
Translation Proxy Online Documentation

Release latest

Aug 16, 2023

Contents

1	Introduction	1
1.1	Features	2
1.2	White-label	3
1.3	Pricing - plans	3
1.4	Pricing - grandfathered pay-as-you-go	3
2	Getting started	5
2.1	Registration & Login	5
2.2	Setting up Your First Project	5
2.3	Running a Discovery	6
2.4	Giving a Quote	9
2.5	The Workbench	11
2.6	The 3-Phase Workflow	11
3	The Dashboard 2.0	13
3.1	Introduction - Project overview	13
3.2	Pages list	24
3.3	Crawl	31
3.4	Content	47
3.5	Snapshot	55
3.6	Settings	62
3.7	Page modifiers	90
3.8	Publish	93
3.9	Third party integrations	122
3.10	Workbench	123
4	Account settings	125
4.1	Account settings	125
4.2	Third party integrations	127
4.3	Translation memory	128
5	The Workbench	129
5.1	Introduction	130
5.2	Moving Around	131
5.3	Views	138
5.4	Translation of Segments	142
5.5	Collaboration	148

5.6	Cookbook	151
6	Maintenance	153
6.1	Project maintenance	153
6.2	Automation possibilities	153
6.3	Automatic pre-translation	154
6.4	Caching And The Proxy	155
7	Cookbook	157
7.1	Localising Wistia video embeds	157
7.2	Whitelabel Setup	158
7.3	JS/JSON/XML Translation	160
7.4	Resource Translation	164
7.5	HubSpot Forms	165
7.6	Page requests, CDNs and Caching	170
7.7	Page Request Evaluation	173
7.8	Staging Domains	177
7.9	Left-Right conversion	179
7.10	SSL Certificates	180
7.11	Proxy modes - X, P, live	182
7.12	HTTP Headers	183
7.13	Secure Login - Passing Cookies to Scans	184
7.14	Site Search	187
8	Technical Reference	195
8.1	Architecture	195
8.2	Request Handling	196
8.3	Classification of Content	197
8.4	Translation Memories	198
8.5	Using TMs	198
8.6	Page modifiers	199
8.7	Client-Side Translator	200
9	Troubleshooting & Support	203
9.1	Contacting Support	203
9.2	Issues	204
10	FAQ	211
10.1	General	211
10.2	Caches	212
11	Glossary	215
11.1	Page List Tags	215

CHAPTER 1

Introduction

Translation Proxy is a cloud-based translation proxy solution designed to make websites available in several languages. A translation proxy is a layer placed between the browser and the original website, through which the visitor sees the original website in a different language.

There are several proxy solutions available on the market, yet the translation proxy is unique in that:

- it is a solution targeted primarily at LSPs - which also means that this is the **only** LSP-independent solution on the market
- also available as a white-label solution
- standard-compliant XLIFF export option ensures CAT-independence
- an automated XLIFF export/import option is available between CAT / Translation management systems and the proxy
- in-context live preview available in certain CAT-tools and the online Workbench at translation-time

What does that mean?

If you are a **business owner**, it can help you reach a wider customer base by providing information to your potential customers in their native language. What's more, you can do more than just translating the text on your website, you can **localize** it: you can also adapt your message, the images displayed, or even your product range offered to the targeted culture. All of this is possible without the need for heavy upfront investment in IT infrastructure and personnel or the hassle with regular maintenance and upgrade. The translation proxy takes care of the IT part so that you could concentrate on the content - and growing your business.

If you are a **language service provider (LSP)**, you can offer cutting-edge website localization services to your customers - even under your own brand name! The translation proxy provides the technology, takes care of the IT infrastructure, leaving you to concentrate on your core business: cross-cultural communication. What's more, your translators don't need to learn to use just another tool, they can keep using their preferred CAT-tools.

Sounds good?

There are several challenges both business owners and language service providers face during website translation. The "ideal" workflow would be to create the content in the original language, get it translated into the desired languages, and then publish all language variants at the same time, from the website owner's own content management system

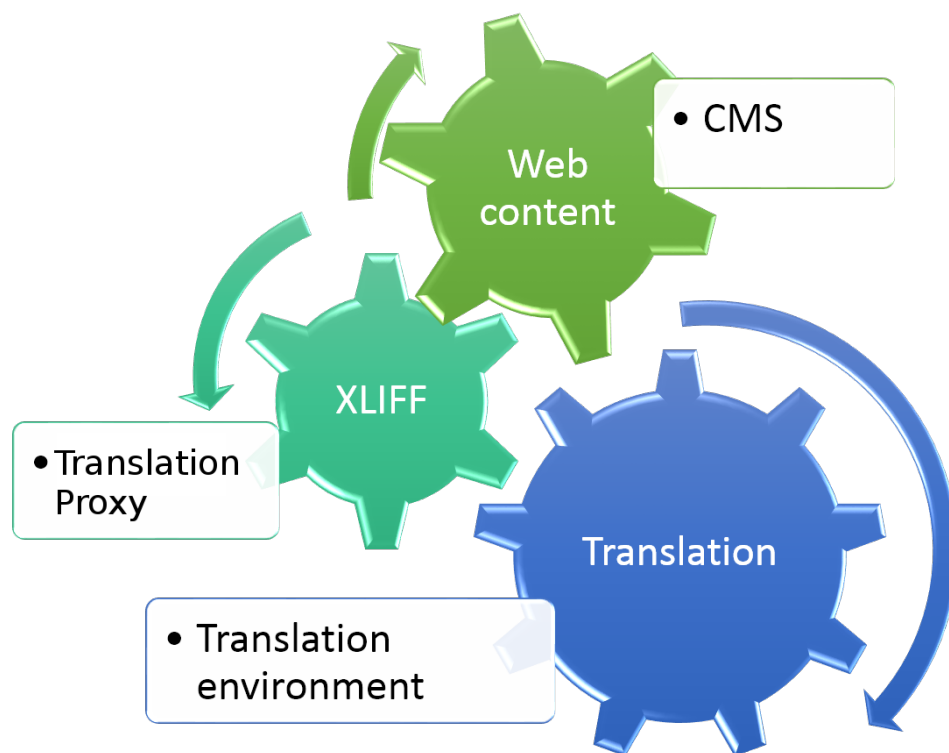
(CMS) - right from the very first page on the website. However, the reality is often different. Apart from the fact that not many CMSs are capable of handling several languages, usually, website localization comes into the picture at a later stage, when there is already a huge amount of data published on the website. In most cases, the website owner can't extract the content for translation. If they can't extract the original, there's no easy way to load the translated content back either. Furthermore, if the website owner can't extract the content into a translatable format, it is impossible to get a proper estimate for the translation costs in time and money...

The proxy can, however, discover the website by following links and grabbing translatable and localizable content - and convert it into a translatable format. This gives a realistic view of the magnitude of the translation task, and, thanks to the translation proxy, even a partially translated site can give full user experience on the website visitor's side.

Data can be extracted with a couple of clicks - and the publication of the translated site is similarly easy.

Where do we fit in the localization workflow

The proxy bridges the gap between the CMS and the translation workflow; it enables you to extract content from the website in a translation-ready format that can be used in *any* translation environment.



Workflow

- If you are a content owner, it means that you have a technology solution that enables you to choose any LSP that suits your language requirements the best.
- If you are an LSP, it means that you can take care of the website translation requirements of your clients.

In either case, technology will no longer be the bottleneck.

1.1 Features

- process HTML, JavaScript/AJAX/JSON, XML (note: translation of Flash is not supported)
- Use the Preview mode to see everything in context.

- Use Advanced Settings to translate text coming from JSON/XML sources.
- Automatically crawl static pages / HTML content only. Add extra AJAX URLs with the proper parameters.
- Fine-tune your settings to help the crawler decide what URLs to handle as same, and what to visit looking for new content.
- Translate forms, messages and dynamic content
- Translate images: replace them with their localized counterparts on a target-language basis.
- Link any external domain via multiple projects.
- Modify page behaviour using customized CSS and JavaScript injection.
- Use regular expressions to filter for content.

and many more!

1.2 White-label

The proxy offers a white label version that can be customized with your corporate logos and domains to create a branded version, allowing you to use & sell the translation proxy as your own product. To create the branded version, seven criteria must be met. See the “White-label setup” section for details.

1.3 Pricing - plans

Based on the feedbacks from our client, we’ve built an all-inclusive pricing model, with a predictable, flat-rate monthly fee and a few add-ons. With the only variable being the number of translatable source words, it’s easy to multiply by the number of target languages to get the final TCO.

1.3.1 Add-ons

- **Translation proxy:** with the new pricing scheme, JavaScript publishing is the default, while the proxy is available as an optional add-on
- **Site search:** recommended if native site search is required
- **SEO setup:** CDN / reverse proxy integration is required for subdirectory publishing, highly recommended to enhance SEO effectiveness
- **Layout adjustment:** depending on the structure of the original site and the language pair, the translation might yield to text expansion, breaking the look and feel of the site. Fixing issues could involve a need for professional attention, changing the layout of the translated pages.

1.4 Pricing - grandfathered pay-as-you-go

Our old pricing follows the ‘pay-as-you-go’ model, so you only get charged for what you use. The total cost is made up of 2 types of fees: one-time fees and a monthly charge. It’s still available for the time being upon demand.

1.4.1 One-Time Fees

- **Discovery:** 1 EUR or 1.2 USD / 1000 pages for every Discovery
- **Scan:** 2 EUR or 2.4 USD / 1000 source words for each **new** word during a Scan (if no new words are found, a Scan counts as a Discovery)
- **Translation Memory:** storing your (human or machine) translation in the database costs 10 EUR or 12 USD / 1000 source words / target language

Content extraction and translation memory fees apply only the first time around, which means 102% repetitions are not counted. Once a segment is stored, subsequent scans will treat it as repetition and **no additional charges** will apply.

1.4.2 Monthly Fees

A 1 EUR or 1.2 USD / 1000 page requests monthly fee is applied for serving your translated content over the proxy.

In exchange, you get a guaranteed 99.99% website availability, and a capacity to handle practically unlimited traffic. You also have the option to serve the translated site from your server (in which case no proxy fee will apply), but in this case, availability and traffic handling depend on your infrastructure.

CHAPTER 2

Getting started

Let us give you a quick overview of how the proxy is used. In this section, we give you a quick introduction to the Dashboard 2.0 and we'll also set up a simple project.

2.1 Registration & Login

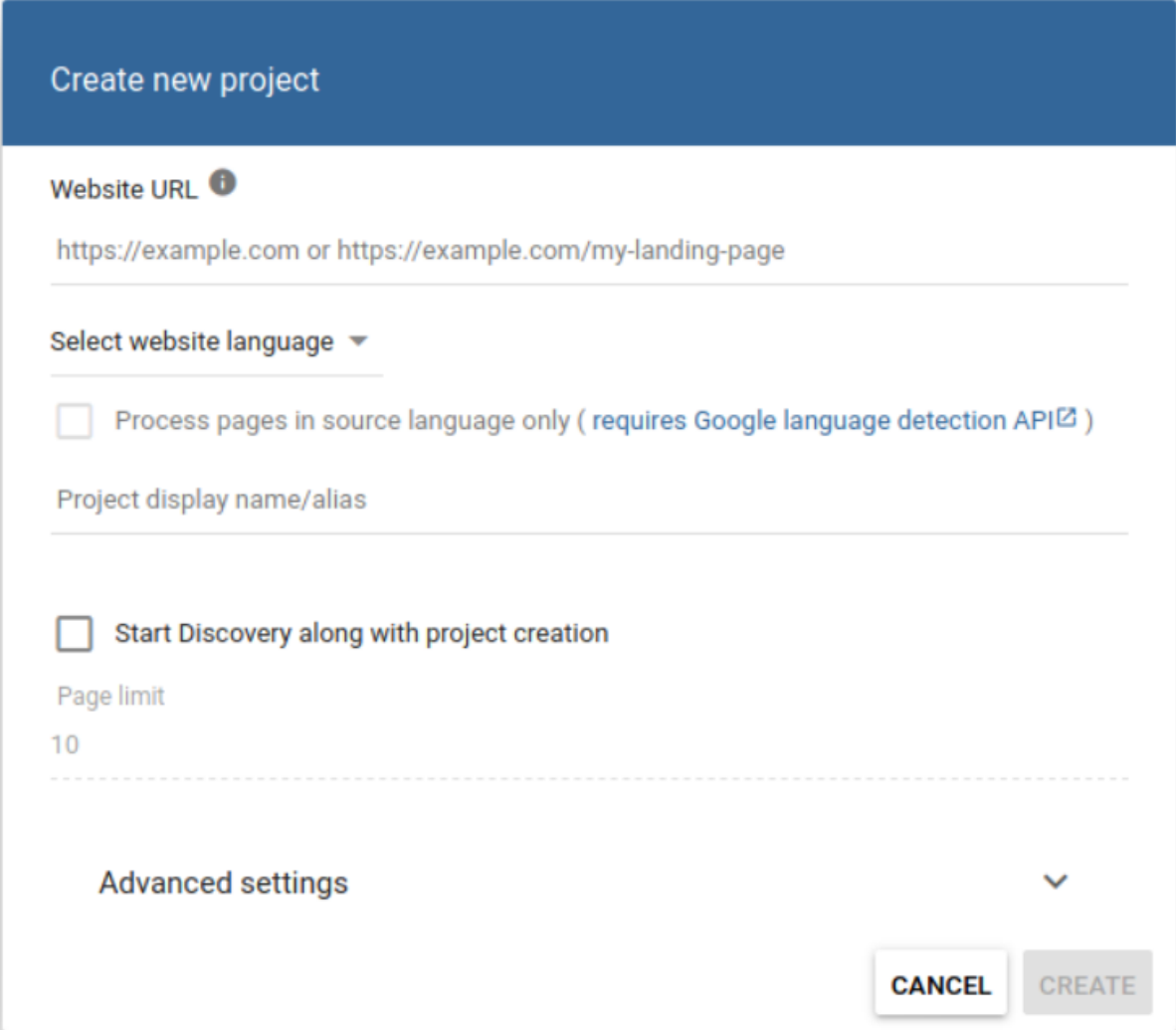
To use the proxy, you need to register and set up an account for the service. You can start using the service right away after registration.

Logging in, you will be taken to the Dashboard 2.0, which is the new and improved project management center, every detail of which we'll get to in later sections of this manual. There will be no existing projects at the outset, so let's try setting one up.

2.2 Setting up Your First Project

To do this, click on **Create new project** dropdown at the top and choose **Add project**.

This opens the **Create new project** dialog box, where you can enter the URL of the website you would like to translate and select the website language. You can also add an alias to your project. This is an easy to remember name that you can recognise your project from. You can also decide to start a Discovery immediately after creating your project to skip the next section. Click on Advanced Settings to access extended functionality, such as custom SRX files.



The screenshot shows a 'Create new project' form with a blue header. The form contains the following fields and options:

- Website URL** (with an information icon): `https://example.com` or `https://example.com/my-landing-page`
- Select website language** (dropdown menu)
- ☐ **Process pages in source language only** (requires [Google language detection API](#))
- Project display name/alias**
- ☐ **Start Discovery along with project creation**
- Page limit**: 10
- Advanced settings** (with a downward arrow icon)
- CANCEL** and **CREATE** buttons at the bottom right.

Project Add

2.3 Running a Discovery

The next step is to figure out what (and how much of it) to translate. You can do so using a Discovery. This effectively means running a crawler on the site and allowing the proxy to ‘explore’ it in its entirety. It can then provide statistics for you such as a list of pages visited and the word count.

✓ Type
✓ Scope
✓ Pages
✓ Fine-tune
5 Summary

Crawl Settings Summary

Type: Wordcount (Discovery)
Store content: No
Scope: Re-visit current pagelist & Find new pages

Add a memo to the Crawl
Supports markdown syntax, maximum 1000 characters. More information about markdown syntax: <http://commonmark.org/help/>

WRITE
PREVIEW

Page Limit

100

- URLs in the page list are visited first

Cost Projection*

Cost name		€	Cost
Word count for	100 pages	*0.001	€0.10
Content extraction for	N/A unique words	*0.002	N/A
Translation for	N/A target languages	*0.01*0	N/A
			€0.10

*This estimation is based on an empty database.

BACK
START CRAWL

Crawl

wizard

If you decided not to immediately run a Discovery on project creation, once you click ‘Create’, you’ll be directed to the Crawl wizard. This 5 step wizard is designed to guide you through setting up crawls. When running your first crawl, it’s recommended to select Discovery on step 1 and ‘Re-visit current pagelist & Find new pages’ on step 2. On step 3, you can specify a page limit. This is the maximum number of pages that the crawler is allowed to visit. This feature lets you to start a crawl without worrying about costs getting out of control. Make sure that this field has a reasonable number and that it’s never left empty. Step 4 has some advanced settings. These can allow you to fine-tune the crawl’s settings. The defaults are safe. So click next for now. Finally, in step 5, you can review your settings and add a memo. Then you can click Start crawl.

Depending on the size and speed of the site, a Discovery can take quite some time to finish. A spinner on the Crawl list will indicate that the system is currently working.

After the process is over, you’ll receive an e-mail about the results. You can also view them in the Crawl list.

Learn more about [Crawl statistics](#).

Wordcount (Discovery)

Done

example@example.com

Started July 01, 2019. 11:36:33

Finished July 01, 2019. 12:01:15

Took 0:24:45

Visited 489 pages

No more pages left.

No memo was provided.

Statistics of pages processed this crawl

Match

Segments

Source words

Latin characters

Latin characters with spaces

Asian glyphs

102%

38,174

121,476

603,137

688,825

0

101%

92

1,634

8,242

9,787

0

100%

626

4,524

23,426

27,330

0

99%

1,864

12,579

66,148

78,173

0

98%

53

678

3,465

4,113

0

Unique

4,974

70,894

366,283

433,106

2

Selected total:

7,609

90,309

467,564

552,509

2

Crawl Settings Summary

Page limit

Unlimited

CrawlJob log links

https://app.easyling.com/ui/recrawl/log/6i7qzgxal6yyi5/6i7qzgxal6yyi5_15

61973790177_95562-00000.log

Collect selected resource types

☐ new HTML pages

☐ JS, CSS resources

☐ images, binary resources

Skip content type check

☒

Cost Projection*

Cost name

€

Cost

Word count for

489

pages

*0.001

€0.49

Content extraction for

90309

unique words

*0.002

€180.62

Translation for

2

target languages

*0.01*90309

€1,806.18

€1,987.29

*This estimation is based on an empty database.

Previous crawls

Wordcount (Discovery)

Done

Started July 01, 2019. 11:36:33

Finished July 01, 2019. 12:01:15

Took 0:24:45

Visited 489 pages

No more pages left.

Wordcount (Discovery)

Done

Started June 25, 2019. 14:40:21

Finished June 25, 2019. 14:43:27

Took 0:03:09

Visited 2 pages

No more pages left.

Wordcount (Discovery)

Cancelled

Started June 25, 2019. 14:38:25

Visited 0 pages

Cancelled - displayed statistics are partial results of the crawl.

Content extraction (Scan)

Done

Started June 13, 2019. 16:45:25

Finished June 13, 2019. 16:45:33

Took 0:00:11

Visited 0 pages

No more pages left.

LOAD MORE

Crawl

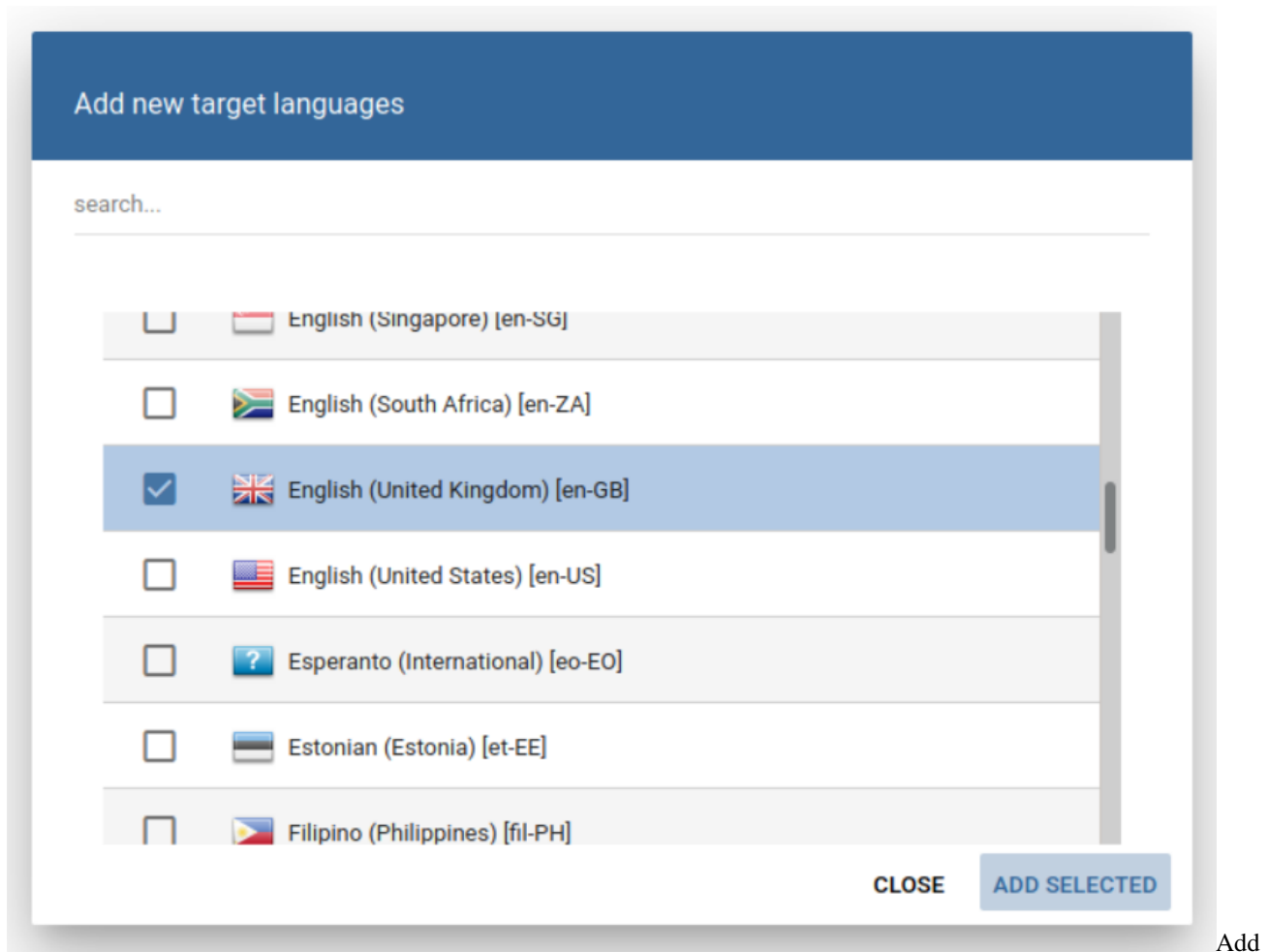
list

2.3.1 Add a Target Language

You will also need to add your target language(s), so use the option on the Target languages card in the Project overview to add then to the project. It's not just that there is not much to do in terms of translation without a target: many crucial features (including the Preview proxy and the entire Workbench) are entirely unavailable as long as no target languages are set.

