translation.autoTranslate

Type: Boolean
Default: true

Save the last translation target language and on the next page load, automatically translate to it.
If the browser url contains lang parameter in the url, Website Translator will use it for translation. Otherwise, if lang parameter is not available, the saved language will be used.

WebsiteTranslator.Options.translation.translateOnlyAllowedTags

Type: Boolean
Default: false

Translate only tags and their children which have translate="yes" attribute.

WebsiteTranslator.Options.translation.translateAttributes

Type: Boolean
Default: true

Translate HTML attributes.

WebsiteTranslator.Options.translation.onLanguageSelected

Type: Function
Default: () => Promise.resolve(false)

Custom actions on selecting languages. If implemented, this function should return Boolean -> true when the Language must be Translated with Website Translator; false otherwise.

WebsiteTranslator.Options.translation.thirdPartyTranslationLanguages

Type: Array<string>
Default: []

Define languages which are supposed to be translated with a third party.

UI options

WebsiteTranslator.Options.ui.headless

Type: Boolean
Default: false

Hide MT toolbar with translation progress.

WebsiteTranslator.Options.ui.tooltipShowDelay

Type: Number
Default: 500

How much time to wait after the user has started to hover translation to show suggestion popup (in ms).

WebsiteTranslator.Options.ui.toolbarPosition

Type: String
Default: bottom

Where toolbar is positioned. Values can be:

- "top" – Show the translation toolbar at the top of the page
- "bottom" – Show the translation toolbar at the bottom of the page

WebsiteTranslator.Options.ui.mainContentElement

Type: HTMLElement
Default: null

Main container HTMLElement for the web page that contains all scrollable content.

WebsiteTranslator.Options.ui.showPopup

Type: Boolean
Default: true

Show or hide the translation popup Also when disabled, highlight for translated sentences won't be visible.

WebsiteTranslator.Options.ui.showLanguagesInNativeLanguage

Type: Boolean
Default: false

Show languages in native language in language selector.

WebsiteTranslator.Options.ui.translate

Type: String
Default: source

Translate UI, values can be:

- source – Show UI in source language
- target – Show UI in target language

WebsiteTranslator.Options.ui.showTranslationControls

Type: Boolean
Default: true

Show or hide "restore" and "cancel" translation buttons on UI.

WebsiteTranslator.Options.ui.layout

Type: String

Default: menu

Change display mode of the language selection in UI, values can be:

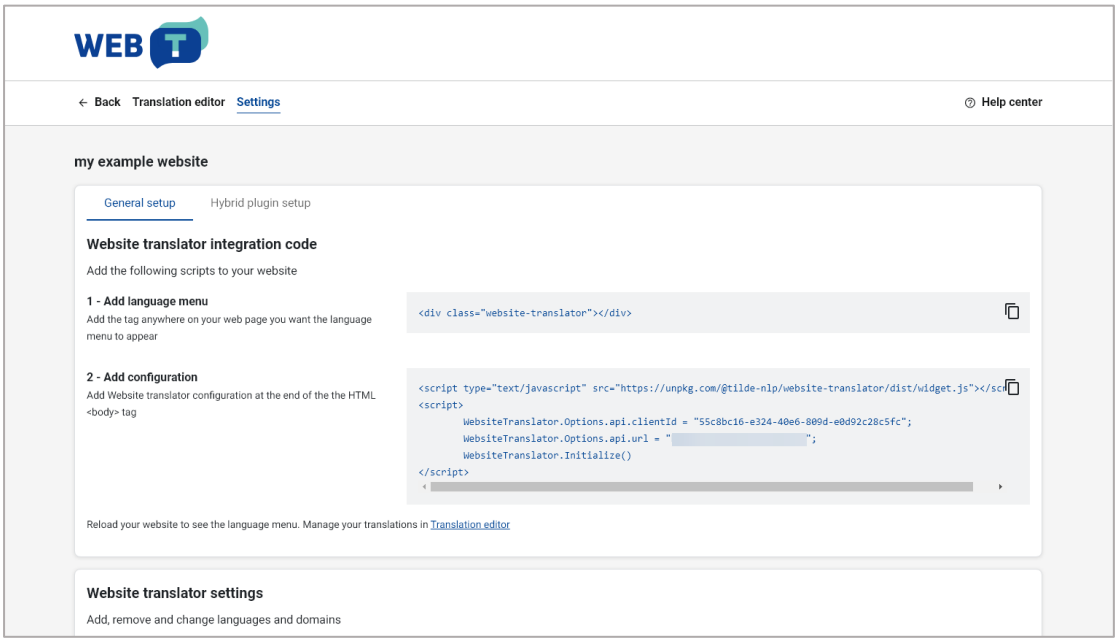
- "menu" – Show available languages in select.
- "list" – Show available languages as list items.
- null – Hide language selector.