8

Chapter 2. Getting started



target language

You can add any number of target languages. You can use the search function to lookup languages based on locale code or country name.

2.4 Giving a Quote

You can use the results of the Discovery to give a quote (based on **unique** source words) to your clients about the estimated work-hours and expenses you forecast for a given project.

The results are an accurate indication of the translation costs associated with the text. However, with websites, it is prudent to **consider other (technical) details** before taking the word count results of the Discovery process at face value.

Investigate the source site and consider the following:

- Is there a great deal of **dynamic content** generated by JavaScript?
- Any **Site Search** functionality? A **webshop**? A **forum**?
- Any other **web apps** that would have to be localized?
- Do the **average word lengths** of the source and target languages differ significantly?
- Is the **direction of the target language** different than that of the source language?

- Which pages are targeted for translation? Which pages need to be excluded? Ask the client if they have a specific page list.
- Does the site have mixed-language content? If yes, ask the client to specify the source language(s) they need translated.
- Is there an extant Translation Memory that could be used?
- Is there any region-specific content? Does the site use geolocation?
- Is there any content behind a secure login?
- Are there any subdomains? Example.com and blog.example.com require two separate projects that need to be linked.
- Should bots be treated differently to reduce costs?
- How should new content be handled?
- Are there any other special requirements?

If you answered yes to any of those questions, that will require some deliberation, often beyond the primary focus of translators: UI fixes and a measure of fiddling with what's under the hood - take those expenses into account when you make your quote.

If you are unsure as to how to go about translating a part of a website, feel free to contact our **Support Centre** and we'll help you get an accurate picture of the required effort and costs.

It is also advised to negotiate the expected workflow with the client at the quoting phase. The translation of a website is, in most cases, a never-ending process, as new content is added to the original site at certain intervals.

The question is, how content added *after* the initial quote should be treated - both from a technological and fiscal viewpoint. It is a good idea to ask the client about their intentions for update cycles and new content.

Do they wish to publish at the same time in all languages? Or publish on the original site without delay, and then publish the translations later, as they become available? The different options will require different approaches when you get to the maintenance phase of the project.

As a translation proxy is practically a translation layer on top of the original site, serving translations from the datastore by replacing original content on-the-fly, new content will not be replaced, as translation is not stored for that. In practice it means that newly added content will appear in the original language on the translated site. This is called *bleedthrough*. There are 2 approaches to this phenomenon: let bleedthrough happen, to make new content available right away, even if it is in a different language, or block new content from appearing until translation is done. Both have their clear advantages and drawbacks, so you have to discuss with your client which option is more acceptable for them - and set up your project accordingly.

2.4.1 Sales tool for mass production

We also offer a Sales Tool to help LSPs and freelancers in growing their business.

If you have a well-defined group of potential customers you'd like to offer your translation services to, like hotels or restaurants with only monolingual websites in your area, the translation proxy makes it easy for you to impress the business owners. Just collect the URL addresses, add them to the Sales Tool, and the translation proxy will automatically create a project for all webpages according to the settings you specify. Once the translation and post-editing of the translated main pages are ready, you can send a link to the business owners. If your potential clients are impressed with the translated page and the fact that no IT involvement is required on their end, you have a better chance to win the deal.

2.5 The Workbench

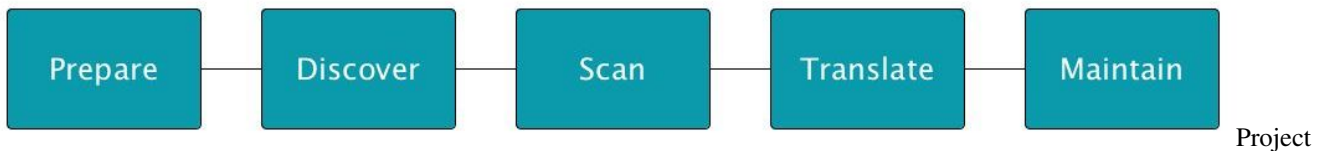
You can export all source segments, translate them in your CAT tool of choice and then reimport your results. But going through that cycle for every small change would get rapidly tedious - wouldn't it be great if you could edit & control your translations in the cloud, where it would all update in real time? You're looking for the Workbench.

In Pages list, you can right click any page and choose 'Translation workbench (list view)' and you'll be taken to the Workbench in a different tab.

If the Dashboard 2.0 is the Project Management Center of the proxy, then the Workbench is the cloud CAT tool, where translation itself takes place. There are many features you can use in the Workbench to make working with websites easier - see the 'Workbench' section of this manual for the details.

2.6 The 3-Phase Workflow

Barring some detail (withheld for the sake of a convenient introduction) the above process is all that you need to get a website translation project going.



Workflow Overview

Our idea of a project's lifetime can be summarized in the 3-phase Workflow.

2.6.1 1. Discover & Quote

Set up a project and Discover it. Have a Unique Word Count total and a general idea of any technical issues involved. Give your quote to the client (perhaps demo/impress them via the Live Preview). Win the bid.

2.6.2 2. Ingest & Translate

After you are entrusted with the project, collect all text content into a database (overcoming any technical issues that may arise in the process). When you have your data, export it to your CAT tool of choice or translate in the Workbench to a selection of target languages. Reimport and edit. Use our Proofreading and Workflow features to ensure quality.

2.6.3 3. Publish & Maintain

After the translation is greenlit by the proofreaders, you can verify the serving domain and publish the translated website. Add a language selector to the source site. Generally, it is with publishing a website that a deadline is met.

But don't forget that a website is a living thing, with new content arriving every day - the final stage of website localization is always maintenance - making sure that new content gets translated according to schedule, all the while ensuring that visitors to the site will not be inconvenienced by bleedthrough of untranslated content.

Maintenance is the "long tail" of website translation - there are a variety of features in the proxy that make it a lot easier than it would be otherwise.

In the following pages, You will find everything there is to know about using the proxy. Keep reading!

The Dashboard 2.0

3.1 Introduction - Project overview

The Dashboard 2.0 is your new and improved command center. It contains a variety of features you can use to manage your projects. In this manual, we'll take these options in the order that they appear in the menu on the left side of the screen.


When you open the Project overview, the screen will display a few general settings described below. Depending on your rights on the project, some of these may be hidden from you.

3.1.1 Project info

i Project info

Domain:

Project code:

Source language:  Hungarian (Hungary) [hu-HU]

Project created: August 25, 2020. 10:39:42

Project alias: for docs writing

Alternative domains: ?

DELETE PROJECT

info

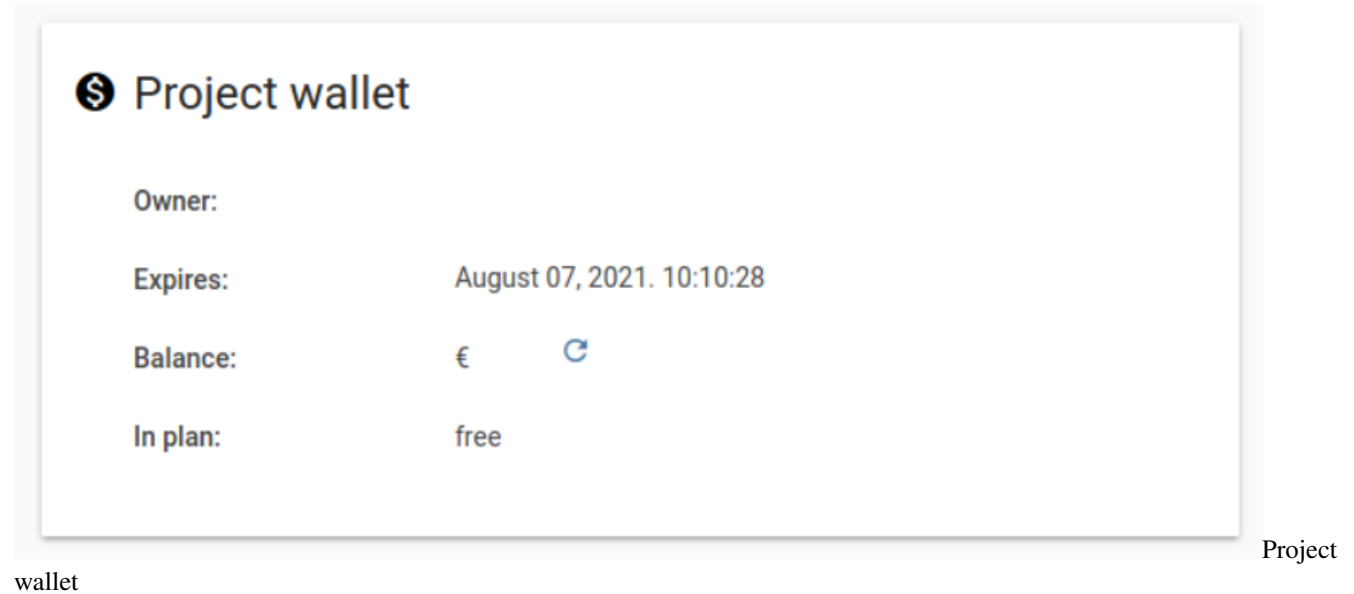
Project

This card displays the key properties of your project. They are the following:

- **Domain:** Exactly what it says on the tin, the website address is a property of your project that cannot be altered once declared during project creation.
- **Project code:** An 8 character long identifier that's unique to your project. It can be used to track projects and is necessary when you ask help at our support email address.
- **Source language:** The language you translate from. It can't be changed here but if you need to, you can email support and we'll be happy to change it for you after the necessary precautions.
- **Project created:** The creation date of your project.
- **Project alias:** This alternative name will be displayed in the Project dropdown below the URL, for easy identification of your projects. Project aliases are project-internal, they will not be displayed anywhere on the translated site.
- **Alternative domains:** Sometimes a website serves content both on the `www` subdomain and the naked domain, such as `example.com`. In these cases, it is useful to set things up over the proxy so that the different URLs are handled similarly. After creating a project, this field is automatically filled with the complement of the Domain. Add any further subdomains that contain identical content to this list. Separate them with commas.

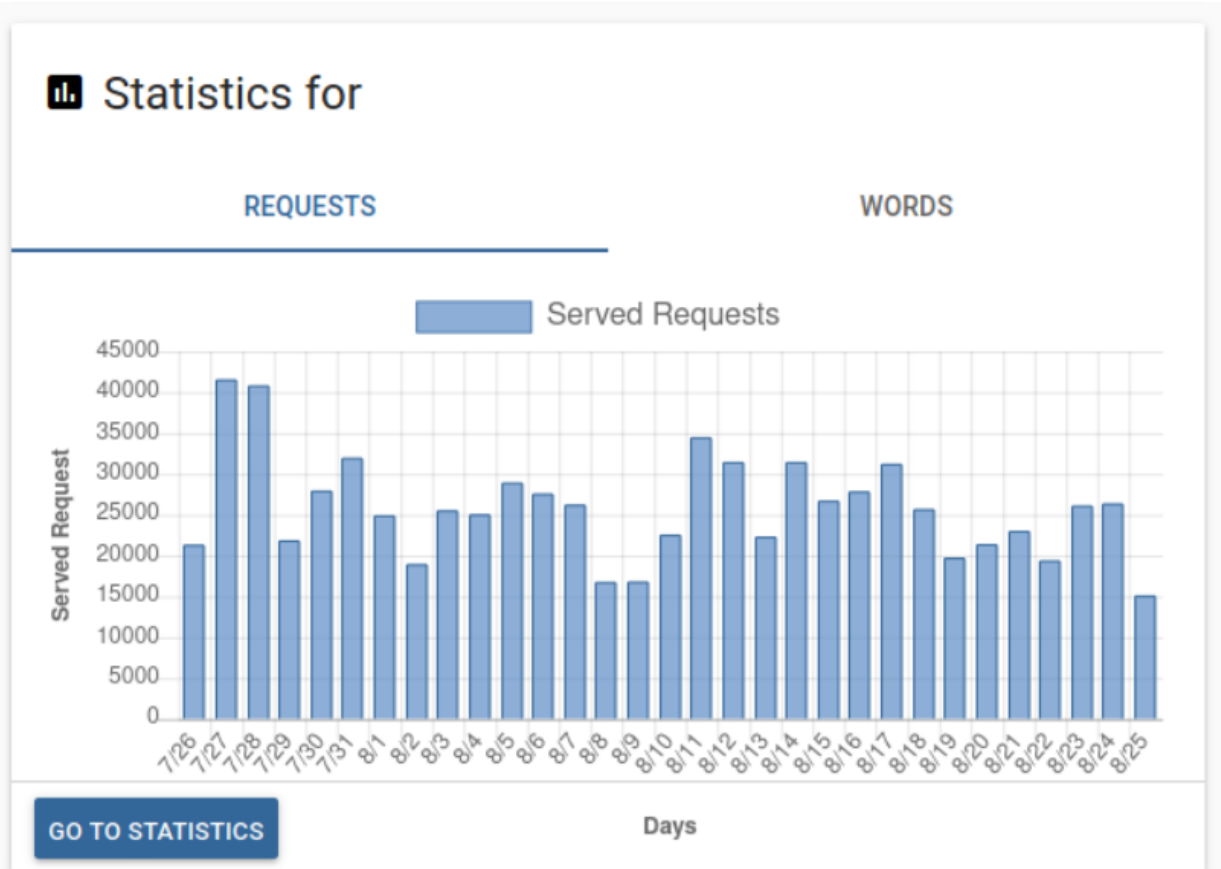
You can also use this card to delete or remove the project. Only the owner can delete a project completely. If you were simply invited but aren't the owner and you 'remove' a project, you will simply be removed from it.

3.1.2 Project wallet



This card shows you information on the wallet of the project you are viewing. In most cases, this is the wallet of the user who created the project. You can see if it is expired and the remaining balance and let the user know if they need to top up.

3.1.3 Statistics



Statistics

Here you can see the number of served requests and words added to your project recently. For more information, click the 'Go to statistics' button.

3.1.4 Miscellaneous options

Miscellaneous options

Project workflow: Simple Translation Workflow

Gateway domain:

Segmentation: ☒

Manual publishing: ☐


Misc

options

On this card, you have a couple of options that are the set once and forget kind. You may need to reference these settings later so it's handy that they are present here.

- **Project workflow:** Change the number of project participants and project workflow type using this dropdown. See [Collaboration](#) for more information.
- **Gateway domain:** If the site you wish to translate has a strict firewall that blocks the proxy, the admins can set up a gateway for you. In this case, all of the requests go through that gateway and will arrive to the origin server from a fixed IP address that can be whitelisted. Here you can see whether the gateway is enabled.
- **Segmentation:** Whether the segmentation rules are applied to new segments that are added to the project. As this has the potential to break the project, not even the owner can change it. If you wish to have it disabled, please contact the support team about it.
- **Manual publishing:** It is an advanced project control feature that gives project owners the ability to hold back the translations from being published on the live page (but not the preview, as it will always display the latest translation available!) until further notice. Once activated, a new item will appear in the Bulk Actions menu of the Workbench, "Publish". Selecting this action will cause all selected segments to be synchronized with their displayed translations, and once the action finishes, the markers in the status bar on the right of the entries will change to reflect this.

3.1.5 Crawl settings



Crawl settings

☐
Process pages in source language only (requires Google language detection API)

Ignore queries:

wpv_view_count, fbclid, gclid, jsessionid, t, mrkto_*, PHPSESSID, msclkid, _g

Group pages by ignoring query parameters:

Matching source segments are merged across pages in the group

Crawl

settings

Settings that affect the way crawls work:

- **Process pages in source language only:** For every segment it encounters, the crawler uses Google's language selector API to determine whether it is in the selected source language. As this is a paid API, you must supply a key to use it.
- **Ignore queries:** By default, the crawler treats `example.com/?query=example1` and `example.com/?query=example2` as different pages. Adding query here will prevent this behaviour. It will treat the 2 pages if they were exactly the same.
- **Group pages by ignoring query parameters:** The crawler will visit both `example.com/?query=example1` and `example.com/?query=example2` but segments will be treated as coming from the same page (`example.com` in this case).

3.1.6 Staging domains

Staging domains

Domain	Default staging	Live	Actions
example.com	✓		⋮

Staging domains

Although it is true that the project address cannot be changed after the project is created, the Staging domain feature can still be used to change the origin server to which requests are sent.

For details, please see the Cookbook recipe [on Staging domains](#)

3.1.7 Documentation

Documentation

[Get started with the Translation proxy](#)

[Learn more about the Dashboard](#)

[Learn more about the Workbench](#)

Docs

A set of links pointing here so that it's always at hand.

3.1.8 Crawl info

 **Crawl info**

 **Wordcount (Discovery)**
Done

Started

August 25, 2020. 14:24:08

Finished

August 25, 2020. 14:24:11

Took

0:00:03

Visited

1 pages

No more pages left.


[GO TO CRAWL HISTORY](#)

Crawl

info

Some information on the latest crawl ran on the project. It is the same info you'd find on the right side of the [Crawl list](#).

3.1.9 Linked projects

 **Linked projects**

When navigating from one project to another, if projects are linked, the appropriate translation will be displayed to the visitor. **Make sure you publish all the projects linked together, for all the target languages.**

Add

Linked

projects

If content is coming from multiple domains, like `example.com` and `blog.example.com`, you'll need to create multiple projects to translate them. Linking them **enables the cross-mapping of URLs**.

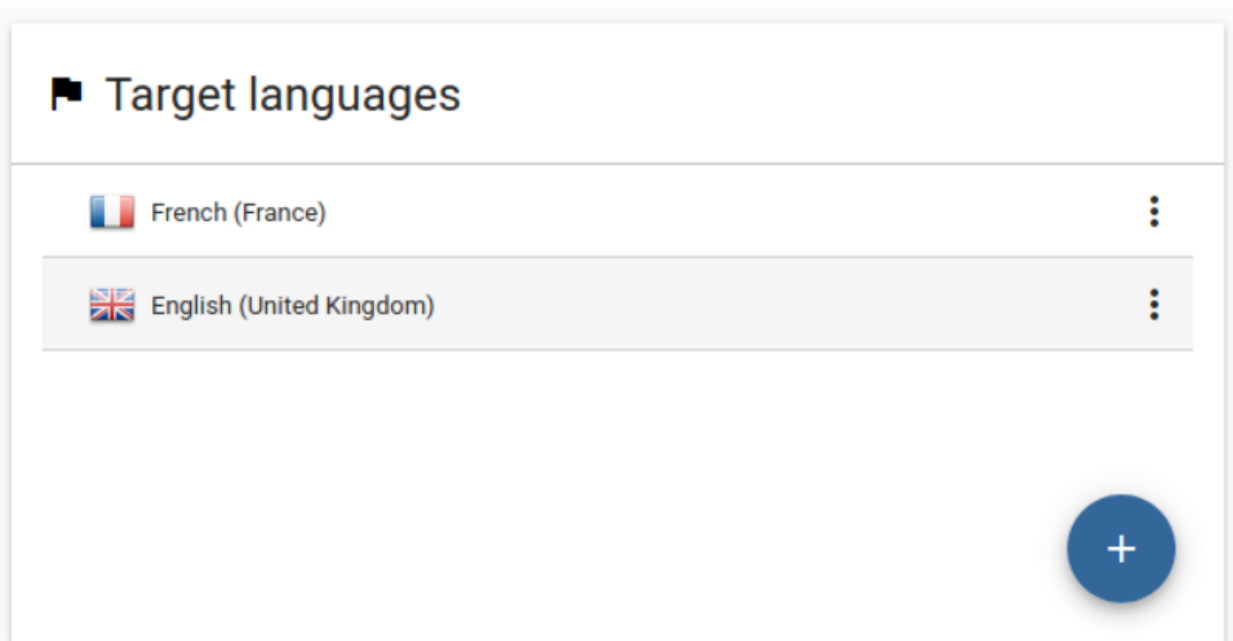
A few restrictions apply to project linking:

- project linking is **two-way**: project A has to be linked on project B and then project B has to be linked on project A to complete the process.
- linked projects have to be **published together** and with the **same target languages**.

Form solutions tend to use content from external domains. The localization of **HubSpot forms**, for example, is a frequent and somewhat challenging use case of project linking.

The topic of translating such content using proxy features only is covered in detail [on this cookbook page](#).

3.1.10 Target languages

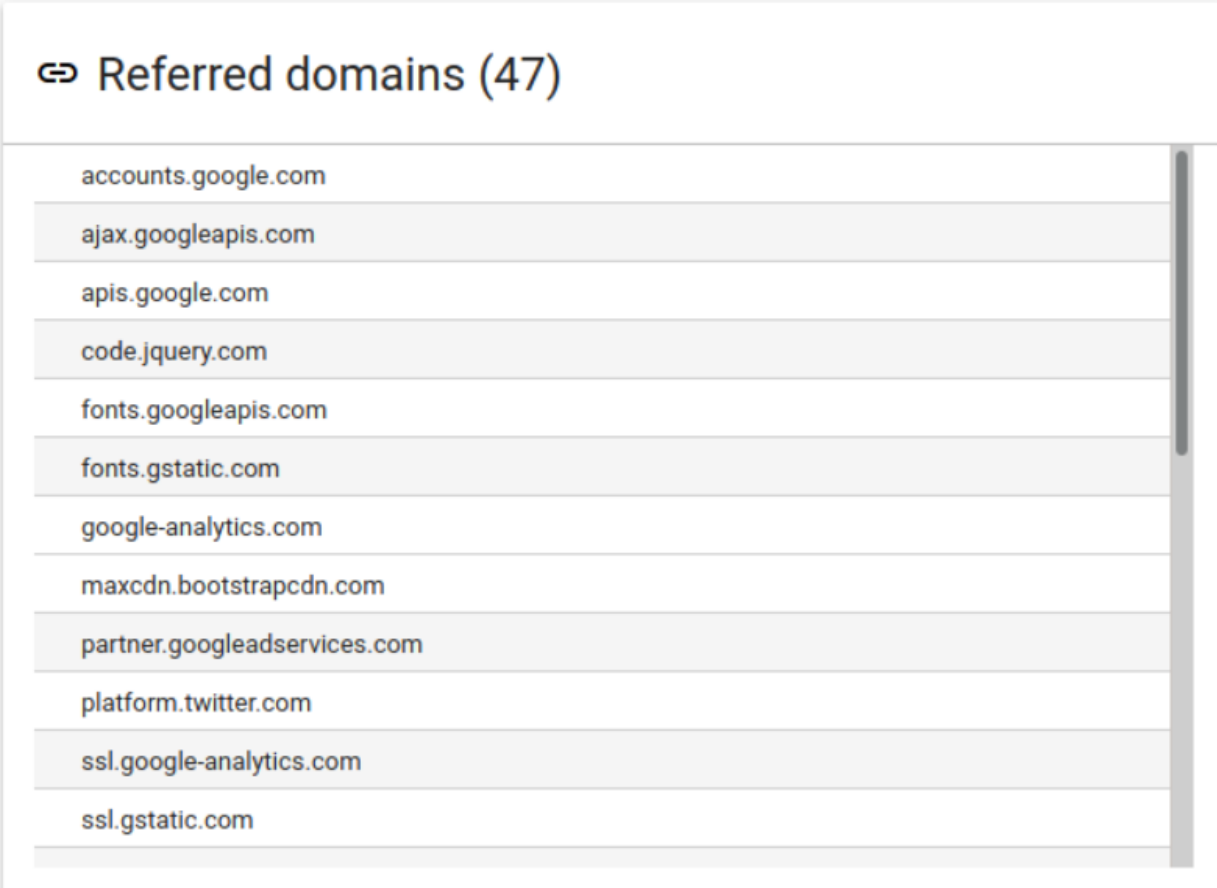


languages

Target

Here you can see the current target languages and add more if you wish to.

3.1.11 Referred domains



↻ Referred domains (47)
accounts.google.com
ajax.googleapis.com
apis.google.com
code.jquery.com
fonts.googleapis.com
fonts.gstatic.com
google-analytics.com
maxcdn.bootstrapcdn.com
partner.googleadservices.com
platform.twitter.com
ssl.google-analytics.com
ssl.gstatic.com

Referred

domains

This is a list of the external domains that the Crawler encountered. These are sites that the original site relies on (e.g. to embed YouTube videos, maps or for analytics) or links to.

3.1.12 Content timeline

Content timeline	
June 23, 2020. 17:03:49	Project export
June 23, 2020. 17:03:49	Project export
June 23, 2020. 17:03:49	Project export
June 23, 2020. 17:03:49	Project export
April 16, 2020. 11:00:20	Project export
February 17, 2020. 12:34:48	Project export
February 17, 2020. 12:29:30	Project export
January 24, 2020. 15:19:19	Crawl (Content extraction) (40)
November 21, 2019. 16:48:39	Crawl (Content extraction) (39)
November 21, 2019. 11:08:00	Crawl (Content extraction) (38)

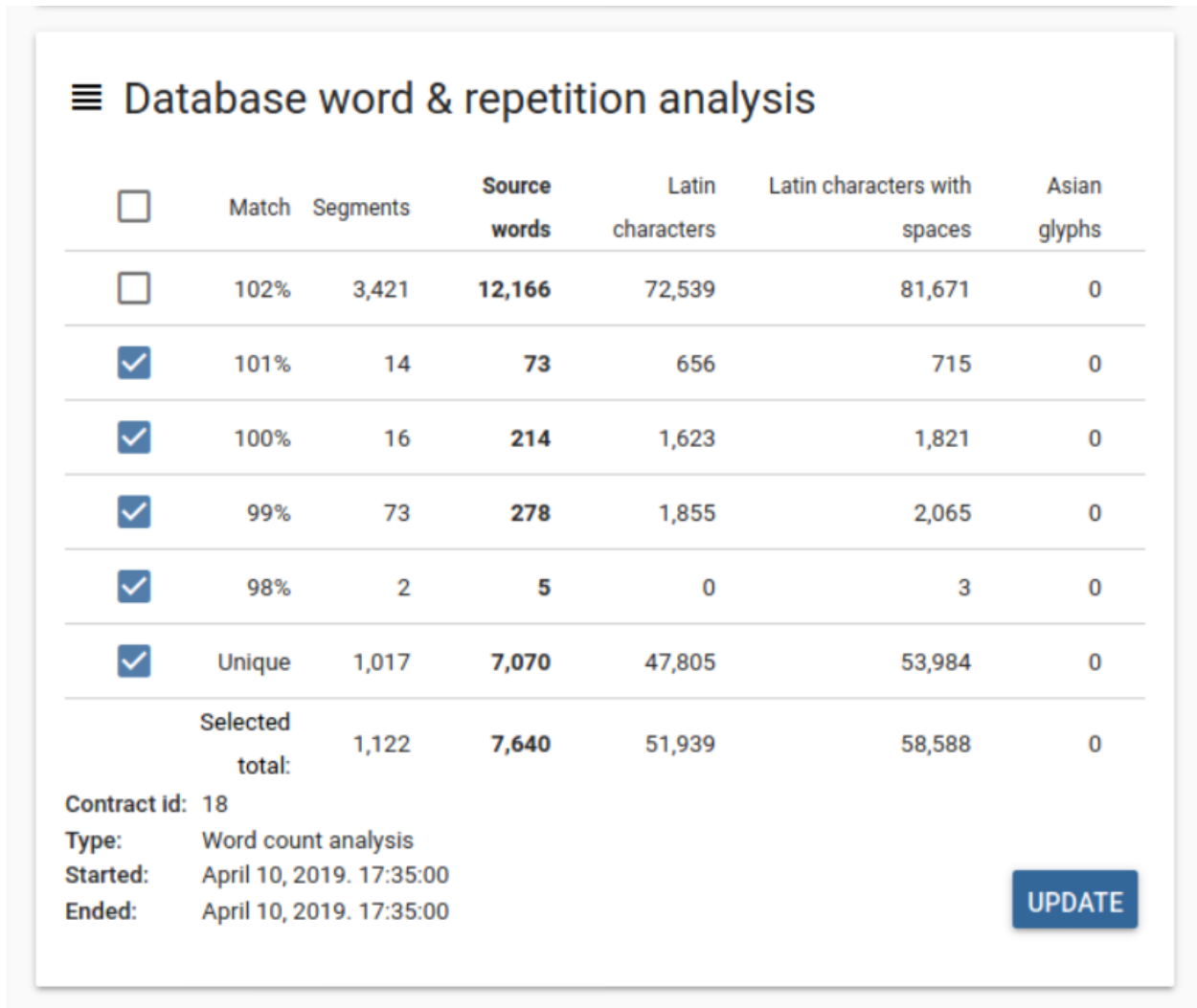
ADD NEW EVENT

Content

timeline

This list contains the main events where content was manipulated on the project. These include crawls, content exports and imports as well as the creation of Work packages.

3.1.13 Database word & repetition analysis



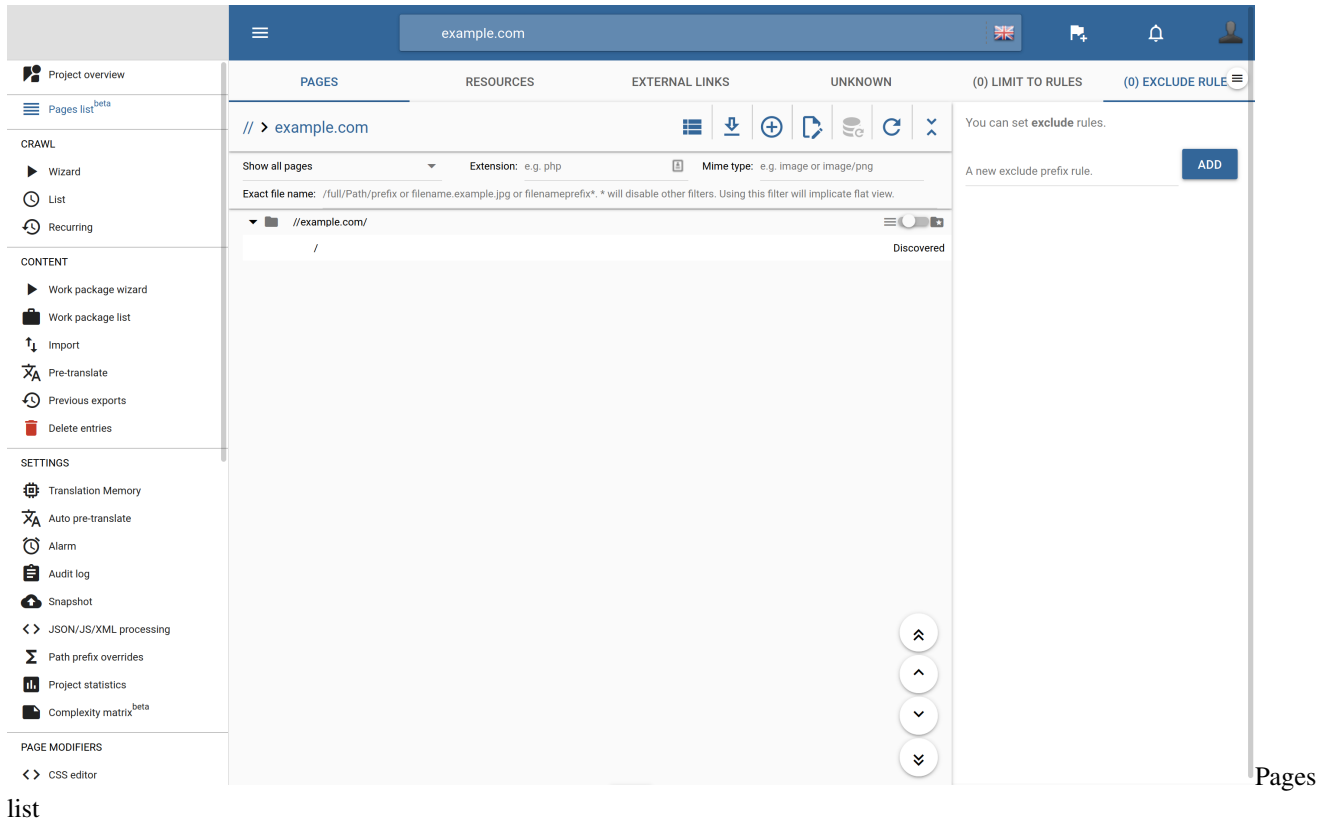
Database

word & repetition analysis

Here you can see the number of words and repetitions on the site as found by the crawl specified on the bottom. Note that the data may be outdated if the crawl is outdated. You can use the 'Update' button to fill the chart with fresh data (or old, if you wish).

3.2 Pages list

The page list lets you keep track of the various URLs that are (or were) seen by the proxy. Aside from the overview of and direct access to content on the individual pages, the page list also allows you to visit the various proxy preview domains, fine-tune inclusion/exclusion settings and check the translation progress of individual pages.



list

This menu has two main parts visible when you first open it: the “Pages, resources, external links & unknown” pane and the include/exclude rules.

Also notice the language selector on the right of the project selector. Information in the translation progress bars will refer to the selected language, and both the List and Highlight Views will open for the selected target language. The same is true for the various types of Preview links.

Make sure that you have selected a target language to gain access to many powerful features in the page list. If you do not (or there is no target language on the project at all), neither the Workbench links, nor the Previews will be available, and most of the context menu options will be greyed out.

3.2.1 Pages, resources, external links & unknown

Here you can see all the items that the proxy is aware of (either because of crawls or user visits). They are split into 4 categories: pages, resources, external links and unknown.

Pages are links that contain translatable content. By default, these are the HTML pages on the project but you can move others (like JSONs) from the other sections.

Resources are images, PDFs and similar that the proxy can’t automatically extract content from. However, you may want to localise these by uploading a replacement.

External links (like those pointing to YouTube videos) can be replaced (e.g. if the video is available in the target language). Note that if you are working on a project for `www.example.com`, `blog.example.com` is considered *external* to it

There should be no entries under Unknown but if you can’t find something you may want to check it here.

Above the list itself, there are multiple actions you can do and filters. These are:

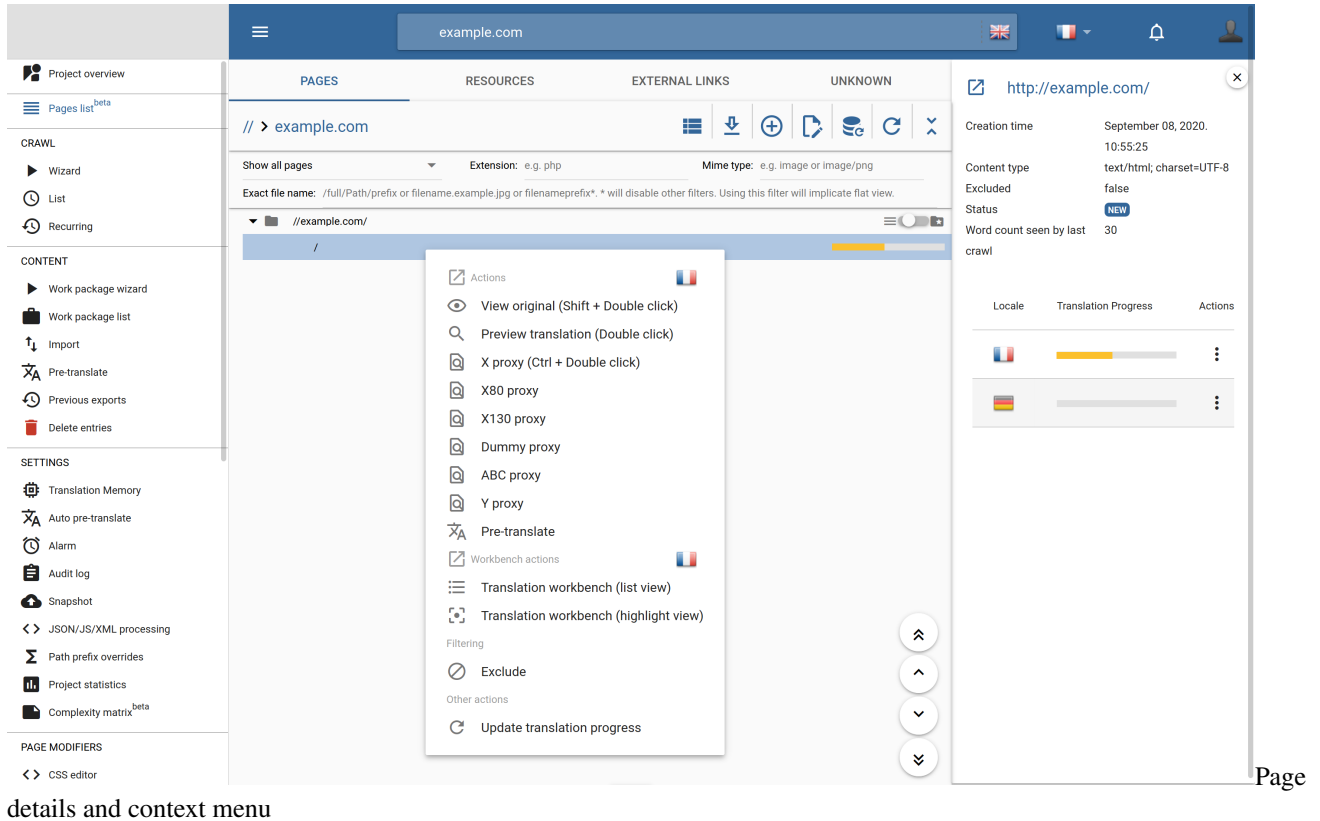
- **Switch to flat/folder view:** Switch between the flat view (default for resources), which is just a list of URLs similar to the old Dashboard and the folder view (default for pages) that allows you to see the pages as if they were files on your computer. External links are only available in flat view so this option is hidden for them.
- **Show/hide preview** (only for resources): Whether to load the images or just display their filename. If you have a slow connection or a data cap, we recommend hiding these.
- **Export URL list:** Create a CSV or Excel export of the current list. You can filter the links by translation state or free text search. Due to Excel's limitation on the number of lines per sheet, a CSV export is recommended for large-scale projects.
- **Add new:** Run a *page-specific crawl*. Use it to add new pages or refresh already extracted pages in a “surgical” fashion. HTTPS URLs are added using fully qualified URLs. If you specify the pathname only, such as `/about-us.html`, the proxy will default to requesting the page over HTTP. The same applies to any resource that is available via both HTTP and HTTPS: add both URLs to ensure that you have all possible versions of the page or resource.
- **Mass page exclusion:** Enter a list of URLs that you wish to exclude. Use this when you can't add an exclusion rule. You also have the option to “un-exclude” aka include some pages again.
- **Update translation progress:** Updates the bars showing the progress. This can take from a few seconds to a few minutes of time to finish, depending on the amount of content to be aggregated.
- **Reload:** Reloads the list of pages.
- **Show/hide filters:** Controls the filters discussed below.

Note that some of them are only available for either of the types.

Below these actions, you can search by file name or filter pages, resources and external links

- By their inclusion status,
- Their extension (especially useful for resources)
- Their [MIME type](#). Note that the extension of a file isn't necessarily the same as its mime type.

Page details and options



details and context menu

Clicking on a page in the list reveals further information about it on the right side. Here you can see basic details such as the time it was first seen, its content type, status and word count. You can also see the translation progress for *all* languages (the progress of the currently selected language is displayed at all times).

There are a number of states that a page can be in throughout the lifetime of a project:

- **Unvisited:** Indicates pages that were already seen in the source of another page, added to the list, but for one reason or another, not visited by a crawler. For example, if you run a Discovery with a page limit of 8 and allow it to collect new URLs at the same time, it will likely visit the landing page, “/”. The landing page is likely contain many references to various pages on the same domain. The Discovery will exit after 8 pages, but also add the rest as “Unvisited”.
- **Discovered:** Pages that were included in a word count previously are said to be Discovered. The content on them is accounted for in at least one of the word count statistics, but the translatable content **has not been extracted for translation** yet.
- **NEW:** “New” in this part refers to the fact that the page’s content is extracted and in a “pristine” state: no entries on it have been translated yet. It is usual to find pages in “NEW” state right after the first content extraction crawl (when a project has all the content ready for translation, but no translations have been provided yet).
- **Progress bar:** will be shown when a non-zero percentage of the entries that the given page refers to have been translated. Note that due to automatic propagation, translating one entry can influence the progress bars of multiple pages). Hovering over the progress bar will show more specific numbers in the tooltip, but progress is also color-coded as follows (end value decimal places are omitted):

- progress < 30% - red
- progress < 60% - yellow

- progress < 100% - green
- progress = 100% - blue

You can right-click on a page to display page-specific actions. Again, depending on context, some of them may not be available.

- **View original:** Load the original page in a new tab.
- **Preview translation:** Load the Preview. Use this mode to ensure that everything is translated.
- **X proxy:** In this mode, all content that the proxy can pick up is replaced with `xs`. This mode can be used to find content inserted via JavaScript and such because these won't be `xed` out.
- **X80 & X130 proxies:** These mode behave like the `x` proxy but simulate different lengths of text by shortening or expanding it to 80% and 130% respectively. This mode is useful to detect layout anomalies that arise from text expansion.
- **Dummy proxy:** Displays the page as received from the original server. Use this to debug issues where you suspect the original server treats the proxy differently than other users (e.g. the firewall blocks it by IP).
- **ABC proxy:** A special, improved variant of the `X` proxy. In this mode, text is replaced with either `x`, `a`, `b` or `c` depending on its status.
 - `x`: Content that can be translated but wasn't scanned yet.
 - `a`: No translation is available yet but the segment was scanned and is ready for translation.
 - `b`: The translation is the same as the source *or* it was translated but the translation was removed.
 - `c`: The segment is translated.

We hope to further improve this mode. If you have any suggestion, please don't hesitate to contact support.

- **Pre-translate:** Brings up the pre-translate dialog for this specific page.
- **Translation workbench (list or highlight view):** Brings up the Workbench which is an online computer assisted translation tool. List view is similar to what you'd find when using MemoQ. A list of segments and their translations side by side with suggestions from the TM on the right and an editor near the bottom. Highlight view is similar but instead of the segment list, the page is loaded and you can hover on the text that you need to translate. This allows in-context translation.
- **Exclude/select:** Allows you to mark the specific page as translatable or excluded.
- **Update translation progress:** You can update this progress for individual pages if you wish.
- **Mark for replacement:** Once an image is marked as translatable, a new Manage translations button becomes available. After clicking it, you can upload localised versions of the image that will be displayed on the translated sites.
- **Add to resources for link replacement:** Allows you to move an external link to resources so that you can process it as such.

Page deletion

A project, especially one with few inclusion/exclusion rules or one that uses URLs with many query parameters can over time accrue many pages in the page list. Page deletion is now available as a user feature.

By clicking on the **Trashcan** icon on the *Pages list* page, a pop-up window shows up.

Delete pages

Delete from ☒ Pages ☐ Resources ☐ Unknown

```
^https://example.com/foobar.html$
foobar.html
foobar
pdf
\.img
```

☐ Dry run: preview the URLs to be deleted
CANCEL DELETE

Delete

pages pop-up

There is an option to select what kind of content you want to delete, *pages*, *resources*, and *unknown*. The default option is *pages*, which will only look for pages matching the regex.

The most important part of the process is defining the right **regular expression (regex)** for your needs. (You can find an online regex tester [here](#).)

Once you have your regular expression ready, by checking the checkbox *Dry run: preview the URLs to be deleted* and click the **Preview** button, you can run a preview of the pages that are about to be deleted to avoid incidents like deleting pages by mistake, and you can also check if the pattern matching found each and every page you intend to get rid of.

Once you made sure you want to delete all the pages in the preview, just uncheck the checkbox *Dry run: preview the URLs to be deleted* and hit the **Delete** button.

You will get notifications through the process. Once the page deletion is ready, you can refresh the pages list, and the desired pages should be removed.

Deleting pages from the project does not prevent them from being re-added, at least not in itself (for that, additional configuration is needed that actually addresses the root cause of the page being added, for example ignoring query parameters or excluding pages from the Crawls).