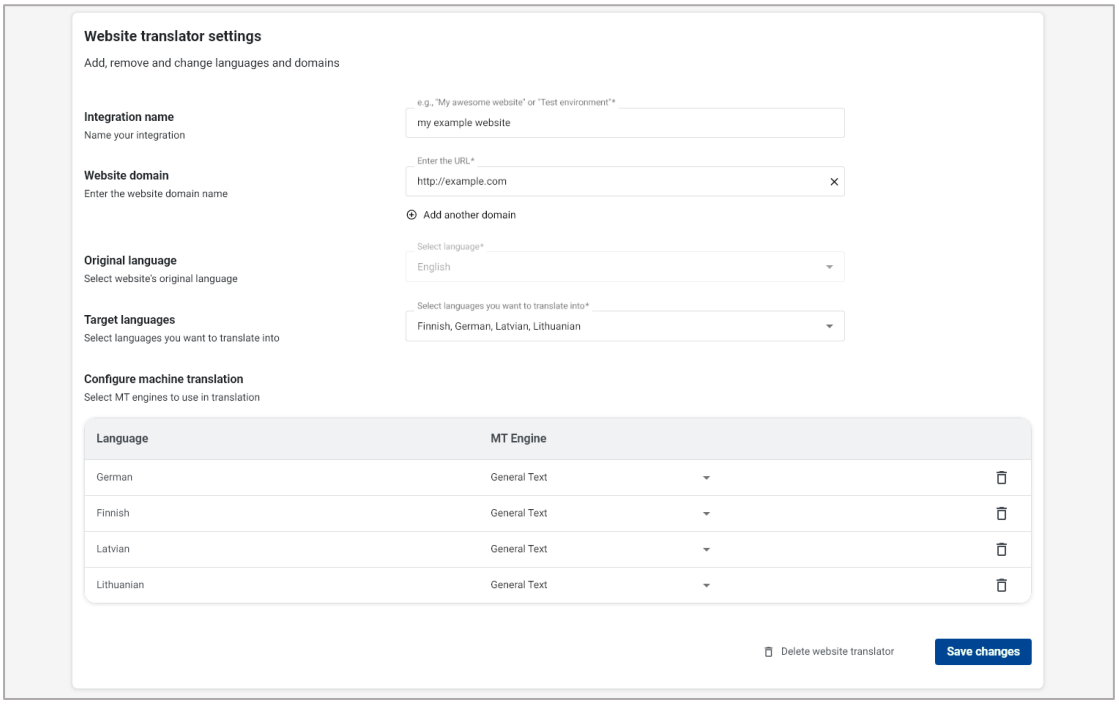
Edit existing integration

To change and add new languages or update the domains:

1. On the **Translation Hub**, open My integrations.
2. On the list of integrations, find the one you want to edit.
3. Click **Settings**.
4. On the **General setup** tab, scroll down to **Website translator settings**.
5. Make your changes and click **Save changes**.

If you have your website open, you will need to reload the page (Ctrl+F5) to see the changes.