3.2.2 Include & exclude rules

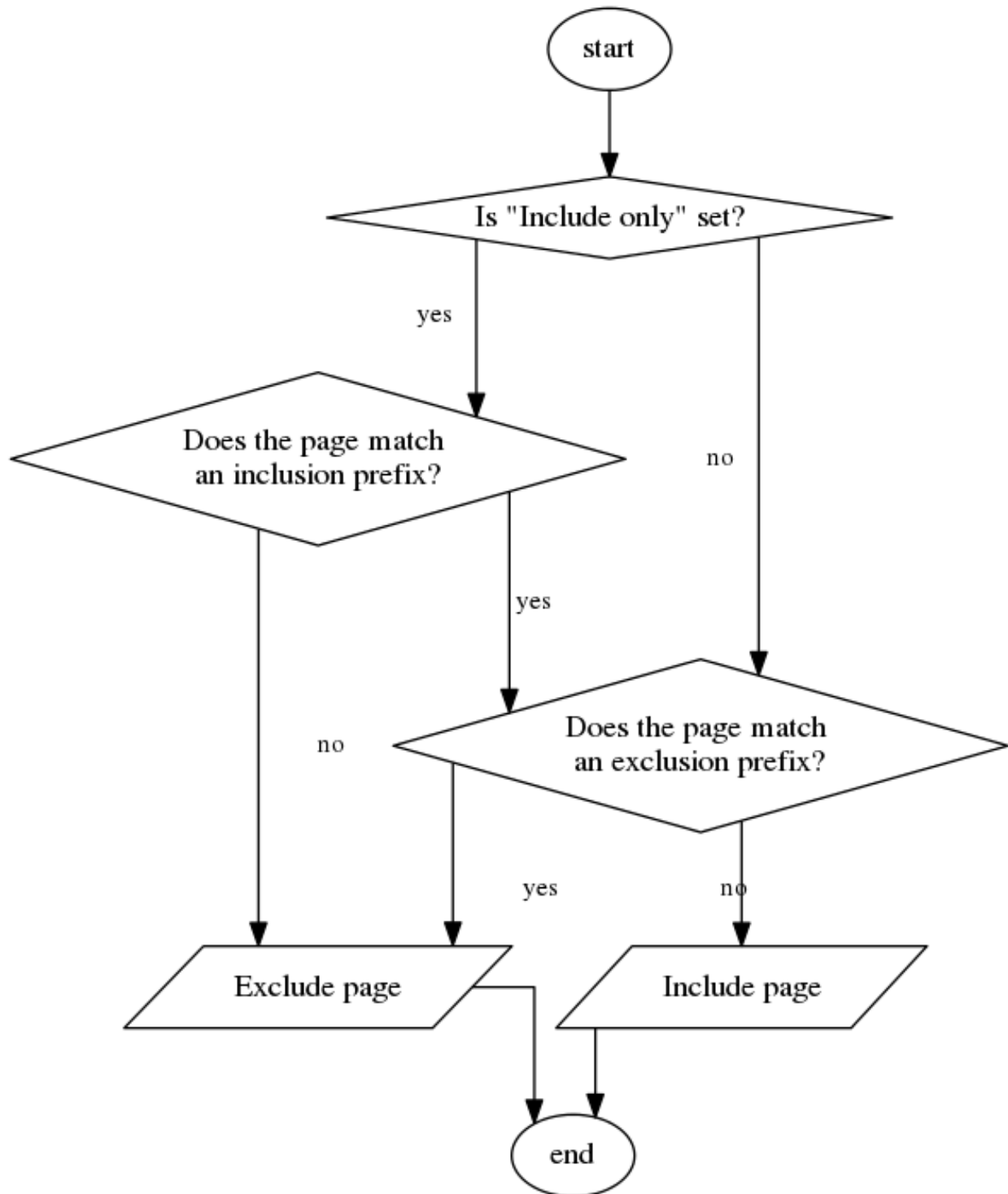
Inclusion rules let you set the **scope of translation**, that is, the list of pages that the crawler is allowed to visit. You can enter the prefixes that you wish to limit the scope to (or exclude from it). You can also exclude individual pages should you see fit.

The Dashboard has a number of features that support inclusion/exclusion prefixes, such as Auto-pretranslation or Work packages. However, the rules you specify here are special and powerful: they have influence over the entire project. An excluded page will stay *untranslated* over any proxy domains (both Preview and Live), and excluded pages are ignored by the crawler.

NOTE: You can enable/disable the application of rules using the checkbox associated with them or you can delete them completely.

Rule Application

Whenever the proxy comes across a page in an affected context (crawling, serving, translating, etc.), it will evaluate it according to the rules you provide. This process is summarized in the flowchart below:



Inclusion/Exclusion Evaluation

Evaluation

of Inclusion/Exclusion Rules

A few points of note concerning inclusion/exclusion rules:

- Path names are first checked for inclusion, then exclusion.
- A path name has to match only one from the set of inclusion rules. Each such rule is applied to the path in

sequence until a match is found or there are no more rules.

- The rules are **strings** and matched from the beginning of the pathname. The proxy does not analyze them in detail or produce complex internal representations.
- Query parameters are supported (but be careful, you can't really count on a query parameter to have a set position!).
- If a path falls outside of the scope of your inclusion rules or an exclusion rule applies to it, it will be greyed out in the page list and the text "Excluded by rule" will be visible next to it.
- Paths that are excluded by rule can't be included using the context menu. You need to edit your rules if they gobble a path that you want to include.
- Manual page exclusions overwrite all other inclusion rules.

NOTE: If all pages are excluded on a project, crawls *cannot be started*, even if the currently active rules would allow for the inclusion of *some, as-of-yet undiscovered page*. In this case, crawls will exit after 0 pages visited. You have to ensure an entry point: that at least one of the known pages is in an included state. Otherwise, the crawler can't set its foot in the door.

Though it may seem nonsensical to exclude every single URL on a project, we note this unusual case because it can come about from inadvertent use of *inclusion rules*.

Consider, for example, that if you set `/en/` as the sole "Include only" rule on your project, but no page starting with `/en/` is in the page list, then not a single valid entry point is provided to the crawler.

3.3 Crawl

3.3.1 Crawl types

Presumably, one of the most important tools for your working process is the crawler, since it allows you to map out sites by running a *Wordcount (Discovery)* for proposals for instance or start *Content extraction (Scan)* for translation. There are two more crawling options, and we will discuss these as well in this article. You can set up crawls using the [Crawl wizard](#).

You can also read a more technical description of the crawler [here](#).

Wordcount (Discovery)

Before you can translate a website, you need to extract the content from it using the Content extraction (Scan) function. Scans, however, write to the database, which, as you know from our pricing model, will cost you money. In that sense, it would be risky to let loose an unlimited Scan on a website you're not familiar with. To forego Content Extraction and limit the crawler's actions to finding out about the structure and word count of a given web site, a more tentative approach is needed: a Discovery.

The process maps the structure of the site by scanning each page for link elements and trying to follow these links. The content of the website is not stored, only the URL address of the visited pages, together with their status info. Any page that is verified to exist is marked as Discovered, and the ones that returned an error message (most commonly redirection (HTTP301-302) and page not found (HTTP404)) are marked *Unvisited* in the page list. For more information on HTTP status codes, see [here](#). Although this process doesn't extract content, it provides a preliminary wordcount and a repetition rate as well. It has a cost of 1 EUR or 1.2 USD per thousand pages.

IMPORTANT! If you exclude pages during discovery, changing the rules only will not include them in the new discovery; you have to add them manually through the "Add pages" dialogue.

Your primary choice is between limited and unlimited Discoveries.

A limited Discovery is the best way to “get to know” a website - if you set a reasonable safety number, such as the default 100, you can be sure the process won’t go overboard when it finds an enormous forum or a structurally complicated web-shop. Based on what a limited Discovery tells you about a site, you can set prefix exclusion rules in the page list or cherry-pick unnecessary pages for exclusion. By increasing the page limit and running Discoveries in succession while adjusting your inclusion/exclusion rules, you can work your way through a site structure incrementally. Of course, if you have a clear idea of the size of the site, you can always set up an unlimited Discovery, wait for it to finish and set up your exclusion rules afterwards. We don’t recommend running unlimited crawls. It’s always safer to add a large limit instead.

Once the discovery is ready, you will receive an e-mail notification and the statistics will show up on the *Crawl list* page. Based on this you can give a rough estimation for the website translation cost - both in time and money.

Content extraction (Scan)

The main difference between Wordcount (Discovery) and Content extraction (Scan) is that Wordcount does not extract or store content. They are used for statistical purposes, such as providing a word count or mapping a site’s URL structure.

A Content extraction (Scan), however, writes into the database. It extracts and stores the source content to be translated. New source words added to the database are billed, therefore care must be taken when starting Scans. If you are not really sure about a site, stick to Discovery until you gain a better understanding of its structure! Unlimited Scans especially require attention: they will relentlessly add all content to the database according to the specification you set for them and are likely to only stop when your wallet is depleted.

Settings you used to experiment with Discoveries can also be used to initiate Scans - for a detailed explanation of the different settings, please take a look at the menu called *Crawl Wizard*.

There is a card called *Database word & repetition analysis* on the *Project overview* menu, which shows the number of words written into the database and will only change when you Scan the page.

New content detection

New Content Detection is basically a hybrid between the Wordcount (Discovery) and Content extraction (Scan). It runs through the set of pages selected for translation, but it does not store content at all, it’s just providing a wordcount statistics based on pre-existing source entries.

TLS Content extraction (Scan)

If you select the tweak called *Target language specific content* in Advanced settings, then you will be able to run a *Target Language Specific Content extraction (Scan)* in Crawl wizard.

When the translatable site has various content by target languages, and the remote server would serve the requests based on the given locale codes, you can ingest the content with this type of crawl. The proxy uses the `X-TranslationProxy-Translating-To` header during the crawl, containing the four-letter locale code used for the request, so the remote server should process this header, and serve the request based on the provided locale code.

3.3.2 Crawl wizard

With the *Crawl wizard* tool, you can configure and run a crawl. In this article, we will discuss the configuration options, the results and statistics, and also the possible reasons behind a failed crawl.

STEP 1 - Type

1

2

3

4

5

Type

Scope

Pages

Fine-tune

Summary

Wordcount (Discovery)
€1/1000 pages

Count the number of words and repetitions across the visited pages. No translatable content is stored.

Content extraction (Scan)
€1/1000 processed pages+
€10/1000 unique words

Count words, plus store any new translatable text in the Cloud Database for translation or exporting.

TLS Content extraction (Scan)
€1/1000 processed pages+
€10/1000 unique words

Target Language Specific content extraction. Count words, plus store any new translatable text in the Cloud Database for translation or exporting.

New content detection
€1/1000 pages

Count the number of words and repetitions inside any new content across the visited pages. No translatable content is stored.

Shortcuts

* Recommend for pro users

☐ Start a Discovery, look for new pages, count the words but don't store anything. Page limit: _____

☒ Extract new content from existing pages only (~170)

NEXT

Crawl

types

There are four types of crawls, but most of the time, you probably will see three of them as a selectable option in your crawl wizard.

Wordcount (Discovery): The process maps the structure of the site by scanning each page for link elements and trying to follow these links. The content of the website is not stored, only the URL address of the visited pages, together with their status info.

Content extraction (Scan): It extracts and stores the source content to be translated. New source words added to the database are billed, therefore care must be taken when starting Scans.

New content detection: It runs through the set of pages selected for translation, but it does not store content at all, it's just providing a wordcount statistics based on pre-existing source entries.

TLS Content extraction (Scan): When the translatable site has various content by target languages, and the remote server would serve the requests based on the given locale codes, you can ingest the content with this type of crawl.

You can read more about the selectable crawl options [here](#).

STEP 2 & STEP 3 - Scope

Progress bar: 1. Type (checked), 2. Scope (active), 3. Pages, 4. Fine-tune, 5. Summary

Option	Description
Re-visit current pagelist & Find new pages	Find new pages, re-visit existing ones, up until limit is reached.
Current page list only	Revisit the pages in the project page list only. Don't add or process any new URLs.
Specific pages only	Add/select specific URLs or sitemap.xml

Buttons: BACK, NEXT

Crawl

scope

Re-visit current pagelist & Find new pages: The crawler visits the existing pages first, and during this process, it also collects new pages, which will be visited after the current pagelist. In the next step you can set a page limit, but please note, that if this number is smaller than your pagelist, the crawler won't visit new pages.

Current page list only: New pages won't be added or processed, just the pages that already exist in your project. In the next step, you can check this list.

Specific pages only: By selecting this option, you can specify a specific URL list or sitemap (in .xml format), and in this case the crawler will visit and process only these pages, unless you selected *JS*, *CSS resources* and/or *images*, *binary resources* as collectable resources, because in this case these will be visited as well, and added to *Resources*. If you select the option *Also add new URLs not in the list above, if referred, but as "Unvisited"* (takes precedence over *Page Freeze setting!*), and the specified pages contain links, then those will be added to the pagelist as *Unvisited*.

STEP 3 - Pages

The content of this step will change based on the previously selected *Scope* option.

If you select the *Re-visit current pagelist & Find new pages* in STEP 2, then you can limit the number of pages:

✓ Type ✓ Scope 3 Pages 4 Fine-tune 5 Summary

Page Limit

100

- URLs in the page list are visited first
- new URLs over the Page Limit will show up as 'Unvisited'
- Leave empty for an unlimited crawl (**not recommended**)

BACK NEXT

scope

If the selected option is *Current page list only* in STEP 2, you will see this list:

✓ Type ✓ Scope 3 Pages 4 Fine-tune 5 Summary

Page list

- /
- /about/
- /prices
- /pricing/
- https:/
- https:/2011/11/
- https:/2012/06/
- https:/2012/09/
- https:/2012/10/
- https:/2013/02/
- https:/2013/03/
- https:/2013/06/

BACK NEXT

scope

Selecting option *Specific pages only* in STEP 2 enables you to add the pages you would like to scan, or you can also provide a sitemap:

✓

Type

✓

Scope

3

Pages

4

Fine-tune

5

Summary

Page Limit

100

- URLs in the page list are visited first
- new URLs over the Page Limit will show up as 'Unvisited'
- Leave empty for an unlimited crawl (**not recommended**)

BACK

NEXT

scopeCrawl

STEP 4 - Fine-tune

Origin (Source) snapshots

✓ Type ✓ Scope ✓ Pages 4 Fine-tune 5 Summary

ORIGIN (SOURCE) SNAPSHOTS COLLECT RESOURCES LIMIT TO/EXCLUDE RECURRENCE SESSION COOKIE

Origin Snapshots

Snapshots let you store and later serve the source site's contents in a **fixed state**: as they're encountered during this crawl. The snapshot contents **will not change even if the original site updates** until you update them in a future crawl.

[Origin \(source\) snapshot is disabled. Change settings](#)

BACK NEXT

Snapshot

You can create and enable a new [Origin Snapshot](#) in the Snapshot menu in Dashboard 2.0. Note that you need to run a *Content extraction (Scan)* or *TLS Content extraction (Scan)* for content to build the Origin Snapshot before it becomes functional. In this step, you can configure the origin snapshot.

Reuse existing pages and store new pages: In this case, you instruct the Crawler to skip the pages that are already in the Snapshot you choose and thus reduce the building cost. The choice of whether or not add new pages gives you the possibility to simply ignore the new pages that were added to the source site.

Reuse existing pages and don't store new pages: Selecting not to add new pages allows you to build a snapshot that has the pages updated but no new ones added. This option is useful if you made changes to the JSON Paths and as a consequence need to rebuild the Snapshot. As a result of building the Snapshot, all content that was set to be picked up by the *Scan* is added to the current Snapshot.

Don't reuse existing pages, update/store all encountered pages: Choose this option when it is the first time you build an origin (source) snapshot or if you want it to reflect the state of the site at the time of crawling.

Collect resources

resources

Collect new HTML pages: Any pages the selected crawling process finds during a crawl will be added to the page list. Note that this is an explicit action by the user, therefore the Page Freeze setting does not affect in this case.

Collect JS, CSS resources: Instruct the crawler to add newly found Resources to the project. The proxy makes collecting Resources a trifle, and there are a plethora of in-depth methods to translate content embedded within them. But the translation of Resources is also known to be one of the trickiest parts of website localization. Beware, and woe betide the unready!

Collect images, binary resources: By enabling this option, image or any binary content found on the site is added to the Resources screen, where you can add localized counterparts and instruct the proxy to serve those in place of the original.

There is often a great deal of image content on websites, so indexing all of them can take a very long time to finish; it is a good idea to turn this option off the first time around, and enable it only later, when it is clear that image localization is also going to play a part.

Also collect resources from external domains: Allows you to pick up resources that are linked to on the site.

Collect short links: WordPress sites tend to give a shorter link to every page in the form of `https://www.example.com/?p=123456`. In most cases these are safe to ignore as they merely duplicate content available on other links but due to misconfigurations, they can contain content that you want to ingest.

We recommend that you only select Collect new HTML pages as that's where the majority of the content is likely to be. Resources that you need to localise can be added later using the pages list. This allows you to keep the Pages list as short as possible and thus improve manageability and reduce the time crawls need.

Skip content-type check if already determined: Enabled by default since it only applies to already crawled pages.

Use ETAGs from last crawl: Disabled by default. You can read more about ETAGs [here](#). By enabling this option you could reduce remote server load on subsequent crawls, but also note, that skipping pages could lead to invalid crawl wordcount.

Limit number of simultaneous requests Control the number of requests a crawl is allowed to send to the original site simultaneously. Use this function to prevent the crawl from flooding a smaller server with requests.

Limit to / Exclude

The screenshot shows the 'Limit to / Exclude' configuration page. At the top, a progress bar indicates the current step is 'FINE-TUNE' (step 4 of 5). Below the progress bar, there are five tabs: 'ORIGIN (SOURCE) SNAPSHOTS', 'COLLECT RESOURCES', 'LIMIT TO/EXCLUDE' (selected), 'RECURRENCE', and 'SESSION COOKIE'. The 'LIMIT TO/EXCLUDE' tab is active, showing two sections: 'LIMIT TO (0)' and 'EXCLUDE (0)'. The 'LIMIT TO' section has a text input field with the placeholder 'Limit crawl depth to' and an 'ADD' button. The 'EXCLUDE' section has a text input field with the placeholder 'Exclude'. At the bottom right, there are 'BACK' and 'NEXT' buttons.

Crawl

limitations

Limit to rules: You can set prefix-based rules here, which will temporarily set the scope of the crawl. For instance, if you add a single inclusion prefix `/en/`, the crawler will only process pages that can be found in the `/en/` directory.

Exclude rules: Set prefix-based rules, which will temporarily exclude the given directories from the scope of the crawl.

Limit to: There is more than one way of limiting the scope of a crawl. In scenarios where you have to crawl richly interlinked pages (a wiki, for example) you can use the crawl depth to limit how deep the crawl should be allowed to follow links.

Exclude: You can specify any regex to apply to all URLs and ignore the given page if there is a match. This is a user- and crawl-specific setting. As always, we recommend that you test your regular expressions on regex101.com before using them. In this case, you can export your page list from the pages list on the Dashboard 2.0 and use that list as a test string for your regex.

Recurrence

✓

Type

✓

Scope

✓

Pages

4

Fine-tune

5

Summary

ORIGIN (SOURCE) SNAPSHOTS

COLLECT RESOURCES

LIMIT TO/EXCLUDE

RECURRENCE

SESSION COOKIE

Recurrence

With recurrence you can schedule a crawl to run at specified intervals.

🕒

Current crawl: non recurring

The crawl is going to start ~Now

BACK

NEXT

crawl

Here you can set scheduled crawls, which will automatically extract new content at daily, weekly, or monthly intervals. You can read more about recurring crawls [here](#).

Session Cookie

✓

Type

✓

Scope

✓

Pages

4

Fine-tune

5

Summary

ORIGIN (SOURCE) SNAPSHOTS

COLLECT RESOURCES

LIMIT TO/EXCLUDE

RECURRENCE

SESSION COOKIE

Session Cookie

Login areas on a site keep track of user sessions via cookies. Extract these cookies and pass them to the Proxy crawler to access these website areas.

e.g PHPSESSID=xegpzebdddfejxzuvqzpq

BACK

NEXT

resources

You may extract the cookies from a user session and pass them to a crawl. This feature is most useful when you need to deal with login walls and content that is only revealed within the context of a user session.

Note that we cannot guarantee that the cookie extraction method works with all proprietary logins, as there are several solutions which can use additional security measures such as IP checking, or checking the User-Agent header. As all the requests from the proxy and the crawler come from the Google Cloud, it is necessary for the target site to not block them. The success of a crawl via the use of session cookies is highly dependent on the implementation details of a site.

You can read more about extracting cookies that you can pass to crawls [here](#)

STEP 4 - Summary

The screenshot displays the 'STEP 4 - Summary' interface, which is part of a five-step process. The steps are: Type, Scope, Pages, Fine-tune, and Summary (the current step, indicated by a blue circle with a '5').

The interface is divided into two main sections:

- Crawl Settings Summary:**
 - Type:** Wordcount (Discovery)
 - Store content:** No
 - Scope:** Re-visit current pagelist & Find new pages
 - Add a memo to the Crawl:** A text area with a 'WRITE' button and a 'PREVIEW' button. Below the text area is a 'Page Limit' section with a value of '100' and a note: 'URLs in the page list are visited first'.
- Cost Projection*:** A table showing estimated costs for different crawl actions.

Cost name		€	Cost
Word count for	100 pages	*0.001	€0.10
Content extraction for	N/A unique words	*0.002	N/A
Translation for	N/A target languages	*0.01*0	N/A
			€0.10

*This estimation is based on an empty database.

At the bottom right of the interface, there are two buttons: 'BACK' and 'START CRAWL'.

As the last step you can check the crawl's settings, and also a *Cost projection* will be displayed for you. Please note that this estimation is based on an empty database.

The statistics will be emailed to you and made available in the crawl list where you can use the [Statistics card](#) or the [Crawl analyzer tool](#).

3.3.3 Crawl List - history

The screenshot displays the Translation Proxy dashboard with a focus on the 'Crawl List - history' section. The interface is divided into several panels:

- Wordcount (Discovery) Done:** Shows a completed crawl with the following details:
 - Started: July 01, 2019, 11:36:33
 - Finished: July 01, 2019, 12:01:15
 - Took: 0:24:45
 - Visited: 489 pages
 - No more pages left.
 - No memo was provided.
- Statistics of pages processed this crawl:** A table showing the breakdown of processed pages:

	Match	Segments	Source words	Latin characters	Latin characters with spaces	Asian glyphs
<input type="checkbox"/>	102%	38,174	121,476	603,137	688,825	0
<input checked="" type="checkbox"/>	101%	92	1,634	8,242	9,787	0
<input checked="" type="checkbox"/>	100%	626	4,524	23,426	27,330	0
<input checked="" type="checkbox"/>	99%	1,864	12,579	66,148	78,173	0
<input checked="" type="checkbox"/>	98%	53	678	3,465	4,113	0
<input checked="" type="checkbox"/>	Unique	4,974	70,894	366,283	433,106	2
Selected total:	7,609	90,309	467,564	552,509	2	
- Crawl Settings Summary:** A summary of the crawl configuration, including:
 - Page limit: Unlimited
 - CrawlJob log links: https://app.easyling.com/ui/recrawl/log/6i7qzgxal6yyl5/6i7qzgxal6yyl5_15
 - Collect selected resource types: ☐ new HTML pages, ☐ JS, CSS resources, ☐ images, binary resources
 - Skip content type check: ☒
- Cost Projection*:** A table showing the estimated costs for different services:

Cost name	€	Cost
Word count for 489 pages	*0.001	€0.49
Content extraction for 90309 unique words	*0.002	€180.62
Translation for 2 target languages	*0.01*90309	€1,806.18
Total		€1,987.29

*This estimation is based on an empty database.
- Previous crawls:** A list of previous crawls on the right side of the dashboard:
 - Wordcount (Discovery) Done:** Started: July 01, 2019, 11:36:33; Finished: July 01, 2019, 12:01:15; Took: 0:24:45; Visited: 489 pages; No more pages left.
 - Wordcount (Discovery) Done:** Started: June 25, 2019, 14:40:21; Finished: June 25, 2019, 14:43:27; Took: 0:03:09; Visited: 2 pages; No more pages left.
 - Wordcount (Discovery) Cancelled:** Started: June 25, 2019, 14:38:25; Visited: 0 pages; Cancelled - displayed statistics are partial results of the crawl.
 - Content extraction (Scan) Done:** Started: June 13, 2019, 16:45:25; Finished: June 13, 2019, 16:45:33; Took: 0:00:11; Visited: 0 pages; No more pages left.

list

In this section, you can see information about your crawls. On the right side, you can see all of them sorted by their status (active, queued & completed). In the middle, you'll see all the details that are available for the selected crawl.

The first section contains some basic information: start and finish times, the number of pages visited, the reason of termination and any memos that were added.

The **Statistics** section gives you an overview on the content the crawl found. This card has a separate section dedicated to it [here](#)

The **Crawl settings summary** is similar to step 5 of the Crawl wizard. This can help you to identify the crawl in case you forgot to add a memo. It can also be particularly useful when diagnosing crawl-related issues (like a crawl that found no new content). In the CrawlJob log links section, you find a button taking you to the [Crawl analyzer](#).

The **Cost projection** estimates the cost of using the proxy for this site. Note that this assumes an empty database meaning that this is an estimation of the *total costs* of the project not just the new content.

Request summary gives you information on the requests the proxy sent. The number of requests aren't necessarily the same as the number of links added to the project because it's necessary to send a request to determine the type of a link which may not be added in the end.

3.3.4 Recurring crawls

Recurring crawls instruct the crawler to periodically go through the site looking for new/changed content. They are most useful once the initial translation is complete and the project moves into the *maintenance* phase of its life cycle. You can see your scheduled crawls in this section

You can read more about scheduled crawls and the automation possibilities they allow in conjunction with Auto pre-translate [here](#).

3.3.5 Statistics

Statistics of pages processed this crawl						
<input type="checkbox"/>	Match	Segments	Source words	Latin characters	Latin characters with spaces	Asian glyphs
<input type="checkbox"/>	102%	38,174	121,476	603,137	688,825	0
<input checked="" type="checkbox"/>	101%	92	1,634	8,242	9,787	0
<input checked="" type="checkbox"/>	100%	626	4,524	23,426	27,330	0
<input checked="" type="checkbox"/>	99%	1,864	12,579	66,148	78,173	0
<input checked="" type="checkbox"/>	98%	53	678	3,465	4,113	0
<input checked="" type="checkbox"/>	Unique	4,974	70,894	366,283	433,106	2
	Selected total:	7,609	90,309	467,564	552,509	2

Crawl

statistics

To understand the way statistics work in the proxy, we have to start with the most important fact: in our application, the largest unit of measurement is a block, which is usually represented by a `<p>` or a `<div>`. Blocks break down to segments, segments to words, words to letters. Since the translation proxy deals exclusively with content in webpages, HTML tags also play an important part in weighing the repetitions.

It is also important to note that the statistics in the translation proxy are different degrees of repetitions. During a crawl, the website's content is "repetitioned" against itself, simulating a translation process not unlike the Homogeneity feature in MemoQ.

Repetitions

Below is a breakdown of the various repetition percentages in the Statistics.

102% - Strong contextual repetitions: These are block repetitions. Every segment in the block is a 101% repetition, and all the tags are identical. We do not charge for these repetitions and they are propagated automatically within the project.

101% - Contextual repetitions: These repetitions are comparable to the 101% repetitions in MemoQ, or Context Matches in SDL Trados Studio. Both tags in the segment, and contexts (segments immediately before and after) repeat.

100% - Regular repetitions: This one is straightforward, and comparable to the Repetitions count in MemoQ or Trados. The segment is repeated exactly, including all tags.

99% - Strong fuzzy repetitions: In this case, a repetition is found after few transformations on the segment before comparing: tags from the ends are stripped out, words lowercased, numbers ignored.

98% - Weak fuzzy repetitions: Here, all tags are stripped out, not just the ones in the end; words lowercased, numbers ignored.

102% repetitions

102% repetitions warrant special mention. The proxy deals with *HTML block entries*, and has a very “overarching” view of them, since it strives to ensure that the same entry is never translated twice, regardless of which page it shows up on.

Consider a **navigation bar** of a website. During a Discovery, the proxy will come across the navbar *for the first time* on the landing page. It will aggregate the word count of the block elements in the navigation bar as **unique** and moves on to analyze the next page.

Of course, navigation bars are such that they are shown on all pages of a website, so when the proxy sees the same navigation bar for the *second time* somewhere else, it doesn’t count it as unique: instead, it adds the associated word count as a **102% repetition**.

So, the navigation bar is liable to be counted as many times as there are pages: so don’t be alarmed if you see large numbers in the 102% repetition row – any sort of repeated content is mercilessly added there. Just keep in mind that this is work that the proxy is saving you.

Cost

102% repetitions should not be counted when creating *cost projections* based on a word count result. So, remember the formula: for any given word count result, `total minus 102% repetitions` is the maximum amount that you need to extract or translate.

Examples

You will find an illustration of the various repetitions in the table below. Hover your mouse over any of the matches to highlight the differences.

3.3.6 Crawl Analyzer

After using the *Crawl Wizard tool*, and running a *Wordcount (Discovery)* or a *Content extraction (Scan)*, you can check its results under the *List* section by either downloading the `.log` file, or using the *Crawl log visualization filter*.

When you click on the filter button, a new window will pop up, where you can narrow down the results of the log.

Crawl Settings Summary ▼

Page limit	100
CrawlJob log links	https://de-de-gereblye-app-proxytranslation.com/ui/recrawl/log/155715079 2959_35637-00000.log
Collect selected resource types	<input checked="" type="checkbox"/> new HTML pages <input checked="" type="checkbox"/> JS, CSS resources <input type="checkbox"/> images, binary resources
Skip content type check	<input checked="" type="checkbox"/>

Open

crawl log filter

Crawl status

Crawl log visualization filter


Crawl status: processed, failed, skipped, other ▼

Response type: html, js, json, xml, css, image, other ▼

Response code: 200-299, 300-399, 400-499, 500-599 ▼

(optional) Regexp to match log lines to be displayed

CANCEL

SHOW LOG 

Crawl

log filter

Processed: The page was successfully added to the pages list, the crawler was able to process it.

Skipped: In this case, the Proxy could not process the page, because of its `content-type`, or because of the configuration of the collectible resources in *Crawl wizard*. There are four main reasons behind this status:

- *No content type, page collection is disabled:* The crawler hadn't received a content-type before, and also the collection of new HTML pages is disabled.
- *HTML page, page collection is disabled:* The content-type is `text/html`, but the collection of new HTML pages is disabled.
- *Not HTML page, resource collection is disabled:* The URL points to a resource (e.g. the content-type is `text/javascript` or `text/css`), but the collection of resources is disabled.
- *Content with type < content-type > is not processed during this crawl:* The content-type header designates an unprocessable type, for example if the header is proprietary (such as `"application/example-script"`), or it is an unsupported content-type, for example `text/plain`.

Failed: The proxy was not able to process the content, you can see below the list of the reasons behind this status:

- *Path is externalized:* The page is externalized
- *Request sending failed:* The crawler wasn't able to send a GET request
- *Content is not an HTML page:* The content is not HTML, the crawler wasn't able process it
- *Not processable:* The crawler wasn't able process the given content-type
- *Too large, size: < size of content >, limit: 1048576:* Files above 1.5 MB are not handled
- *Parsing error:* Invalid script, or HTML format for example
- *Proxy request failed, Response not processable, Response processing was aborted, Skipped because processing failed:* Error during processing. Sometimes it is not quite obvious at first glance why the crawler failed, but in this case we can check our logs for you

Response type

You can also filter the log results by content-types, for example, you can select different resource types, like CSS, JS, images etc.

Response code

With this filter you can narrow down the log list by [HTTP status](#) codes:

1xx (Informational): The request was received, continuing process

2xx (Successful): The request was successfully received, understood, and accepted

3xx (Redirection): Further action needs to be taken in order to complete the request

4xx (Client Error): The request contains bad syntax or cannot be fulfilled

5xx (Server Error): The server failed to fulfill an apparently valid request

Regexp

Inserting here links or other texts, will be treated as strings, but you can specify `java/util/Pattern` regular expressions as well. You can test your regex [here](#).

3.4 Content

3.4.1 Work packages

Work package generation is the recommended method of dealing with exports/new content. Along with the general import and export features, work package generation becomes available after the first round of Scanning (Content Extraction) has finished and translatable content (segments) has been placed in the database.

Initially, the Work package list will be empty, you can create one using the Work package wizard.

Work package wizard

Work Package Generation Wizard

GENERATION SETTINGS SOURCE FILTERS TARGET FILTERS

Work Package Name

Source language English (United Kingdom) [en-GB]

Work package languages

Options

☒ Split package by every words.

(Enter 1 to start new work package for every page. If unselected, work packages will be as big as possible, the soft limit is 20,000 the hard limit is 30,000.)

☐ Create separate work package for hidden elements

☒ Automatically export as XLIFF

☒ Export entries without translation only

☐ Send XLIFF files to

[Configure XTM](#) [Configure XTRF](#) [Configure Dropbox](#)

[GENERATE](#)

package wizard

Work

This wizard is designed to simplify the creation process of Work packages. It is split into 3 tabs.

Generation settings

This is where the most important settings are. In most cases, you won't need to leave this tab. It has lots of options. Let's take a look at them one by one.

Work package name: You can name your work package any way you like. As a suggestion, it is recommended that you give it a name that accurately reflects its contents, so as to make project management using work packages easier. So, for example, if you generate a work package for Arabic as a target language on April 10th, 2020 for all new content, you would name it something like *ar-SA-untranslatedContent-2020-04-10* and so on.

Source language: The source language for the project is displayed in this field (it can be changed on the old Dashboard main screen).

Work package languages: You can add multiple target languages for the Generation Process, but remember that a separate Work Package will be created for each target language.

Options Miscellaneous options you can use to fine-tune the properties of your Work Package.

- Split your package at preset words to create easily manageable chunks to assign to translators. Disable it to put everything in one big file (Note, however, that Work packages will be automatically split at 30.000 words)
- You can choose to create separate work packages for hidden elements, and set the Work package generation to automatically take care of exporting for you. As with the general export function, you may elect to export only those entries that have no translation yet.
- If you configured XTM, XTRF or Dropbox to integrate with the proxy, you may instruct the software to automatically send the exported XLIFF files to either one of those services. Use the dropdown menu to make your choice (only available if the checkbox is checked).

Source filters

Source contains: Add a Java regular expression. Only segments where the source matches will be included.

HTML tag: Enter a comma separated list of tags that you wish to include. Only content found in those tags will be included.

HTML class: Add specific classes that must or must not be on a specific element for it to be included in the Work package.

Timeline (always selected): You can specify which Content Extraction cycle you'd like to see in the Work Package that is to be generated. By default, it's from `Project created` until `Now`. Click on either of the dropdown menus to change the start or end dates.

Filter pages (always selected): You have 3 options here:

- *Current selection* refers to your selection in the Pages list. If you select this, the Work package will follow the inclusion/exclusion rules you set there.
- *Selected pages:* You can specify a list of pages that you wish to add.
- *Selected folders:* Add include and exclude rules for this Work package. This is completely separate from those in the Pages list.

Target filters

Filter by translation source: You can specify whether you want segments with machine or human translations. This can be very useful if you had to machine translate some part of the website to meet deadlines but want to go back and have them reviewed by humans.

By publish state: This setting is only applicable if manual publishing is turned on.

Segment assigned to: Any workflow state in the Workbench. The available options are influenced by the workflow of the project (set on the Miscellaneous options card of the Project overview).

Has unresolved comment refers to the comments in the Workbench.

By comment (or target) contents: A Java regular expression to filter targets. This can be used if you need to fix the spelling of `donut` to `doughnut` everywhere in a British English text.

By date range refers to the date the segment was last edited.

Last modified by: Select a user whose translations you wish to include. Can be useful when reviewing translations.

Work package list

example

Source language: English (United Kingdom) [en-GB]
Creation time: September 09, 2020. 16:00:05

Progress report

Work package languages

French (France) [fr-FR] 30 words
 German (Germany) [de-DE] 30 words

Generation parameters

Split package by: 2000 words
Create separate work package for hidden elements ☐
Automatically export as XLIFF ☒
Export entries without translation only ☒
Timeline
From September 08, 2020. 14:35:08 Crawl (Content extraction) (0)
To September 09, 2020. 16:00:04 Work package generation (1)

Work packages

example

September 09, 2020. 16:00:05

30 words
 30 words

NO MORE WORK PACKAGE

package list

You will receive e-mail notification about your generated work packages and they will appear in the list. In the middle, you can see the settings used to create the selected package. On the right, in the list, every element has a context menu with important actions:

Generate progress report: Generate a detailed report on the progress made on this package. This will be emailed to you. The same information is available on the Progress report card.

Refresh progress report: Refreshes the Progress report card

Export XLIFF: Open the export as XLIFF dialog and export this package as XLIFF. It will be emailed to you and will become available in the Previous exports menu.

Pre-translate: Open the Pre-translate dialog with settings preset to translate this Work package.

Create new Work package with the same settings: Opens the Work package wizard preset to match the settings of this package. You can change its settings then click generate.

3.4.2 Import translation

Import

dialog

When your XLIFF files have made a round in your favourite CAT tool, it is time to reimport them, which is a straightforward process - open the Import menu. Here you can click on “Choose file” to browse for your translated XLIFF files. Click on “Import translation” and wait for the import process to finish.

If you have Dropbox linked to your account, you can also import from there by selecting the Dropbox import tab. You can read more about the Dropbox integration [here](#)

You may optionally set the translation workflow status of the imported segments. The options available depend on the workflow set on the Miscellaneous options card of the Project overview.

You can choose to apply the *Approved* attribute that XLIFF supports to mark the segments as confirmed. Most CAT tools allow you edit this attribute on a per-segment basis. Check the box to update the translation progress.

NOTE! Importing is not instantaneous, it may take some time to finish depending on the size of your XLIFF! You will receive an email once it’s finished.

3.4.3 Export translation

By clicking on this link, the general Export dialog will open on the old Dashboard. Most of the options will be familiar from the Work package generation dialog. Choose a file format of your preference (with XLIFF being the *de facto* industry standard, it is recommended that you go with this option, as it is more structured than CSV).

Languages to export You can choose to export more than one target languages in one go, or choose to export them all.

Export range Click on “change” and use the timeline window to specify the time period, and the export function will be limited to that timeframe. Select any two timestamped buttons to define the start and end of the time frame.

Export You can choose between exporting only those entries that lack a translation, or include all entries.

Unique segments only Check this option to ensure that identical segments are not duplicated in your exported XLIFF file.

Skip excluded and pending segments from export Export only those segments that have been approved for translation on the Workbench.

Trim export to contain as few tags and whitespaces as possible Use this option to clean the source segments of tags for the purposes of translation.

Split enter the number of segments that you'd like to have in a single file, and the export will split your segments into separate files accordingly.

Copy source to target where empty Some CAT-tools require a target entry on import. Activating this option will copy the source content to the target entry during the export, so the tool can be used to edit it.

Send XLIFF files Choose among your XTM, XRTF or Dropbox accounts to send your exported files to. Each of these accounts has a link at the bottom of the dialog that will take you to the given account's setting screen.

Click on Export to execute. Wait for the process to finish - you will receive an e-mail when your files are ready.

3.4.4 Pre-translate

Pre-translate

French (France) [fr-FR]



☐ Overwrite existing translations

☒ Everything

☐ Segments from , until will be included.

.....

☐ http://example.com/path/to/file

☐ Work package code

TRANSLATION MEMORY

MACHINE TRANSLATION

Pre-translation threshold

99 %

Confirm threshold

☐ 99 %



Only pre-translate if the country code matches as well as the language
(e.g. Swiss German won't be translated using the standard German TM)

By using the Pre-translate menu, you can initiate a process which supplies untranslated segments with a preliminary translation in order to mitigate bleedthrough on the proxied pages. You can use pre-translate with a Translation Memory as well as any of the supported Machine Translation APIs.

When you pre-translate using a Translation Memory, the project's default memory will be used. By default, this is populated with the segments that you previously added. You can change this using the [Translation Memory page](#), where you can add more content to your memory if you need to.

In order to use Machine Translation, you need at least one API configured for your project. You can access configuration for available Machine Translators by clicking *configure* next to the desired Translator, or by selecting them on in the menu bar.

Select the appropriate radio button to pre-translate everything, segments added between a specific time period, a specific page or a work package.

3.4.5 Previous exports

<input type="checkbox"/>	Creation time	File format	File name	Target language	Word count	Status	Actions
<input checked="" type="checkbox"/>	September 10, 2020. 14:54:56	XLF	example__example.com_de-DE_1599742495873.xlf.zip		30		
<input checked="" type="checkbox"/>	September 10, 2020. 14:54:56	XLF	example__example.com_fr-FR_1599742495725.xlf.zip		26		
<input type="checkbox"/>	September 09, 2020. 16:00:07	XLF	example__example.com_de-DE_1599660007068.xlf.zip		30		
<input type="checkbox"/>	September 09, 2020. 16:00:07	XLF	example__example.com_fr-FR_1599660006775.xlf.zip		26		

NO MORE EXPORT DATA
DOWNLOAD SELECTED FILES
⚠ Please enable popup windows for this site. This is necessary in order to download multiple exports.

[Previous](#)

exports dialog

Although the notification e-mail does contain the link to your exported file, there is no reason to track it from your mailbox. This dialog will list and organize all previous exports for your viewing with the appropriate creation time, target language and word count displayed alongside Export list entries.

If you need multiple files at once (e.g. the XLIFF files for all of the languages), you can select them using the checkboxes and click Download selected files. Note that most modern browsers block popup windows by default. For this feature to work, you must enable them for the Dashboard 2.0.

Using the Actions column, you can send XLIFF files to XTM, XTRF or Dropbox.

JavaScript exports get special treatment. They are the scripts used for Client-side translation so they have different actions. As these exports aren't meant to be worked on by translators, you can't send them to external systems. The only option you have is to Publish it. After clicking it, you can add the line of JavaScript as described on the [Client-side translation page](#).

3.4.6 Delete entries

Delete entries

- ☒ Delete all the source entries and their translation(s) created after Project created ▾ , until Now ▾ matching regexp

```
^(s*CSEException.*)$
ContentFragmentRenderingError (d+)
TemplateParseError (d+)
UnknownError (d+)
(. *Could not send socket message 'received' for user 'd+' to the user message event router.*)
(. *Could not send socket message 'read' for user 'd+' to the user message event router.*)
PluginInitializationError (d+)
([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})
([A-Z0-9]{7}-[A-Z0-9]{7})
```

Using newlines for multiple expressions has the same effect as using the Vertical Bar or Pipe Symbol (|). Use the external [Regex101 tester](#) or [RegexPlanet](#) to test & optimize patterns on small input snippet.

- ☐ Cleanup: delete all the source entries with no translation on any target language.

N.B.: entries are deleted if and only if they have no associated translations on any of the target languages and have never had any. Viewing an entry in the Workbench also constitutes a translation, sparing that entry from deletion.

DELETE

Delete

entries dialog

Use this section to delete segments from the proxy. You have 2 main options:

- **Delete based on source content and timeline:** Select a time period (the default is Project created to Now) and enter a [Java regular expression](#) to search in the source of the segments.
- **Delete all the source entries with no translation on any target language** allows you to clean the project from unwanted entries that have no translations yet.

3.5 Snapshot

3.5.1 Origin Snapshots

Origin Snapshots are your first line of defence against bleedthrough: it is a way of storing and reusing content that was received and picked up from the remote server (in other words, the source site). The goal of Origin Snapshots is to create a controlled snapshot of the source content that you can safely use to translate and serve content over the proxy.

We understand that Snapshots may sound daunting at first. [Thinking of them as a photo album](#) may help.

You might want to set up an Origin Snapshot for publishing content and another to be used during translation. The content of the Snapshot being published should be highly translated, and new content should be collected in a different Snapshot to be used in Preview mode and during translation.

Benefits of Origin Snapshots

- If you have at least one Origin Snapshot built and enabled, you have essentially decoupled new content from what you are serving over the proxy (you are in control of untranslated content).

Note: Origin Snapshots can only be built by Scans. They will never pick up new content unless you expressly run a Content Extraction to update them.

See Case Study #1 for an in-depth coverage of this scenario.

- Multiple Origin Snapshots let you decouple published content from untranslated material – you are free to update any Origin Snapshots you are not currently using for published content over the proxy. This way, you are always serving 100% translated content (using the Snapshot that is highly translated), and you can schedule for translation updates (on all the others you are free to update anytime with a Scan).

See Case Study #2 for an in-depth coverage of this scenario.

When to build?

The best time to build an Origin Snapshot for the first time is during the first Content Extraction (Scan). By this time, you should have used Discovery a couple of times to gain a general understanding of the site structure and word count, and you have all necessary exclusion rules set up.

When you are confident in your Content Settings, set up an Origin Snapshot before you run a Content Scan (see below for setup directions) on the webpage and build it. This way, you will have an Origin Snapshot that contains all content you're basing your quote on.

Setting up an Origin Snapshot

You can create and enable a new Origin Snapshot by checking **Enabled** in the Snapshot menu in the Dashboard 2.0. Note that at this point, the Snapshot is enabled but **unbuilt**.

WARNING!: You have to build the Origin Snapshot before it becomes functional!

Building the Origin Snapshot

After enabling the Origin Snapshot, you need to Scan for content to build it. In step 4 (Fine-tune) of the Crawl Wizard some options become available:

✓ Type ✓ Scope ✓ Pages 4 Fine-tune 5 Summary

ORIGIN (SOURCE) SNAPSHOTS COLLECT RESOURCES LIMIT TO/EXCLUDE RECURRENCE SESSION COOKIE

Origin Snapshots

Snapshots let you store and later serve the source site's contents in a **fixed state**: as they're encountered during this crawl. The snapshot contents **will not change even if the original site updates** until you update them in a future crawl.

#2 Snapshot from 15/05/2018 ▼

- ☐ Reuse existing pages and store new pages
- ☒ Reuse existing pages and don't store new pages
- ☐ Don't reuse existing pages, update/store all encountered pages

BACK NEXT Build

Snapshot Screen

Select the Snapshot you want to build and choose one of the 3 options. By choosing to reuse existing pages, you instruct the Crawler to skip the ones that are already in the Snapshot you chose and thus reduce the building cost. The choice whether to add new pages gives you the possibility to simply ignore the new pages that were added to the source site. Selecting not to add them (the second option) allows you to build a snapshot that has the pages updated but no new ones added. This option is useful if you made changes to the JSON Paths and as a consequence need to rebuild the Snapshot. As a result of building the Snapshot, all content that was set to be picked up by the Content Scan is added to the current Snapshot.