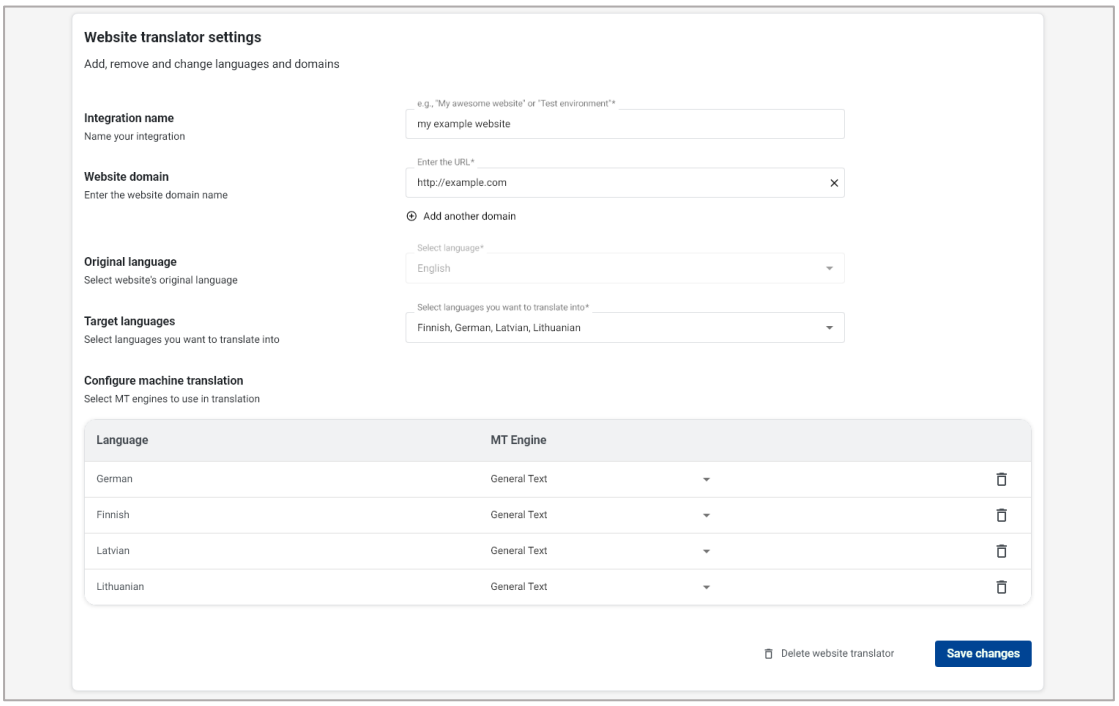


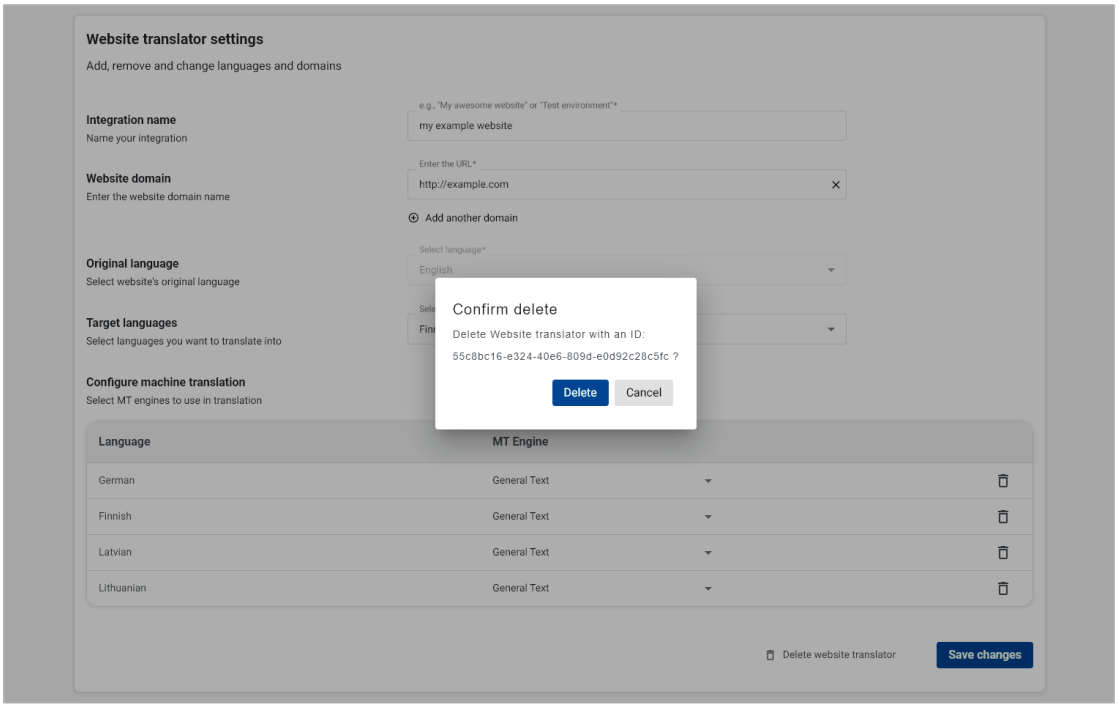


Delete integration

To delete an integration:

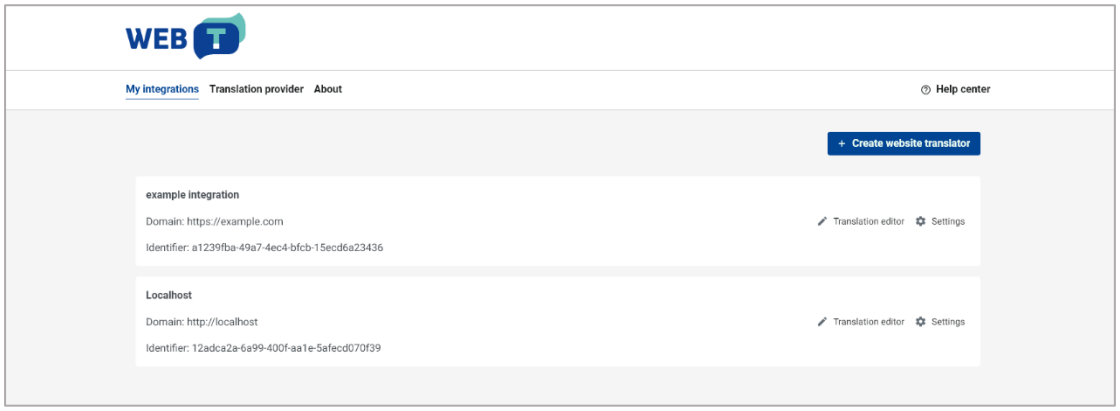
1. On the **Translation Hub**, open **My integrations**.
2. On the list of integrations, find the one you want to delete.
3. Click **Settings**.
4. On the bottom of the page, click **Delete website translator**.

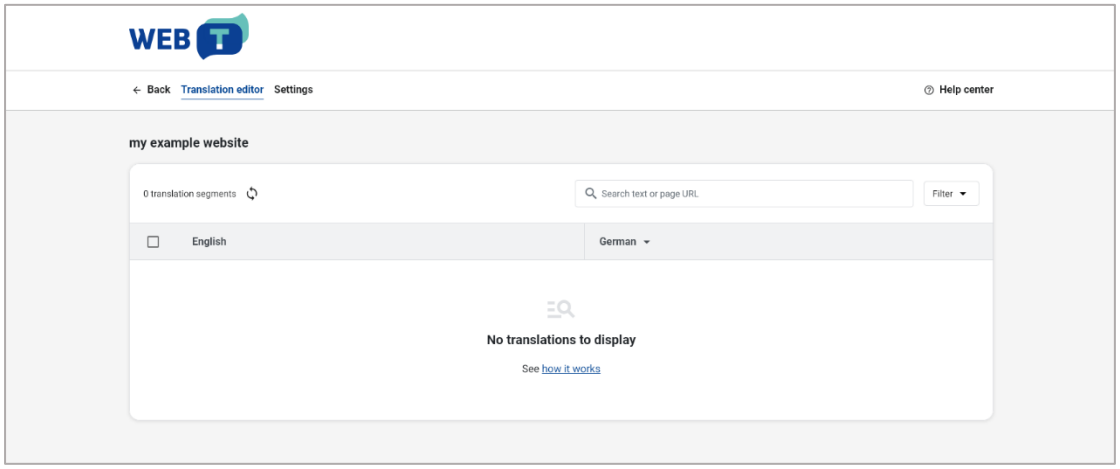




Manage translations

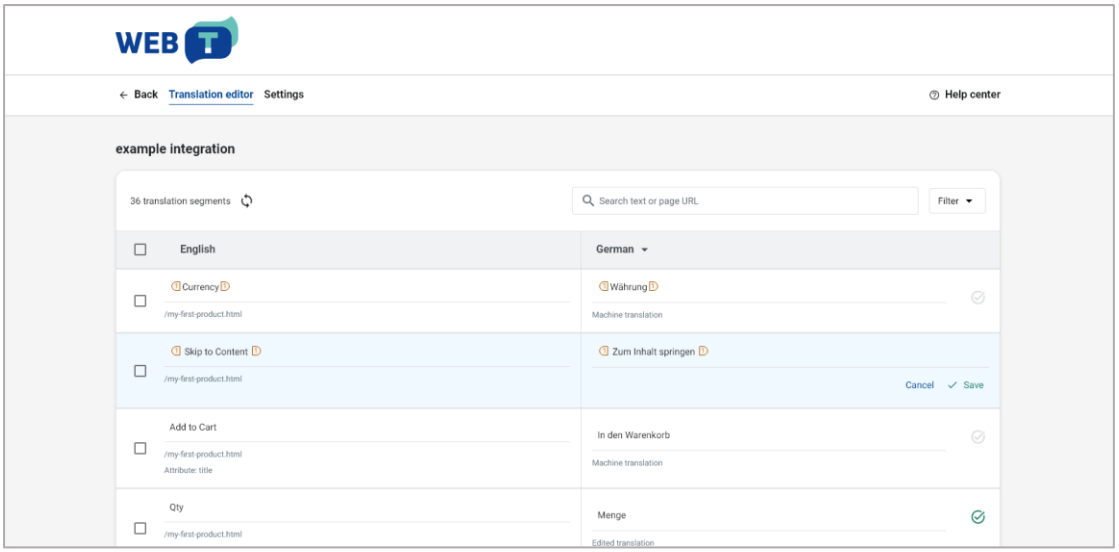
View, edit and delete translations on the Translation Hub. Translate your webpage with the plugin for the translations to appear, refresh the Translation editor table to see new translations.