Custom settings

If you have multiple Origin Snapshots that you've built throughout multiple Content Scans, you can visit the Snapshot menu again and configure which Origin Snapshot should be used for which proxy mode, as shown in the screenshot below:

ORIGIN (SOURCE)

TRANSLATION (TARGET)

Configuration

☐ Disable snapshots
 ☐ Enable for all proxy modes
 ☒ Custom settings

<i>Published site</i> ⓘ	#1 Serving snapshot	▼
<i>Live test mode</i> ⓘ	#1 Serving snapshot	▼
<i>Preview mode (-p)</i> ⓘ	#2 Snapshot from 13/05/2018	▼
<i>Highlight/popout view</i> ⓘ	#2 Snapshot from 13/05/2018	▼
<i>All other modes</i> ⓘ	Disabled	▼

Origin (Source) snapshots

	Last update by crawl	Go to history
#1 Serving snapshot	February 21, 2019. 15:25:13	↗
#2 Snapshot from 13/05/2018	May 13, 2019. 14:45:44	↗

Custom

Origin Snapshot Settings

Choose a highly translated Origin Snapshot for publishing, so that you can be sure that visitors to the proxied site will not experience bleedthrough.

Meanwhile, you can use other, more up-to-date Snapshots for the preview/live test etc. proxy modes to work on the translations of new content.

Your Origin Snapshot setup is complete.

3.5.2 Translation Snapshots

The Translation Snapshot option is used to achieve great boosts in page serving speed by enabling the proxy to skip processing the page instantly if the remote server's response matches the response that was used to generate the Snapshot during building. By not having to insert the translations separately, page serving can be accelerated by several milliseconds, which can add up to a noticeable speed-up in the case of larger pages with lots of translated content.

The Translation Snapshot is built or overwritten every time a page is loaded through the proxy, with a few notable exceptions. The snapshot is not overwritten if the content served matches the content received (i.e. no processing was done on it), nor are entities larger than the hard-coded maximum entity size (960kb) saved. Furthermore, if a site changes its contents too fast (there are too many cache misses, i.e. the cached content differs from the actual), the Proxy will stop caching the given entity to prevent overusing the database. Should this happen, snapshots must be cleared manually to restore normal operation and reset the cache miss limit.

What is it for?

The purpose of the Translation Snapshot is to reduce the number of requests to the **source site** and decrease response times on the proxy by skipping the content replacement process (which is basically the entire document pipeline) if the content on the **original site** or – if enabled – in the Origin Snapshot remains *unchanged*.

Enabling Translation Snapshots

Go into the **Snapshot** menu in the Dashboard 2.0 and enable Translation Snapshot. You can also add a maximum of **10** custom-named snapshots. It's generally recommended to set up at most **5** Origin and **5** Translation Snapshots.

Building the Translation Snapshot

After you enable it, the Translation Snapshot is filled exclusively by Page visits on the pretty (published) domain. Previews/visits to the temporary domain will have absolutely no bearing on the contents of the Translation Snapshot.

While separate from the Origin Snapshots, the contents of the Translation Snapshot will nevertheless be determined by your Origin Snapshot settings. If you have no Origin Snapshot, then the Translation Snapshot will be filled by whatever is returned from the source site for a request (a page visit). If you have an Origin Snapshot, then the Translation Snapshot will contain whatever the currently enabled Origin Snapshot is making available to the visitor on the proxy.

The 'Keep Cache Strategy'

The Keep cache is actually a serving mode of the Translation Snapshot, used mainly to prevent the source language bleedthrough effect, or Bleeding Effect in short. It operates by checking the cached content's translation ratio against the remote server's response. If the cached content is found to have a more complete translation (i.e. its translation ratio is higher), the remote response is discarded and the cached content is served instead. This does not mean an increase in page load speed, but by preventing yet-untranslated elements from appearing in the served page, the Bleeding Effect is eliminated entirely. As new translations are entered into the database, either manually or via XLIFF imports, the difference between the cached response and the actual response decreases, and the newly-translated elements are displayed automatically. Additionally, this check is run every time the source content is found.

3.5.3 Settings of individual Snapshots

These options can be set for each Snapshot individually by clicking on their respective rows in the table and choosing the appropriate tab of the side menu that appears.

Settings

Here, you can change the name of the chosen Snapshot. You can also restrict the scope of the Snapshot using the Content Matching text boxes. These accept wildcard-compatible path specifications that set whether or not the snapshot can be used on the page(s) in question. The checkboxes allow you to set what component(s) of the site to serve from the snapshot (e.g. you can set up to cache HTML and JS/CSS but not the Images if your client wants to avoid bleedthrough on normal text but doesn't mind it in case of images such as logos). You can also see when the Snapshot was last updated (or built) using a Scan.

Export

Using this menu, you can generate a zip file containing all the pages that are in the selected snapshot. Note that this service isn't free. It will create a cost equal to that of one proxy request per page in the page list (not per page in the snapshot). Once the export is created, you receive a link to download it via e-mail.

Copy

This is where you can copy the contents of your selected Snapshot into an other one. You must first select the target Snapshot from the drop-down menu then click Copy. You can then review your settings and if everything is in order click Copy again. Similarly to Export, it will entail the cost of one proxy request per project page. The contents will be transferred in the background and you will receive an e-mail notification. Choosing the "Clear Target before copying" option removes all content from the Snapshot you are copying to, creating an exact copy of the original Snapshot without any pages from an older build accidentally remaining in the target.

Clear

This part does exactly what the title suggests. Choosing to clear a Snapshot removes all content from it. Note that until you rebuild the Snapshot, the content itself isn't deleted completely and can be recovered. If you accidentally clear a Snapshot, it's recommended to immediately notify everyone else working on the project not to overwrite it and to contact our support team.

3.5.4 Case Studies

Case Study 1: Freeze Site Before Translation begins

In this example, the client has declined to use a staging server, as well as declining to halt content updates for the duration of the initial translation. They do, however, insist that bleedthrough must not occur at any point in the translation process. This results in a continuously changing source that makes it nearly impossible to achieve 100% translation.

In this case, the solution is to enable the Origin Snapshot, and populate it with a Scan before the first round of translation commences. This creates what is effectively a static snapshot of the site, which remains the same regardless of the updates the client makes in the meantime, providing a stable environment for the translation and review processes.

By using an Origin Snapshot to serve the translated sites, they are decoupled from the original, and content updates there will not be reflected in the translations.

However, source content will accumulate in the meantime, and once the snapshot is purged, bleedthrough will occur! To counter this, the second scenario can be enacted.

Case Study 2: Decoupling Content Update from Ingestion and Publishing

For this example, the client has declined to use a staging server to allow you to ingest the content ahead of publishing for translation. They did, however, agree to notify you once new content is published, and have acknowledged that this will cause the translated sites to lag behind the original until the translations (and possibly reviews) are completed. Alternatively, coupling into the previous scenario, the initial translations are in place, but the source site has moved ahead in the meantime, with the translated site being served by the initial Origin Snapshot.

Once the initial translations are completed, the snapshots are set up according to the image below:

The screenshot displays two side-by-side configuration panels for the Translation Proxy. The left panel is titled 'ORIGIN (SOURCE)' and the right panel is titled 'TRANSLATION (TARGET)'. Both panels have a 'Configuration' section with radio buttons for 'Disable snapshots', 'Enable for all proxy modes', and 'Custom settings'. The 'Custom settings' option is selected in both. Below this, there are dropdown menus for 'Published site', 'Live test mode', 'Preview mode (-p)', 'Highlight/popout view', and 'All other modes'. The 'Published site' dropdown is set to '#1 Snapshot from 21/02/2019'. The 'Live test mode' dropdown is set to '#2 Snapshot from 13/05/2018'. The 'Preview mode (-p)' dropdown is set to '#2 Snapshot from 13/05/2018'. The 'Highlight/popout view' dropdown is set to '#2 Snapshot from 13/05/2018'. The 'All other modes' dropdown is set to '#2 Snapshot from 13/05/2018'. Below the configuration section, there is a table titled 'Origin (Source) snapshots' with columns for 'Last update by crawl' and 'Go to history'. The table contains two rows: '#1 Snapshot from 21/02/2019' with a date of 'February 21, 2019, 15:25:13' and a link to history, and '#2 Snapshot from 13/05/2018' with a date of 'May 13, 2019, 14:45:44' and a link to history. Below the table, there is a text input field for 'New Origin (Source) snapshot name' and an 'ADD NEW' button. The right panel has a 'Translation (Target) snapshots' section with a text input field for '#0 default cache' and a text input field for 'New Translation (Target) snapshot name' with an 'ADD NEW' button.

Sample

Snapshot configuration

By driving the published site from a separate snapshot entity, you gain the ability to decouple the content ingestion cycle from the translation and update cycles.

From this point on, the published site will not reflect any updates until the assigned entity is refreshed. By specifying the update snapshot in the Crawl Wizard, you write the server responses into the newly created snapshot, which will drive the Preview and Highlight modes, allowing you to conduct the translations and ICR using the new content. Once the client signs off on the translations, you simply switch the Origin Snapshot entities being used (and purge the Translation Snapshot, if in use).

Once translations and reviews are completed, simply re-assign the snapshot entities to promote the new content to the translated “Production”

At this point, the previously-live snapshot is freed up for purging and re-building in the next update cycle. Thus, at the next crawl, the default would be selected to receive the updates, leaving the other entity untouched.

To facilitate this process, you can set up a Round-Robin Crawl. These are Crawls that are done periodically, writing content into a different Snapshot every time (i.e. it writes first into #1 then next month into #2 and into #3 on the third. On the fourth month, it starts the circle again by writing into #1). To set it up, you first have to ensure that you have at least two Origin Snapshots (they can be empty at this point). Then, you should set up a Scan using the Crawl Wizard. On step 4, first navigate to the Recurrence tab, select the repeat interval and tick the “Use different Origin Snapshots for every crawl” option. Then, you must select the Origin Snapshots tab and select which ones to use. Once all of this is in place, you can review your settings and start the crawl.

3.6 Settings

3.6.1 Advanced settings

The Advanced settings screen is the *nitty-gritty* of the technical side of the proxy, the various features of which make complex project management and involved content extraction/management possible.

For now, this link takes you back to the old Dashboard but we'll update it shortly.

Some, such as **Freezes**, are general enough to warrant close attention from all users, others, such as the **Tweaks** checkbox list are required only in specific cases, so you can go through many, many projects without ever having to worry about them. In this section, we take a look at all options in the order that they appear on the Dashboard.

Pattern matching: there is often a great deal of text on a website that is not targeted for translation, because it is highly repetitive or numeric in nature, usernames, timestamps and prices fall into this category of content, to mention a few of many possible cases where Pattern Matching might come handy.

You can construct regular expressions and add them here, one regular expression per line.

If that regex has a capturing group, anything that is **matched by that capturing group** will be treated as **translation-invariable**.

It is important to note that if you add regexes here later into the project, your existing segments will remain in the Workbench. Pattern matching does **NOT** delete or exclude any segments.

Example: filtering usernames and dates

```
Posted by youTuTroll on Apr. 4, 13:22, 2010
Posted by joey0405 on Apr. 6, 03:29, 2010
Posted by cunning_linguist on Apr. 6, 03:45, 2010
Posted by bornToBe27 on Apr. 6, 10:01, 2010
Posted by OMEGA on Apr. 7, 11:59, 2010
```

Repetitive and inconsequential from a translation/word count viewpoint, the usernames and dates should be made translation-invariant in the content above. You can add a regex for it (such as the one below) in the textbox and the translation proxy will exclude whatever is matched within a capturing group.

```
Posted by ([\w]+) on ([\w]{3}\.\s[0-9]{1,2},\s[0-9]{2}:[0-9]{2},\s2[0-9]{3})
```

We recommend that you visit <http://www.regex101.com> and test your regular expressions with example snippets before setting them on a live project.

Freeze

The following are important options for when translation has already begun in earnest on a batch of content. At this point, it is a good idea to **freeze** the page list and the segments - that is, prevent *inadvertent* addition of new text or pages to the list you are currently working on.

WARNING: These settings can be overridden via explicit user action: adding Pages via the blue icon in the page list, or running Scans with the appropriate settings in place.

As you based quote on a specific Discovery/Scan (in other words, a specific state of the site at a given time), it wouldn't do to return a few days after winning the project only to find that the news section was expanded with two new items and you have 4500 new source words to deal with - indelibly added to the project, but not part of the original deal.

The following features are available:

Freeze pagelist: Prevent new pages from being added to the pagelist. If you turn this option on, you can use the various preview modes without having to worry about new content.

Handle unknown pages as externalized: Pages that are not already in the pagelist will be *externalized*, that is, not translated *at all*, the same as if the page was manually excluded. Be aware that on a live site, this may result in an SEO penalty (due to duplicate content being detected by the crawler)!

Freeze translation memory: No new translatable segments will be added to a project as long as this Freeze is turned on (automatically enables Page Freeze).

Group pages

Specify path rules to group pages into one entry, preventing new pages from being added, but not removing preexisting pages, and making the grouped pages share a single dictionary, necessitating translation of only one. The pages will be grouped according to the path rules specified in the textbox, one path per line.

This feature is not meant to and does not decrease the amount of pages a Discovery has to crawl, but it makes project maintenance easier, as it prevents your page lists from being overcrowded with repetitive URLs.

Additional Remote Request Headers

If the remote server requires certain headers to be present to serve legal responses, it will not be crawlable by default, as the crawler will not supply these. Entering the required headers here will result in them being appended to every request sent by the proxy, including the crawler requests, letting you crawl the site as required.

Ignore Classes

If a certain class of elements are not ought to be translated, they can be entered here. Elements with these classes will be treated as if they had the `<translate=no>` attribute, and will be treated as translation-invariant.

Ignore IDs

Similar to the “Ignored classes” option, this allows the treating of specific elements as translation-invariant, this time through HTML IDs.

Process Custom HTML Attributes

Some Content Management Systems may employ non-standard HTML attributes on various elements to style the page or otherwise affect aspects of their operation. If some of these attributes contain translatable text, you can enter them into the “As text” field to instruct the proxy to extract them. If they contain URLs that need to be mapped to the translated domain, you can use the “As link” field to instruct the proxy to map those non-standard link elements as well.

In case they contain HTML or JavaScript/JSON that contains translatable text, you can enter them into the “As HTML” or “As JavaScript” fields respectively. These will go through the regular HTML or JS parser to pick up content. Note that in case of JSON in an attribute, you still need to mark the JSON paths in the JSON/JS processing menu of the Dashboard 2.0.

You may want to further filter where a given attribute should be translated. You can do so with CSS-like advanced selectors. These are written after the name of the attribute (before the semicolon denoting the next one). Let’s see a few examples:

- If you want to translate the `content` attribute but only on meta tags, you can write `content meta`.
- If you need to further narrow down the scope, you can specify that `content` should only be translated on meta tags where a `name` attribute is present like so: `content meta[name]`.

- You can also specify the value of the `name` attribute. When you write `content meta[name=foo]`, the `content` attribute will only be translated on `meta` tags if the attribute `name` with the value `foo` is also present.
- It's also possible to chain these values. You can specify that you need an `id` to be present on top of the previous example's rules by adding it to the end: `content meta[name=foo][id]`

In some cases, you can't specify the exact value of an attribute that you wish to filter for. In this case, you can use the following operators:

- `^=`: looks for a string prefix
- `$=`: looks for a string suffix
- `#=`: searches for a regexp match

Both the attribute name and the match value are strings. If they contain any character other than letters, you must wrap them in quotes. You can also use raw strings with `r"` where the backslash character won't escape the ones after it. See the table below for reference:

Tweaks

In this section, you'll find checkboxes for settings that apply to very specific circumstances. For those special snowflakes and occasions. When in doubt, contact us!

- **Retaining original DOCTYPEs:** By default, the proxy generates an HTML5 standards-compliant file to send to the client. If, for some reason, this causes problems due to the site relying on HTML4 standards for operation, some of which may be deprecated by HTML5, enabling this option will cause the Proxy Application to retain the original DOCTYPE declaration of the source page.
- **Determine document type by GET instead of HEAD:** some servers may return different responses to the HEAD request we use to determine document type than the GET request used to download content. Enabling this option forces the proxy to use GET requests for all operations, getting the correct content type from the server (as far as server-side configurations will allow).
- **Detect JavaScript content type:** JavaScript may not be explicitly declared as such on the server, being sent to the client with misleading content types. This causes the proxy to mis-identify such streams and not offer operations reserved for JS files. Enabling this option will force a deep check on each file sent to the client, to determine if they are indeed JavaScript code, regardless of their declared content type.
- **Download images through the proxy:** this will instruct the Proxy Application to attempt to map all `` attributes to the proxied domain. This is especially useful if the images are served only after authentication.
- **Attempt to place tags according to punctuation when using TM-based pre-translation:** if using a TM-based pre-translation, the Proxy Application may encounter segments where it cannot replace the XLIFF tags automatically, due to overly large differences between the contexts (possibly because of a changed site). If this option is enabled, the translator will try to replace the XLIFF tags according to their positions relative to punctuation marks in the segment. If successful, the TM entry's confidence score will be increased by 0.1.
- **Translate excluded pages when viewing them in Preview mode (but still not in live serving mode):** It may be necessary sometimes to view excluded pages as if they were translated, in order to assess their layout, without actually making them available on the live site. Enabling this option allows just that, by propagating translations to the excluded pages if viewed in Preview mode, but keeping them untranslated in Live serving mode.
- **Translate javascript : attribute:** The Proxy is capable of extracting and translating code from the `onclick` attribute of elements. This feature may be used when a site uses this attribute to store translatable content inlined into the attribute and requires this content to be translated. Currently we only process the `onclick` event's contents, all other events in the `javascript` attribute will be ignored.

- **Detect and handle JSON-in-string responses, like** `{"\\\"ResponseCode\\\":\\\"BadRequest\\\"\"}`: string-escaped JSON-format responses can be handled by activating this tweak. If active, the proxy will first attempt to string-unescape the response before passing it to the JSON parser, in order to recreate its base form.
- **Make content in `<script type=`\"text/html\"`></script>` translatable as a whole (don't try to parse it as HTML)**: Script blocks may contain template data requiring translation, which is often signified by the text/html content type (instead of the more usual application major type). In such cases, HTML parsing can be undesirable, and can be bypassed by activating this option.
- **Send a canonical link http header pointing to the original page on externalized pages**: the proxy can insert a Link header into externalized pages, in order to avoid the SEO penalty associated with duplicate content. This header will point to the original address, and have `rel=Canonical` added, to designate the relationship.
- **Perform string concatenation in JavaScript before parsing**: the proxy will perform string concatenation and treat the result as a whole when translating/extracting it. This tweak is only useful if *no computable expression* is featured in a given concatenation.

Manual Publishing

Manual Publishing is advanced project control feature that gives project owners the ability to withhold the translations from being published on the live page (but not the preview, as it will always display the latest translation available!) until further notice.

The feature can be activated from the Advanced Settings page. Once activated, it will affect all translations going forward, but already-existing ones will not be “un-published”.

Once active, a new item will appear in the Bulk Actions menu of the Workbench: “Publish”. Running this action will cause all selected segments to be synchronized with their displayed translations, and once the action finishes, the markers in the status bar on the right of the entries will change to reflect this. If the action encounters an error, the server will attempt to rectify this by publishing the entire entry, after confirmation from the user. If the error is not recoverable, it will list the segments in error.

Default segment state

By default, the proxy will add new segments during a crawl as “Approved”, making them available for translation immediately. However, if the user/client so desires, this behavior can be changed to adding new segments in one of the two other states, “Pending” or “Excluded”. If the default setting is changed, the project owner, backup owners, or users with the Customer Role can alter segment states.

“**Pending**” segments are those that are awaiting a decision on translation. They will not be included in exports unless the relevant option is selected at export-time, and they will not appear in the Workbench for translation unless filtered for specifically. Those users able to alter segment states may change the state of these segments either by approving them for translation or excluding them altogether.

“**Excluded**” segments are those that have been deemed as not requiring translation at all. Unless the relevant option is selected, they are not included in exports and will not appear for translation unless filtered for specifically. Excluding a segment is not final, however: those users able to alter segment states may approve them for translation, making them available again.

3.6.2 Translation memory

Name	Target languages	Default	Actions
test		✓	⋮

Search for suggestion

Search for suggestions French (France) [fr-FR]

Translation

memory settings

Translation memories (TM) are used to store existing translations for segments. You can import translations from *.tmx files or populate the translation memory with translations you have added in the Workbench.

Translation memories are linked to your account and *not* your project. This way, if you want to, you can have one gigantic translation memory for all of your projects. This is useful if you have multiple projects of the same topic.

Clicking the floating action button brings up the **Create & assign Translation Memory** dialog. You can create a new TM or assign any other that you previously created. While you can add multiple memories to a project, only one of them can be the default for writes.

The memories added (created or assigned) to the project will show up in the list on the top card. You have multiple options for all of them:

- **Import/export:** Opens the dialog where you can upload a TMX file or request an export. Exports will be emailed to you.
- **Edit:** Brings up the Edit TM dialog where you have the same options you had when creating the memory.
- **Populate:** Allows you to add the content from this project to the TM.
- **Remove:** Unassigns the memory from the project. The memory isn't deleted. You can reassign it or assign it to any other project.

The screen is furnished with a search field you can use to look up segments in a TM.

IMPORTANT NOTE!

The TMs you manage on this screen and the Workbench are different from the Translation Memory that is billed: they are used for concordance lookups and automatic translation, and they are completely free to create and use.

On the other hand, the proxy TM associated with a project is a database in the cloud that holds translations of segments. When you reimport your XLIFF files or run auto-pretranslate, you are writing to this project TM, which is billed the first time around.

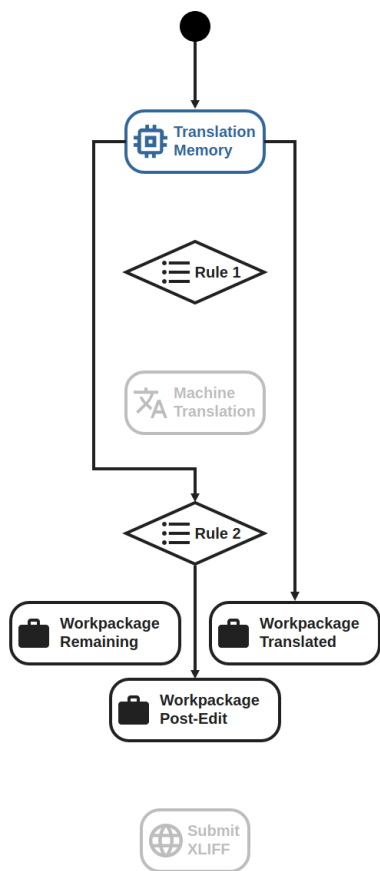
3.6.3 Auto pre-translate

Auto pre-translate

Automatically process newly added content according to the workflow defined below. Use the presets as a starting point, then customize it. Segments are stored in Work Packages. Optionally, XLIFF files can be generated and submitted to external systems for further processing / post editing.

Presets

- ☒ **Translation Memory only** Translate new content using Translation Memories assigned to the project.
- ☐ TM & MT
- ☐ Machine Translation only
- ☐ Custom



Translate with Translation Memory

Pre-translation threshold

99 %

☒ Confirm threshold

100 %

☐ Strict locale 

☒ Pass on unconfirmed segments to Rule2 (e.g. for external post-editing)

pre-translate settings

Auto

Automatize pre-translation: you can set up the project to automatically execute a pre-translate process for any new

content that is found - you can hook it up with your TMs, or any Machine Translation services you subscribed to.

Pre-translation of incoming content can be done without user intervention or oversight. If new content is encountered, and at least one source is configured, a user-configurable timer starts counting down. Content is collected while this timer is running, and once it reaches zero, translation begins from the configured sources.

Content that cannot be translated (if no matches of acceptable confidence level were found in the TM or if the MT-engine returned no translation) gets packaged into a Work package that you can have sent as XLIFF files to external systems, such as XTM, XTRF, or your Dropbox account.

Just like with manual pre-translation, when you use a Translation Memory, the project's default memory will be used. By default, this is populated with the segments that you previously added. You can change this using the [Translation Memory](#) page, where you can add more content to your memory if you need to.

In order to use Machine Translation, you need at least one API configured for your project. You can access configuration for available Machine Translators by clicking *configure* next to the desired Translator, or by selecting them on in the menu bar.

3.6.4 Alarm

Alarm configuration

Set alarm conditions for email notifications

Snooze for 15 minutes between notifications.

☐ Trigger alarm above words translated in memory/second limit.

☐ Trigger alarm above words added to memory/second limit.

☐ Trigger alarm above page view/second limit.

Alarms

You can set up alarms to be notified of any actions in your project that could potentially lead to undue expenses. Enter the value for the timeout (snooze) between subsequent e-mail notifications, to prevent the Alarm from flooding you with e-mails. After that, check one of the alarm options and enter the limit above which you want to receive a notification. You can set an alarm for:












Page Views: if you wish to detect unusually large amounts of requests to the proxy. As Scans and Discoveries count as Page Views, you can also use this alarm to detect if a Scan was started.

Words added: if you wish to receive notification about new content that exceeds the set amount.

Words translated: if you wish to receive notification about Machine Translation or reimport events on the project.

The e-mails are sent to the owner by default, but you can add the ‘Receive notification emails’ right to any participant in Dashboard > Sharing settings.

3.6.5 Audit log

Show before: 26 / 09 / 2020		⊗ Any user ▾	Any type ▾	RELOAD
Author	Created	Event name	Memo	Event description
	September 08, 2020. 16:28:17	Crawl job created		Crawl type: Wordcount (Discovery) Page limit: Unlimited
	September 08, 2020. 16:27:11	Project changed		Changes: default staging domain: example.com
	August 25, 2020. 14:24:48	Multiple languages added		 French (France) [fr-FR]  English (United Kingdom) [en-GB]
	August 25, 2020. 14:24:07	Crawl job created		Crawl type: Wordcount (Discovery) Page limit: 10
	August 25, 2020. 14:23:39	Project changed		Changes: default staging domain: example.com
	August 25, 2020. 14:23:34	Project changed		Changes: Staging domains: example.com
	August 25, 2020. 14:22:45	Invitation redeemed		Invitee username: ***** Invitee email: *****@gmail.com
	August 25, 2020. 14:22:23	Invitation sent		Invitation sent to *****@gmail.com
	August 25, 2020. 10:39:43	Project created		
LOADED				

log

The audit log is the minutes-taking feature of the proxy. It records all activity on the project with a username/e-mail address attached to each entry.

You have multiple options to filter the events that are displayed. Using the dropdown menus on the top, you can view entries created after a specific date, filter for a specific user’s actions or the type of the action.

You can also add memos to the entries by clicking on them. This way, you can track *why* a specific setting change was necessary.




3.6.6 Sharing settings

In the world of website localization, it is rare for a project to be realized by a single person. As project owner or backup owner, use the Sharing settings dialog to invite people into the project and assign roles/editing roles to any one of them.

At the bottom of the section, you will see the button, “Invite people”. Clicking on it, or the blue + button opens a dialog that allows you to enter the details of the project participant to be added. Using the dropdowns, you can assign languages, enable admin/access features and workflow roles for each user you invite. **These settings are not final, you can change them later.**

Features

Sharing settings

User	Role	Assigned languages	Features	Workflow role(s)	Notification emails	Actions
 proxy.user@example.com	Owner (you)	All	All	All	✓	⋮
 Marcus Forrest randomname@generated.earth	Admin (Backup Owner)	All		All		⋮

Invitations

There is no pending invite

[INVITE PEOPLE](#)



Sharing

Settings Screen

The various **features**, (administrative rights) deserve to be detailed, Here they are:

All features: every feature you see below the separator in the list will be added for the user. Note that this has a potential to cause problems: If, for example, you simultaneously set **Advanced features** and **Manage segments**, it results in an editing rights conflict. There are security implications you need to consider when giving your users rights, so use this feature with care.

XLIFF export/import: enables XLIFF export/import of segments for the user. Especially important for those users who will be coordinating the translation effort throughout the project.

Receive notification e-mails: Add user to the list of recipients of Discovery/Content Scan/Alarm, etc notifications.

Can invite others: designates the user as a Project Manager, able to invite others onto the project. Take care not to include other, conflicting roles, such as **Customer**, which could re-restrict access.

Advanced features: give the user the power to edit languages as well as any entry in the project, and editing most setting, up to, and including, the URL inclusion-exclusion rules. Segmentation, publishing and certain advanced settings remain beyond the reach of the user.

Manage segments: make the user a Customer, with the ability to manage (approve or exclude) pending source segments. The user will also require a workflow role and an assigned target language to access the Workbench to manage segments. **WARNING! This right restricts the user to this role, other features will be disabled!**

NOTE Features are predominantly important from a project management viewpoint. **Workflow roles**, on the other hand, are used to manage user interaction **on the Workbench**, based on expertise (whether a user is a translator, proofreader or client will decide what sets of segments are available to them for editing - see the [documentation of the Workbench](#) for details).

After a user redeems their invitation using the link they receive in an e-mail, their address and username will appear under the owner. Their features, and workflow roles and whether they should receive e-mail notifications can be edited by clicking on their line in the table.

You can invite as many participants to the translation project as you'd like.

Project Roles

Based on the project features that are available for a user, the following project roles are available:

- **Owner:** Every project has an owner with unlimited powers over the project: the owner may add or remove anyone on the project, edit any entry in any language, including adding new languages, any change any setting,

including the advanced ones. There can only be one owner on a project, but owners may renounce ownership, designating another user and setting their own privileges.

- **Backup Owners:** same as project owners. They can change any setting and entry, add or remove users, and modify language settings.
- **Project Managers:** can invite others onto the project. Other features and roles can be added as well. Take care not to include other, conflicting roles (restrictions override rights)
- **Advanced Project Managers:** they can edit languages, as well as any entry in the project. They can change most setting, up to and including the URL inclusion-exclusion rules. They cannot change segmentation rules or publishing settings. A few Advanced settings are also beyond their reach.
- **Linguists:** have access to the Simplified Dashboard and Content menu. They are only allowed to edit segments in their designated language and workflow role. They can receive notification emails about project updates, and may be given the power to import/export XLIFF files.
- **Contributors:** default users, capable of editing any entry in their selected language and workflow role, but nothing else. They may receive notification emails and project update emails, but they may not edit their features, nor invite anyone else, nor access any of the advanced settings.
- **Customer:** This role intended for a representative of the customer who exercises oversight over content. This role is read-only, meaning despite the target language added to the user, they are unable to actually make edits. They can *Approve* or *Exclude* segments.

Client Approval

Sometimes there is need for the client to view and check new segments before they enter the translation workflow.

In the **Advanced Settings** menu, default state of new segments can be set to the following:

- Pending
- Excluded
- Approved

When set to Pending or Excluded, new segments picked up will acquire that state automatically and manual approval is needed. In the Sharing Settings menu, the customer role can be assigned for this purpose.

Important: The client role is read-only. Anyone assigned this role won't be able to edit content, but approve them for translation. If you are unable to edit segments on the Workbench, please check Sharing Settings first.

3.6.7 JSON/JS/XML processing

These features used to be under Advanced settings but as they are the most important, they were moved to their own section.

The interface is split into 2 main sections: Live JSON/JS/XML Path Config and JSON path tester.

Live JSON/JS/XML Path Config

This section allows you to edit the settings directly. However, in case of the JavaScript translation options, we recommend that you use the JSON Path tester tool instead.

JavaScript translation options

This field contains the capture group definitions used to extract attribute-value pairs from JavaScript files selected for translation/localization. After entering the capture parameters and re-crawling the site, the proxy will display the selected JavaScript files as translatable pages in the pagelist, from where they can be selected for translation in the List View like regular pages, and any values for the selected attributes will be made available as translatable entries, which are treated identical to regular entries.

Entering “ `html`” (note that the switch is separated by a space!) after the path specification will result in the proxy applying its HTML parser to the match instead of a plaintext parser, stripping out HTML markup and only offering the actual content for translation (otherwise, should the match contain markup, the translator must take care not to alter it, or risk breaking the translated site).

If a field of the JSON being parsed contains further JSON data in a stringified form (`"key": "{\\"key\\":{\\"key\\":\\"value value value\\"}}"`), the path can be passed to a recursive JSON translator by appending “ `json`” to the path, then extending the path on the next line by adding “ `.json`”.

Mark resources as translatable

Using fully qualified URL prefixes, including protocol, host, and possibly path structures, like `https://www.example.com/path/to-be-marked/`, the Proxy can enforce dictionaries over multiple resources in a single rule. This is especially useful if the site under translation contains an API (especially CREST APIs) whose responses also require translation, and each endpoint is served on a different path. In this case, entering the root of the API here will automatically capture all responses from that path without having to individually mark them as translatable from the Resources menu.

XPath Translation

The proxy can translate XML (eXtensible Markup Language) files sent by the remote server, according to the XPath standard of specifying elements of the XML structure. Similar to JavaScript translation, entering the “ `html`” switch will result in the HTML parser being applied, while no switch will parse the match as plaintext.

JSON path tester

LIVE JSON/JS/XML PATH CONFIG

JSON PATH TESTER

Temporary JSON paths

Test, import or save expressions

```

*.profession
**.type
"testFunction"."post".1."%".alert".0
"testFunction"."post".1."%".skip".*
"testFunction"."post".1."%".skip".onlythis".! skip - don't forget to put an exclamation mark at the end of the given rule when you use skip

```

RESET

ADD PATHS TO LIVE CONFIG

SAVE PATHS

SOURCE CODE

ANALYZED CODE

```

1 /*var json = {
2   "name": "Peter Parker",
3   "type": "employee",
4   "profession": "photographer",
5   "cameras": [
6     {
7       "profession": "true",
8       "type": "analog"
9     },
10    {
11      "profession": "true",
12      "type": "digital"
13    },
14    {
15      "profession": "false",
16      "type": "DSLR"
17    }
18  ]

```

+

JSON

Path Tester

We'll use the following JavaScript snippet in the remainder of this section. It illustrates many use cases for JS translation:

```

(function () {

  var exampleVar = "Hello World!";

  var exampleUrl = "https://www.example.com";

  var exampleHtmlString = "<p>Hello World!</p>";

```

(continues on next page)

(continued from previous page)

```

var exampleObject = {
  "sentence01": "Hello World!",
  "sentence02": "Hello Again!",
  "nestedObject": {
    "sentence03": "Hello World!",
    "sentence04": "Hello Again!"
  },
  "exampleArray": [{ "value": "foo" },
                    { "value": "bar" },
                    { "value": "baz" }],
  "exampleNestedJS": "var nestedVar = { nestedKey: \"Nested sentence\"}",
  "exampleNestedHTMLinJS": "var nestedHTML = \"<p>Hello world!</p>\""
};
})();

```

You can copy & paste code into the source code field or if you have the URL you can fetch the entire file via the + button on the bottom right. If the file is minified, you can use the Format code button for better readability. When you click on Analyze code, the file/text will be requested/sent for analysis in the cloud. Once it's finished, you get a highlighted representation of the same code in the Analyzed code tab.

Click on any of the blue + icons to generate a **JS path** for the string in question. They will be added to the Temporary paths field. If you generate paths for all available strings in the example, and add a few processing modes, the list of paths in the upper text field should look like this:

```

"%".exampleVar"
"%".exampleUrl" url
"%".exampleHtmlString" html
"%".exampleObject."sentence01"
"%".exampleObject."sentence02"
"%".exampleObject."nestedObject".*
"%".exampleObject."nestedObject"."sentence04".! skip
"%".exampleObject."exampleArray".0."value"
"%".exampleObject."exampleArray".1."value" skip
"%".exampleObject."exampleArray".2."value"
"%".exampleObject."exampleNestedJS" javascript
"%".exampleObject."exampleNestedHTMLinJS"

```

Some of these paths require adjustment before they'll behave correctly.

Supported strings are highlighted in **red**, and those that are already covered by a listed JS path are highlighted **green**. Your results should look

Temporary JSON paths

Test, import or save expressions

```
"%".exampleVar
#"%".exampleUrl url
"%".exampleHtmlString html
"%".exampleObject."sentence01"
"%".exampleObject."sentence02"
"%".exampleObject."nestedObject".*
"%".exampleObject."nestedObject"."sentence04".! skip
"%".exampleObject."exampleArray".0."value"
"%".exampleObject."exampleArray".1."value" skip
"%".exampleObject."exampleArray".2."value"
"%".exampleObject."exampleNestedJS" javascript
"%".exampleObject."exampleNestedHTMLinJS"
```

[RESET](#)
[ADD PATHS TO LIVE CONFIG](#)
[SAVE PATHS](#)
[SOURCE CODE](#)
[ANALYZED CODE](#)

```
(function () {

  var exampleVar = "Hello World!" PLAIN -;

  var exampleUrl = "https://www.example.com" +;

  var exampleHtmlString = "<p>Hello World!</p>" HTML -;

  var exampleObject = {
    "sentence01": "Hello World!" PLAIN -,
    "sentence02": "Hello Again!" PLAIN -,
    "nestedObject": {
      "sentence03": "Hello World!" PLAIN -,
      "sentence04": "Hello Again!" SKIP -,
    },
    "exampleArray": [{ "value": "foo" PLAIN - },
      { "value": "bar" SKIP - },
      { "value": "baz" PLAIN - }],
    "exampleNestedJS": "var nestedVar = { nestedKey: \"Nested sentence\"}" JAVASCRIPT -,
    "exampleNestedHTMLinJS": "var nestedHTML = \"<p>Hello world!</p>\"" PLAIN -
  };
})();
```

[ANALYZE CODE](#)

Path

results

When you have all the JS paths you need, click Save paths (Replace live config, if there are existing paths) or Add paths to live config. Unless you know for sure that you don't need the previous live config, we recommend that you simply add the new paths.

Keys / Variables

Translatable elements are specified by a dot-separated list of words, each optionally double quoted and constituting either a.) a valid JS variable/JSON key name or b.) a token specifying one or more hierarchical levels (anonymous function, array index or globbing mark).

```
var exampleVar = "Hello World!";
```

The simplest possible case would be `"exampleVar"` to mark the value of the top-level element `exampleVar` as translatable. Anonymous function calls are denoted with `"%"`, and since the entire block of variables is wrapped by an anonymous function `(function () { ... })()`, this leading percent sign shows up in each case. Paths for dynamic JSON responses should be prefixed with `"json"`.

Globs

Use an asterisk (or Kleene-star) to collapse a single hierarchical level. E.g., the value of `"exampleArray"` is an array of objects. To include every index in the array, you can roll three rules into one:

```
"%".exampleObject.exampleArray.**.value
```

Double asterisks are even more inclusive: they recursively glob all child nodes. Exact specification can be restarted by following `**` with a double-quoted form. That is, the rule

```
`%`.**."value"
```

marks any variable or property called `value` it finds *at any hierarchical level* within an anonymous function call. If a JS path *ends* with the `**`, then the entire subtree is marked as translatable. Incautious use of this construct is not recommended.

Processing Modes

Nodes are processed as plain text by default, but you can enable specific processing modes with whitespace-separated postfixes. The available processing modes are `url`, `html` and `javascript`.

URL

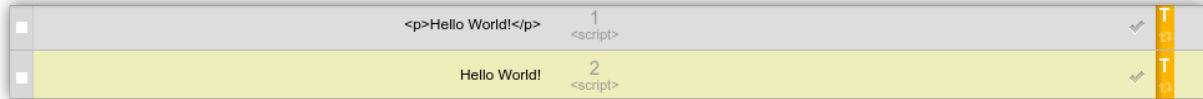
Variables can contain either *the project URL* or some other important location (such as that of a *linked project*) that you would prefer to have **remapped** over the proxy. Don't give in to the temptation to localize URLs in JS as plain text! Instead, use the `url` postfix to map them:

```
"%".exampleUrl url
```

HTML

`exampleHtmlString` demonstrates the fact that JS variables frequently hold markup (for better or for worse). The `html` postfix lets you process these strings as HTML.

```
"%".exampleHtmlString html[@process]
```



Extraction

with and without HTML-processing

The screenshot above demonstrates the difference HTML-processing makes. Picking up HTML-markup explicitly as text is generally considered error-prone and disadvantageous from a localization viewpoint, and isn't recommended.

[@process] is optional. By adding it, you instruct the proxy to apply the translation-invariable regular expressions currently set on the project.

Nested Javascript

Although JS paths are mostly specified in a single line, the `javascript` postfix bends this rule. It tells the proxy to apply the rule in the *next line* to the value of the postfixed JSON path. One level of nesting is supported. It is rarely needed, but invaluable when it is called for.

Plain text:

```
"%". "exampleObject". "exampleNestedJS" javascript
"%". "exampleObject". "exampleNestedJS". "nestedVar". "nestedKey"
```

HTML:

```
"%". "exampleObject". "exampleNestedHTMLinJS" javascript
"%". "exampleObject". "exampleNestedHTMLinJS". "nestedHTML" html
```

Note: The JSON Path tester tool is **not equipped** to display the nested use case.

JSON

For nested JSON elements, the specific `json` selector is also available. Use this mode to mark nested code parts that only contain JSON, as opposed to complete JavaScript code.

Skip

Use this processing mode to mark child node not to be translated. **Order is important!** If you use `*` or `**` to select child nodes to be translated, you can use the `! skip` processing mode to select a child element as non-translatable, but please note, that skip rules must come after the generic `*` rule. Skip is the lowest level mode, but you can increase the priority by adding `!` or `!{number}`. The `!` switch is not only for skip rules, any rule can be appended to increase its priority above other matching rules. You can also comment out elements by adding `#` at the start of a line. In this case the parser won't process the given rule at all.

```
#"%". "exampleUrl" url
"%". "exampleObject". "nestedObject". *
"%". "exampleObject". "nestedObject". "sentence04". ! skip
"%". "exampleObject". "exampleArray". 1. "value" skip
```

JavaScript template literals

Template literals are similar to strings but they support inline variables. They are surrounded by backtick (‘) characters. They are different from regular strings in that they support inline variables and expressions. For example,

```
let publishedText = "Published on " + date;
```

could be expressed as

```
let publishedText = `Published on ${date}`;
```

Cases like the above (or literals without variables) are now supported. The expression will become a single tag in the Workbench and as an x tag (that doesn’t have a separate closing tag) in XLIFF exports.

As an example, consider the snippet below:

```
let getUsername = () => {
  // maybe some asynchronous stuff here that we omit for brevity
  // let's just assume that the result is "user-docs-author"
  let result = "user-docs-author";
  return result;
};

let output = `This part of the documentation was written by ${getUsername()} `;

console.log(output);
```

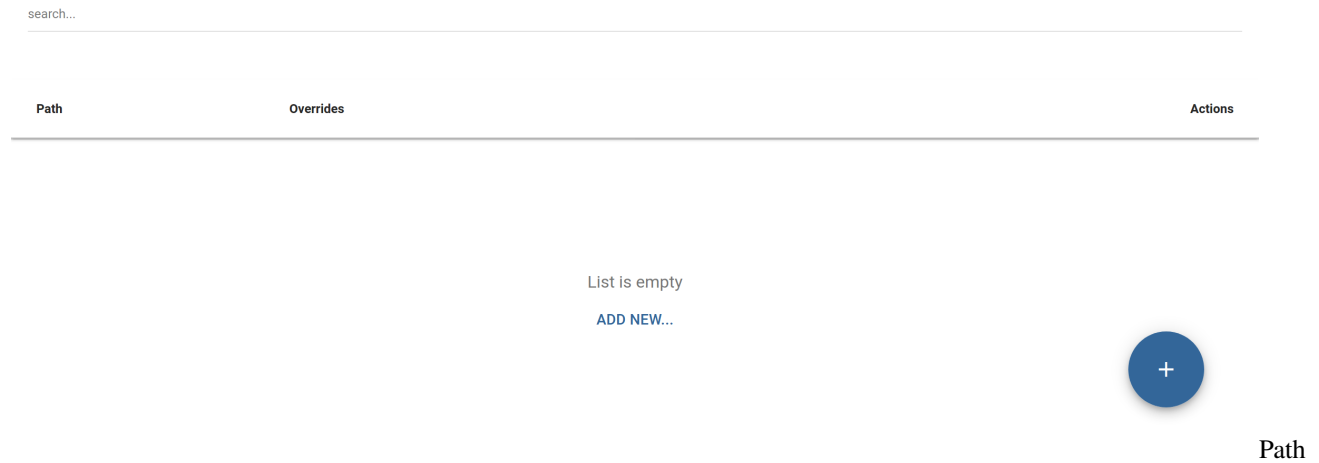
You can translate both `user-docs-author` and `This part of the documentation was written by <tag>` by adding the following JSON paths:

```
"getUsername"."result"
"output"
```

Note that content in template literals can only be processed *as text*. The above-mentioned processing rules, such as URL and HTML, can’t be applied.

3.6.8 Path prefix overrides

The proxy supports the declaration and use of rules called path overrides (alternatively called *Subtree Overrides*). They can be used to apply a variety of changes to an exact (e.g. `/about-us.html`) or prefix paths (e.g. all pages under `/products`).



prefix overrides dialog default view

A project has no overrides by default. Besides the search field at the top, the “Add new path” button is available. Click on it to type or copy & paste your path/prefix. Click on the “Add new path” button to add it to the list. The option to choose the type of the path (exact URL or prefix) is available as a dropdown.

You’ll notice that adding new paths and setting up corresponding overrides are done at two separate stages. Any path can have more than one type of override associated with it (but overrides differ in how many of each can be present on a given path).