Edit translation

Translations may need a bit of tweaking, or you might want to replace them with manual translations.



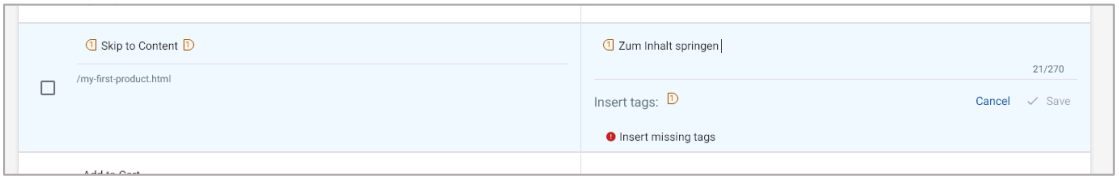
To edit the translation:

1. In the table of translations, find the translation.
2. Click on the row.
3. Edit the translation.
4. Click **Save**.
5. The translation will be updated and marked as “confirmed” in the database.

HTML tags in the source and translation are displayed as numbered blocks. The colouring and numbers let you follow the order and match the opening and closing tags. Translation must contain all the tags from the source.

Missing and mismatched tag errors will be displayed under the segment. All the tags from the original segment must be present in the translation.

Tags can be dragged to reorder or deleted and inserted by clicking on them under the segment.



Confirm translation

To confirm a translation:

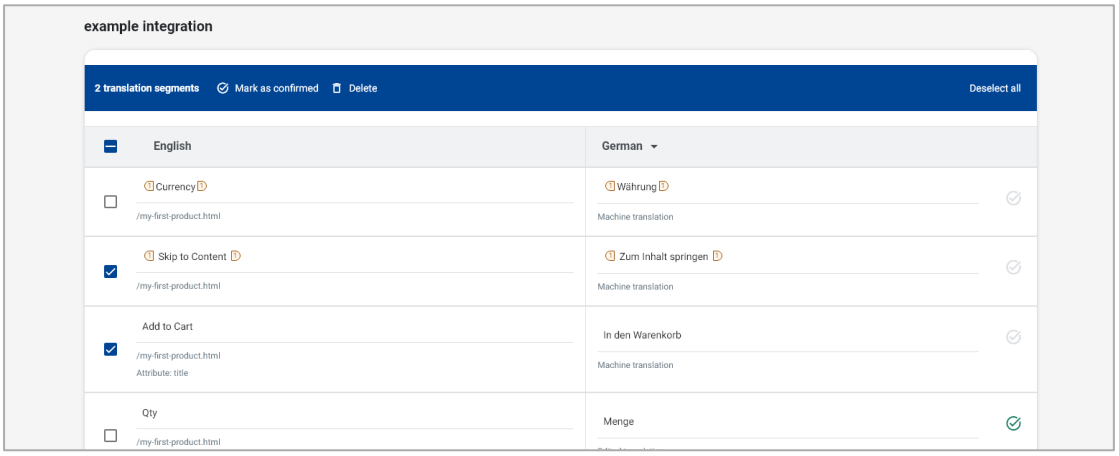
1. In the table of translations, find the translation.
2. Click the “✔” checkmark at the end of the row.

The translation in the database will be marked as “Confirmed”. When saving edits made to machine translation the status is automatically set to “Confirmed”. Machine translations not marked as “Confirmed” are still used in the translation. This feature is meant to help keep track of translations that should be reviewed.

Delete translations

To delete a translation:

1. In the table of translations, find the translation.
2. Click the checkbox at the start of the row.
3. Click **Delete**.



Delete translations having a certain target language

You have decided to remove a certain language from your website and want to delete the saved translations.

To delete all matching translations:

1. In the table of translations, select the target language from the drop-down menu.
2. At the start of the table header row, click the checkbox.
3. Click Delete.
4. Repeat until all translations are deleted.

Delete translations having a certain URL

You have decided to remove a certain page from your website and want to delete the saved translations.

To delete all matching translations:

1. Filter the translations by URL.
2. Repeat the steps from Delete translations having a certain target language for every language.

Delete all translations

Repeat the steps from **Delete translations having a certain target language** for every language. Or delete the whole Website translator integration.

Filter translations

Filter the translations table to:

- Find translations or source text with matching words or phrases.
- See only the translations you have not confirmed.
- Find translations belonging to a certain page.
- See only SEO translations.

Filter by status or translation type

To filter by translation type:

1. Above the translations table, open the filter drop-down menu.
2. Select **Machine translation** or **Manually edited**.

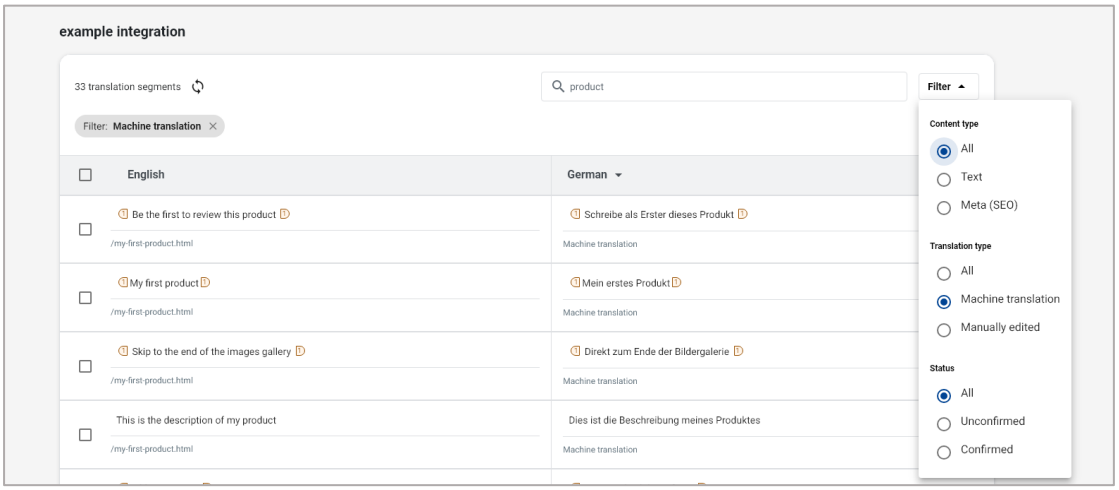
To filter by translation status:

1. Above the translations table, open the filter drop-down menu.
2. Select **Unconfirmed** or **Confirmed**.

Filter by content type

To see only the SEO meta tag translations:

1. Above the translations table, open the filter drop-down menu.
2. Select Text or Meta (SEO).



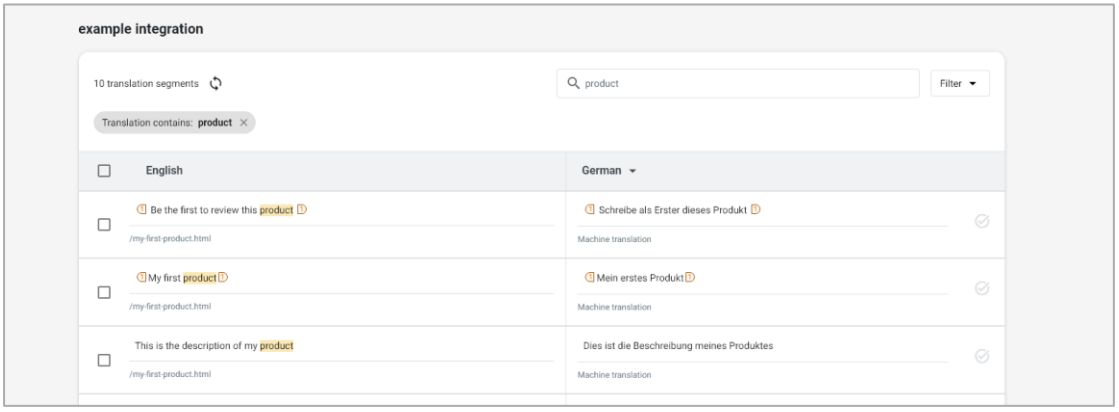
Filter by text

The text is searched in both the original language segments and translations.

To filter the table by source or target text:

1. In the search field, type the word or phrase.

2. Press Enter.



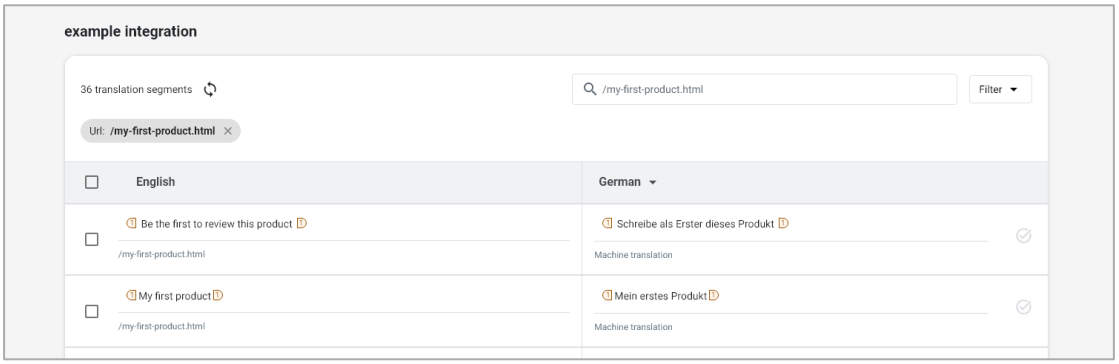
Filter by URL

The table can be filtered by:

1. full url of the page, for example: https://www.example.com/pages/my-page, www.example.com/pages/my-page
2. partial url, for example: /pages/my-page

To filter the table by source or target text:

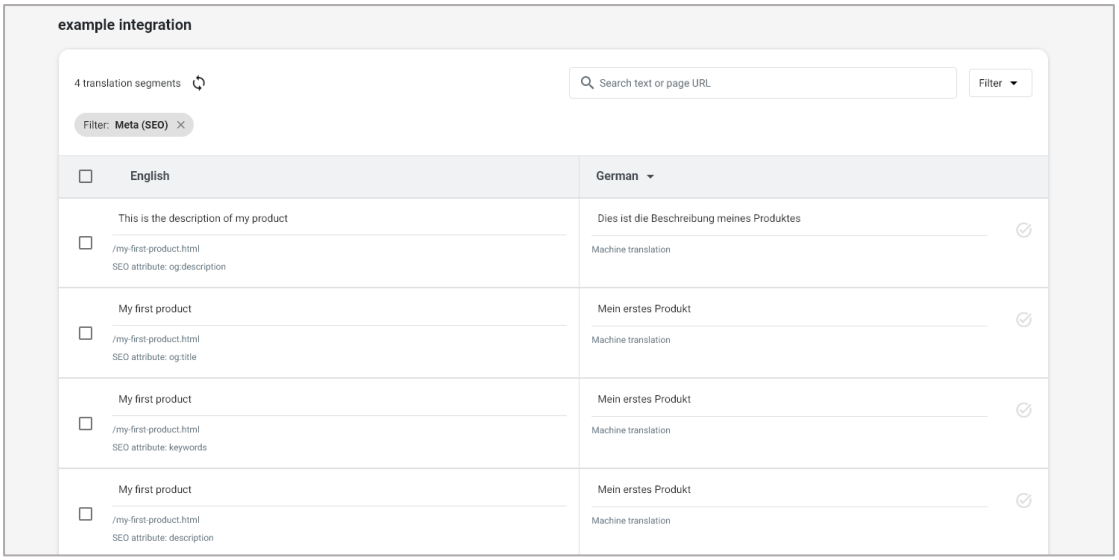
1. In the search field, type the url.
2. Press Enter.



SEO tags

SEO meta tags are translated by the plugin like any other text and appear in the **Translation editor** table. Translations are labelled with “SEO attribute:..”, found right under the page URL.

Filter the table by **Content type** to see only the SEO content.



FAQ

How do I integrate Website translator into my website?

See: Website translator integration

Can I customise the language selector look?

See: Set up language selector or Use a custom language

Do I have to use the default language selector?

The website translator can be integrated with a custom language selector that fits your website's design. Set the layout mode to
null **WebsiteTranslator.Options.ui.layout=null** and implement the selector yourself.

See: Use a custom language

Can I customise the translation notice?

See: Change display position and Hide the translation progress

Can I use machine translation alongside existing human translation?

If your website is already translated into several languages, you can define them as "third-party languages" in configuration. When selecting either of them the website translator will not try to machine translate the content.

See: Integrate into an already multilingual

I want to keep some parts untranslated / Some of my content is in another language.

Content that should not be translated, such as addresses, lists of company names etc. should be marked as untranslatable. Add **translate="no"** attribute to the elements.

Content that is in a different language should be marked with a language attribute. If you don't mark this content as not translatable or different language the Website Translator will attempt to translate it and corrupt the content. Translation from several source languages at once is not supported.

I want to explicitly mark the content that should be translated.

Mark the elements that should be translated with **translate="yes"** and add the **WebsiteTranslator.Options.translation.translateOnlyAllowedTags=true** option to the configuration. No other content will be translated. If some child elements should not be translated, mark them with **translate="no"**.

```
<html lang="en">
    <head>
   <script      type="text/javascript"      src="https://unpkg.com/@tilde-
nlp/website-translator/dist/widget.js"></script>
    </head>
    <body>
        <div class="website-translator"></div>

        <!-- This will be translated because it inherits the document
language "en" -->
        <p>This will be translated</p>

        <!-- This will not be translated because it has the
translate="no" attribute -->
        <p translate="no">This will not be translated</p>

        <!-- This will not be translated because it has a different
language from the document source language -->
        <p lang='ja'>これを訳して</p>

        <ul translate="no">
            <!-- This will not be translated because it inherits the
translate="no" attribute -->
            <li>Coffee</li>

            <!-- This will be translated because it overrides the
inherited translate="no" with translate="yes" -->
            <li translate="yes">Tea</li>
```

```
            <!-- This will not be translated because it inherits the
translate="no" attribute -->
            <li>Milk</li>
        </ul>
    </body>
    <footer>
        <script>
            // Configure plugin
            // 👇 Change XXXXXXXXXX to your Client-ID
            WebsiteTranslator.Options.api.clientId = "XXXXXXXXXX";
           // 👇 Change XXXXXXXXXX to your Translation Hub URL
           WebsiteTranslator.Options.api.url = "XXXXXXXXXX";

            WebsiteTranslator.Options.ui.toolbarPosition = "top"
            WebsiteTranslator.Initialize()
        </script>
    </footer>
</html>
```

I want the translation progress bar to not be visible to website visitors.

Add **WebsiteTranslator.Options.ui.headless=true** option to the configuration. Website visitors won't see the translation progress bar and be able to revert the page to the original language by clicking the restore link.

How can I edit the translations?

Translations can be edited on the Translation Hub. See: Manage translations

Does the plugin translate images and media content?

The widget does not translate images or media content.

Will the edit I made in one page affect the translations of the whole website?

Edits are made on a segment basis. You will be able to change the translation of a whole sentence or paragraph, not a single word in a sentence. If the same segment appears elsewhere on your website, it will use the edited translation. If there are any changes in the segment it will have to be edited and approved separately. For example, editing a translation of an element in a menu or footer will affect all the pages it appears in, but changing the translation of a street name that appears in different sentences will only affect the identical sentences.

Will the translation edits I made in my test environment be used in production?

Yes, if you have used the same Website translator integration. If you want to keep the translations separate, generate Website translator for each of the environments.

I want to add or change languages, or MT engines.

You can change or add new languages and domains in the **Settings** of each **integration**. On the **Translation Hub** open **My integrations** and find the correct integration, then click **Settings**.

How to update translations after changing translation engine?

To update the existing translations after changing the machine translation engine or translation provider:

1. Delete the old translations for the specific language.
2. Translate all content again.

Troubleshooting

I can’t find some of my page content in the Translation editor

Make sure you open/view all the menus, forms and other dynamic content when translating your web page, otherwise, they won’t be translated and saved to the database for editing. Wait for the translation to finish before closing the page.

How to enable debugging for the plugin?

Set **WebsiteTranslator.Options.debug=true** in the JavaScript integration code.

The language selector is not displayed

Make sure you have added an element with **class="website-translator"** to your page.

My website is not being translated

1. The language is not set

Check if the language is declared correctly in HTML. Your HTML should have lang attribute defined. Website translator supports ISO 639-1 language codes with optional ISO 3166-1 alpha-2 country codes.

2. The contents have not been loaded

Make sure the contents of the page have been loaded before initialising the website translator. Different methods can be used to determine if the page has been loaded.

```
$(document).ready(function() {
    ...
    WebsiteTranslator.Initialize()
})
```

For static websites it is recommended to place the Website translator configuration script at the end of the body tag.

Translation doesn’t work with eTranslation provider

Websites using WEB-T with eTranslation integration enabled must be published and accessible online from eTranslation service. Translation with eTranslation will not work otherwise. To fix this, publish these websites online.

Language of the Website translator’s UI keeps changing

The language of the translation bar and popups is set to the language you are translating into.

If you want them to be in the website’s original language, add the following line to the JavaScript integration code:

```
WebsiteTranslator.Options.ui.translate="source"
```

Website translation is slow / Translation makes the page load slower

The first time the content of any page is loaded it will take some time to translate it. The translation will be saved in a translation database and will load faster the next time.

To make it load faster for visitors it is recommended to “pre-translate” all the pages of your website. Go through the pages, selecting each language from the Website translator’s language selector to translate them.

Translation timed out

Waiting for the response from the translation engine took too much time. You can retry translating later.

The Website translator doesn’t look right

Globally set CSS styles can affect the appearance of the website translator. You can restyle it with CSS and fix the issues.