After adding a path prefix, you can click on it to open the settings of individual overrides:

/wp-content*

DEFAULT CHARSET

✓ **CACHE & COOKIE**

CONTENT-TYPE

CONTENT SEARCH & REPLACE

JSON PATH

SPECIAL FORMAT

EXTRACTOR

JAVASCRIPT CONTENT INJECTION

JAVASCRIPT LINK INJECTION

CSS CONTENT INJECTION

CSS LINK INJECTION

Cache & Cookie

This override allows you to declare the Cache-Control and max-age headers for the prefix or URL and optionally clear the cookies. Important! Do NOT add overly general paths or too large max-age values without considering the effects! Please read through [documentation](#) before using the feature. Only one cache & cookie override may be present on each path or prefix.

Cache override public ▼

Max age 86400

Cookie override clear cookies ▼

[SET DEFAULT VALUES](#)

CANCEL

SAVE ALL

set of edits before saving changes

Keep in mind:

- overrides are applied only on **intra-domain** paths (elsewhere called the project URL). They cannot be applied on any path referenced on an external domain.
- overrides are applied early on in the pipeline, before any translated text is inserted. They are also **not** target-language specific.

A

- query parameters (`?q=value` etc.) are **not** supported. Do not add them to path overrides.
- When declaring a content type, make sure you use an *exact* description: `text/html` does not equal `text/html; charset=utf-8` or vice-versa. Be specific.

Important! *Add/edit multiple paths and overrides at once, but remember that all changes are **UNSAVED** until you click on “Save All”!*

3.6.9 Default Charset

Overview

This override is presented as a simple text field, the contents of which will be used for `<meta charset="...">` tags. Useful in cases where the original site is declared to be in an encoding that is incompatible with one of the target languages.

Only one charset override may be present on each path or prefix.

Parameters

- *Default Charset*: the string entered here will be used to override the charset attribute on the localized site.

3.6.10 Cache and Cookie Overrides

Overview

This override allows you to declare the `Cache-Control` and `max-age` headers for a prefix or URL and optionally clear the cookies.

/wp-content*

DEFAULT CHARSET

✓ **CACHE & COOKIE**

CONTENT-TYPE

CONTENT SEARCH & REPLACE

JSON PATH

SPECIAL FORMAT

EXTRACTOR

JAVASCRIPT CONTENT INJECTION

JAVASCRIPT LINK INJECTION

CSS CONTENT INJECTION

CSS LINK INJECTION

Cache & Cookie

This override allows you to declare the Cache-Control and max-age headers for the prefix or URL and optionally clear the cookies. Important! Do NOT add overly general paths or too large max-age values without considering the effects! Please read through [documentation](#) before using the feature. Only one cache & cookie override may be present on each path or prefix.

Cache override public ▼

Max age 86400

Cookie override clear cookies ▼

[SET DEFAULT VALUES](#)

CANCEL

SAVE ALL

An

example of a typical cache override (/wp-content)

The path seen in the screenshot above is a typical use case: it ensures that resources on the `/wp-content` prefix, associated with WordPress sites, can be cached for 24 hours.

Setting `cache-control: public, max-age=86400` on a URL/prefix in this way broadcasts to the network that the resource(s) there can be publicly cached. Depending on the location of the caching node and the pathway of the request, the content will be served from caches instead of going through the proxy pipeline.

This is beneficial for both speed and cost reasons. What is otherwise tolerable server load on the original site might be

unnecessary page view cost overhead over the proxy (with speed overhead not being much of a concern). We provide this capability as a useful cost optimization strategy.

Important! Do NOT add overly general paths or too large `max-age` values without considering the effects! Please read through [our description of the issue](#) before using the feature.

Only one cache & cookie override may be present on each path or prefix.

Parameters

- *Cache Override*: One of *Private*, *Public*, or *Ignore*
 - *Private*: sets `cache-control: private`, preventing caching at all
 - *Public*: sets `cache-control: public`, allowing caching
 - *Ignore*: leaves the header as-is, without modifying the directive
- *Max age*: the maximum age of a cached resource, in seconds.
- *Cookie override*: preserve or clear the `set-cookie` headers - may improve caching, but break functionality!

3.6.11 Content Type

Overview

Override the Content-Type HTTP header for a given path or prefix. Frequently used with template URLs or JS resources with mischaracterized Content-Types, it is sometimes useful to avoid encoding or character escaping troubles.

Multiple `Content-Type` overrides can be added on each path or prefix, but no two such fields may match.

Parameters

- *From* and *To*: the content types defining the mapping.

3.6.12 Search and Replace

Overview

Run a search & replace on the page or pages on the prefix. This lets you apply simple changes to the page source. You can choose between string or regexp replacement.

Add Content Search & Replace

Content-types: text/html.*


Search:

class="hello-world"

Replace:

class="hello-world foo"

☐ Regexp

Target languages:  de-.*

CANCEL

ADD

sequence of search & replace rules on a prefix

In the example above, search & replace is used to extend the list of classes (with a new class `foo`) for any element that possesses only the *hello-world* class.

Content-types:text/html.*

Search:

(Hello) (World)

Replace:

\$2, \$1



Regex

Target languages: de-+

CANCEL

ADD

Using

regex backreferences

The replacement field supports regex backreferences via the `$n` format, as in the screenshot above. The example looks for “**Hello World**” in the page source, stores backreferences to each letter group and reverses their order, resulting in the output “**World hello**”.

Refer to the [java.util.regex.Pattern documentation](#) for the details of the supported regular expression format.

Note that search & replace is a “naive” operation both for strings and regular expressions: HTML is not parsed, nor JavaScript evaluated at this stage (roughly, S&R runs as if it were working on a plain text source file). It cannot be used to solve recursive tasks.

Multiple S&R overrides can be added on the same prefix, and they will be applied *sequentially*. No two replacement strings/regexps may match, the dialog will display an error if you attempt to enter the same replacement rule twice.

Parameters

- *Content-types*: a pattern determining what content types the rule will apply to
- *Search* and *Replace*: patterns that determine replacements. If *Regex* is selected below, capture groups and backreferences may be used.

- *Regex*: treat the above patterns as regular expressions, enabling the use of capture groups and other features.
- *Target languages*: a pattern that determines which locales the rule will apply to. These patterns are applied to the locale codes (such as `de-DE`) and are case-sensitive! Omit to have the pattern apply to all locales.

3.6.13 JSON Path

Override the JS/JSON paths that you have defined in Advanced settings under the menu JavaScript translation options (JSON / variables). With this option, you can use different paths for a given prefix, and optimize the translation of the JavaScript files.

We recommend that you use the [JSON path tester](#) tool to gather these paths.

Parameters

- Override search paths: path-specific JSON paths

3.6.14 Special Formats

Overview

These options enable specialized, non-standard handling of certain documents.

Parameters

- *ASP Format*: used to enable specialized processing of AJAX responses sent by ASP.NET-powered servers.

3.6.15 Extractor

Overview

In some cases, the remote server response deviates severely from the industry standards, but still requires translation. In order to handle these, the proxy has to slice up the incoming stream to extract the relevant contents, then restore the original after processing. This is achieved by using regular expressions to designate patterns requiring handling within unprocessable strings.

All regular expressions are *Java Patterns*. See the [official documentation](#) for the finer points.

Parameters

- *Content-type pattern*: a regular expression that designates `content-types` susceptible to the extractor operation.
- *Check first regexp*: a regular expression that is applied to the *incoming content* to see if it should be extracted.
- *Extractor pattern*: a regular expression that locates and extracts content from the incoming string. *Must* contain at least one capture group!
- *Content prefix*: an *optional* prefix that is prepended to the extracted string before handling. Will be stripped before content is re-inserted into the original!

- *Content suffix*: an *optional* suffix that is appended to the extracted string before handling. Will be stripped before content is re-inserted into the original!

Note that at least one of the last 3 input fields must be filled in.

Example

For example, consider the following snippet being sent:

```
xxx#It's so beautiful!#It's not so beautiful!#xxx
```

1. First of all, the content type must be verified to avoid trying to process images and the like:*Content Type Pattern*: `text/*`
2. Then the content is validated to see if starts with at least one `x`:*Validation Pattern*: `x+`
3. The content is targeted:*Extractor Pattern*: `\\# (.*) \\#`
4. Optionally, HTML-processing may be forced using prefixes and suffixes to transform the content prior to handling:*Content Prefix*: `<p>`*Content Suffix*: `</p>`

Internally, the proxy will see the following when translating:

```
xxx# <p>It's so beautiful!</p> # <p>It's not so beautiful!</p> #xxx
```

This is then transformed:

```
xxx# <p>Es ist so schön!</p> # <p>Es ist eigentlich nicht schön!</p> #xxx
```

Finally, the original form is restored before the content is sent out:

```
xxx#Es ist so schön!#Es ist eigentlich nicht schön!#xxx
```

3.6.16 Inject JavaScript or CSS

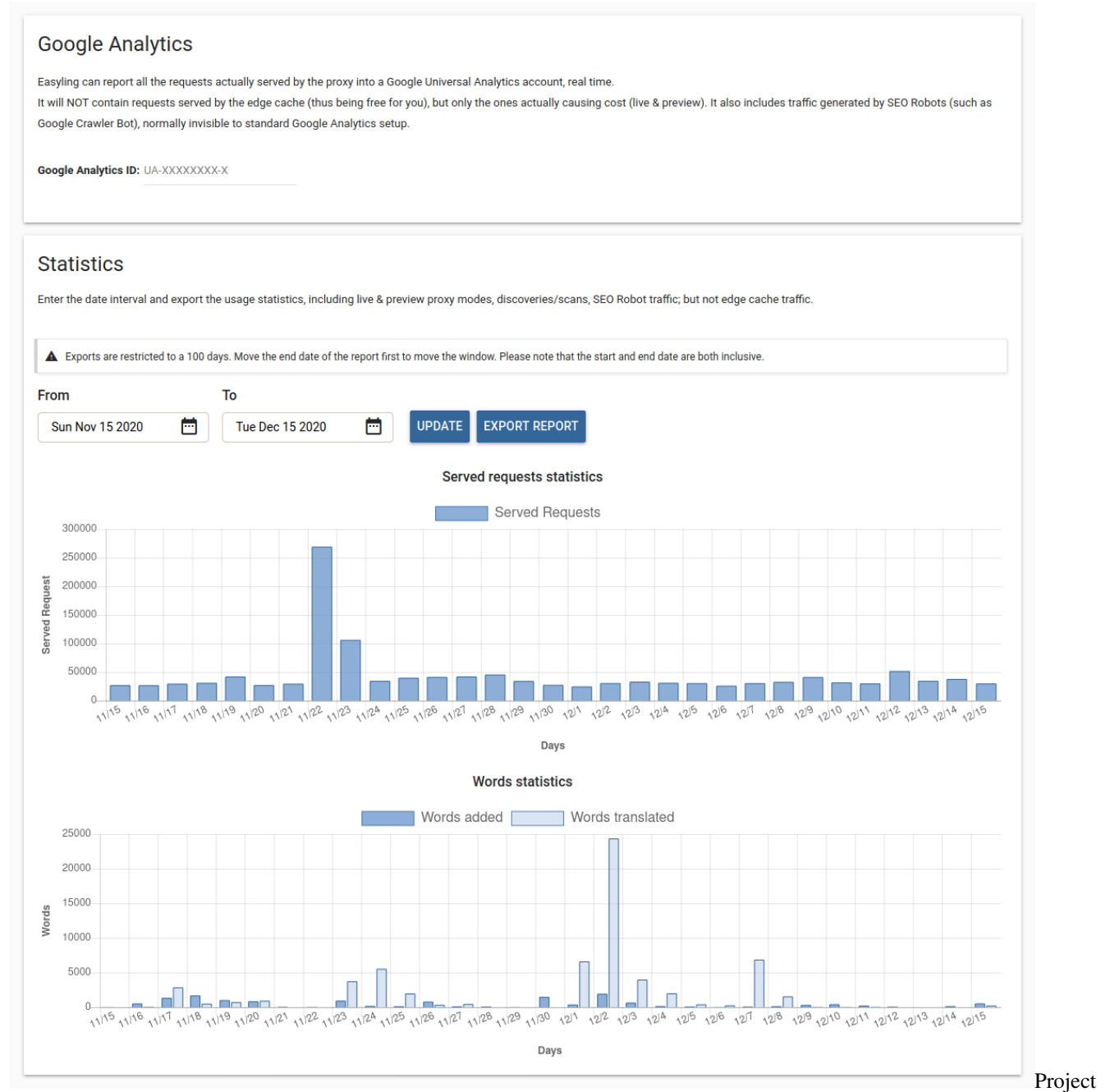
You can provide JavaScript or CSS code to change the default appearance and/or behaviour of the translated pages. The code will be included in the HEAD section of each and every translated page for the prefix or URL

There are 4 options:

- JavaScript content injection: enter the code you want to inject
- JavaScript link injection: enter a link pointing to the code
- CSS content injection: enter the CSS rules that you want to inject
- CSS link injection: enter a link pointing to the stylesheet

Note that the links you inject must be present on the project domain. In layman's terms, this means that if your project is created on `example.com`, you can't inject `notexample.com/cool-styles.css`. You can circumvent this restriction by creating a Content override and inject it as a link.

3.6.17 Project statistics



statistics screen

This screen shows you the most important statistics of the project as well as allowing you to add Google Analytics to it.

Google Analytics

In this section, you can enter a Google Analytics ID. When you do, the proxy will start reporting the requests it receives to Google Analytics. These are the requests that the proxy processes and thus are billed to you.

Statistics

This section gives you an overview of the main statistics that generate cost to you. Served requests are the same as those reported to Google Analytics if you have a code set up. Word statistics show you the amount of content you add and add translations for on a daily basis. You can select a date range of up to 100 days for these charts and for the exports. The export is created in the cloud and is emailed to you. It is an Excel spreadsheet containing the same information. This allows you to process it in external tools.

3.7 Page modifiers

3.7.1 General

The proxy makes translation of websites relatively easy, but the web is an admittedly complex environment that can surprise in countless ways: changes to the page language can cause various layout issues to surface, especially if the developers of the site did not anticipate that the site might get translated.

Word length differences between the source and target languages might cause the menu to crowd.

Differences between the lengths of different text blocks can cause otherwise well-designed CSS rules to behave in unexpected ways.

The plethora of plugins that run on a modern website - there is a big pool of possible glitches right there.

Fortunately, changing the site content over the proxy is relatively easy. The Page Modifiers function is made available for this reason: to empower you to add your own CSS rules and JavaScript snippets to influence the way a given proxied site looks to the user.

Because the datastream must pass through the proxy to have the translation embedded, the proxy can insert JavaScript modifiers, modify style sheets, even embed entire pages that do not exist on the original site.

Note that CSS rules and JavaScript are injected into each page that is served over the proxy, including page content overrides defined in Dashboard 2.0 > Content override.

3.7.2 CSS editor

The proxy can be used to insert locale-specific CSS rules into the site being served. The rules are inserted as the last element of the head on every page served through the proxy. A very common use case for this feature is RTL conversion of a website: almost always necessary when one of the target languages is Arabic.

It is good practice to make each CSS rule language specific (the proxy will always insert the appropriate locale code into the HTML):

```
html[lang="de-DE"] ul.one-selector li a {  
  float: left;  
}  
  
html[lang="fr-FR"] .another-selector h2 {  
  height: auto;  
}
```

If you omit the `html[lang="fr-FR"]`, the CSS rule will be applied in all target languages, which might not be the behavior you expect.

3.7.3 Javascript editor

An editor with syntax highlighting support is available on the Dashboard 2.0. You can add JavaScript code here, which will be injected into the `head` tag of each page. By default, it contains placeholder code that demonstrates page modifier use. You can either enter your code here or select the JavaScript link injection tab where you can just enter the URL of a script on the original site.

The following snippet also demonstrates a plausible page modifier:

```
(function() {

    "use strict";

    document.addEventListener("readystatechange", function() {
        if (document.readyState === "interactive") {
            resizeTextBoxOnContactPage();
        }
    }, false);

    window.addEventListener("resize", function() {
        resizeTextBoxOnContactPage();
    }, false);

    // make function page-specific. Alternatively, add check to eventListener above
    function resizeTextBoxOnContactPage() {
        if (document.location.pathname !== "/contact-us") { return };
        // implementation details here
    }

    // more function definitions here

})();
```

IIFEs (Immediately Invoked Functional Expressions), such as in the script above, are a generally good pattern to use with a page modifier. In this way, a programmer can continue to work within the confines of their local scope, and the function and variable definitions will not intrude on the global namespace - when it comes to page modifiers, it is very important that the modification has as little chance to clash with site code as possible (unless in a considered move).

The structure below can serve as an even simpler skeleton for a page modification:

```
(function () {

    "use strict"

    $(document).ready(modifyPage); // if jQuery is present

    function modifyPage () {
        console.log("modifications go here");
    }

})();
```

Of course, the possibilities of code injection are endless. See our tutorial for [site search integration](#) for a comparatively detailed example to help you get a glimpse of the possibilities of injected JavaScript.

The code injection feature puts the power of all client-side coding at your fingertips, and a truly in-depth discussion of the core web technologies is not possible in this documentation. The excellent [Mozilla Developers Network](#) contains all the details you could ever need, and the [W3schools](#) website contains useful tutorials on various web-related topics if you are just starting out.

Development Tip: a userscript extension, such as Tampermonkey can be used to develop page modifiers locally. While the Dashboard editor is adequate for basic editing, it is not meant to replace the many powerful editors available. Tampermonkey can `@require` your script from the filesystem if it is granted `file://` access (supported by Google Chrome), in which case you can use your editor of choice and retain preview-on-refresh capability.

But keep security in mind!

3.7.4 Page Content Overrides

The proxy relies on the original site for substance: both translatable content and server-side services are provided by the original. The proxy relays and processes request-responses between that server and the visitor.

PCOs, as they are usually abbreviated, are an exception to this rule. They are “virtual” pages that are either created completely from scratch (and thus may not even have a counterpart on the original) or they override URLs that are accessible on the original server with wholly custom content over the proxy.

If a visitor requests an URL over the proxy that has an associated PCO, then the corresponding request is not sent to the origin (even if the URL is made unique by query parameters) and the PCO is served instead. A PCO, in a similar fashion to responses from the original site, will go through the translation pipeline. This bears repetition: **the contents of a PCO can be extracted and translated.**

Headers

Content Types

The source need not be HTML: any custom content-type can be entered (as long as it is a textual type), such as `text/xml` or `application/javascript` along with customized cache headers and status codes.

Of the many content types that can be used, the behavior of `text/html` is special in an important way: CSS and JS page modifiers on the project will be **injected** into such PCOs. The corresponding `<style>` and `<script>` tags are injected even if the PCO is a simple HTML snippet (that is, it lacks an `<html>` or `<head>` tag).

Syntax highlighting is automatically activated for a subset of content types after the Header field is filled out.

Cache Control & Pragma

Loading overrides over the proxy also **count as page requests**, so it is recommended that you add an optimized `max-age` value to each PCO and make them publicly cacheable. Placeholders in each field indicate reasonable defaults. At the same time, you’ll want to set `max-age` to a smaller value *during testing*, in order to avoid having incomplete or work-in-progress PCOs cached for a long time over the network.

Response Codes & Location

The 300-family of status codes requires the Location header to be defined as well, to specify the target of redirection. Note also that only those HTTP status codes can be used that are permitted by the Java Servlet class.

Search result pages

The fact that URLs associated with a PCO are never requested from the origin server can be beneficial when **implementing site search**. If a distinct result page is used to display results, an empty version of that original can be copied verbatim into a PCO. This is particularly useful in cases where site search load is considerable over the proxy and the origin server would benefit from not having to handle search requests that it is not equipped to answer.

Scripts/Libraries

The PCO feature lets you add JS libraries or CSS stylesheets over the proxy as PCO with the appropriate Content-Type. A script declared in this fashion can be included in the `<head>` tag by going to Page modifiers > Javascript editor and **adding a reference to the PCO URL** at the top.

This approach is not recommended for jQuery or similar standard libraries. Inclusion of a script accessible via CDNs should be done using those external URLs, if not for any other reason, then because such external domain requests do not cost money over the proxy.

If, however, a standard library has to be changed in some small way over the proxy, PCOs can come in very handy.

Placeholders / Redirection

PCOs can be used to create “Under construction” pages or to ensure that the user is steered away from a specific URL via a redirection. But keep in mind that there is no prefix support for PCOs: they are always **exact URLs** and can only be applied as overrides in a targeted manner.

Binary Content

Binary content PCOs (including images and webfonts) are not supported. Use the resource localization features to replace images over the proxy with their translated counterparts.

3.8 Publish

3.8.1 JavaScript publishing

This section contains key settings for [Basic publishing](#) as well as the one line of JavaScript to embed the translations to the original website. There are four sub-sections.

Embed code

This part is the most important for the publishing process. You can embed the client-side translator engine via one line of JavaScript code that you can find here. Use the *Copy to clipboard* button, then add it to the website.

Embed code

When you're ready to publish, just copy and paste the tag below into the `<head>` section of your site, preferably as high up as possible. Once you then publish exports of the translations in the [Previous Exports](#) screen, they will be immediately available.

```
<script defer="defer" type="text/javascript" src="https://your_whitelabel /client/pr_code /0/stub.js"></script>
```

COPY TO CLIPBOARD

Embed

code screen

Tweaks

These checkboxes are fairly similar to those that you can find on the *Advanced settings* section. They influence the behaviour of the translator script.

Note that the majority of these tweaks change the embed code, so after changing them, please make sure to update the code on the website.

Tweaks

- ☐ Include all the default parameters values in the link.
- ☐ `languageParameter` : Change the default language selector key
- ☐ `storageParameter` : Change the default LocalStorage key
- ☐ `noXDefault` : Suppress placing an `x-default` link element in the `head` if a translated language is loaded. This may have SEO implications!
- ☐ `rewriteUrl` : Use `history.replaceState` to rewrite the URL shown to the user so that it always displays the selected language.
- ☐ `scriptUrlIsBase` : The loader will search for the translator script based on its own URL. **CAUTION: This is not supported under Internet Explorer!**
- ☐ `disableSelector` : The stub will not inject its own language selector in the sidebar. In this case, it is up to the website to provide links to the various language versions.

Tweaks

screen

There are seven options:

- *Include all the default parameters:* By default, the embed code is kept as short as possible. To achieve this, tweaks that aren't enabled aren't included in the URL. This tweak allows you to include them all.
- *Language parameter:* By default, you can change the target language by appending `?__ptLanguage=${target_language}` to the URL of the site. With this option, you can change it from `__ptLanguage`.
- *Storage parameter:* By default, the user's language choice is stored in LocalStorage under `ptLanguage`. Change it with this option.
- `noXDefault`: By default, the translator adds an `x-default` link element to the translated website. With this option, you can prevent this.
- *Rewrite URL:* By default, the target language is hidden in the URL. Use this tweak to ensure that `?__ptLanguage=${target_language}` is always present in the URL.

- *Script URL is base*: The injected script loads further translator scripts, one for each target language. Use this tweak to try loading them from the original site's domain. Use this feature if the JavaScript exports are uploaded to the original server. Note that this isn't supported under Internet Explorer.
- *Disable selector*: The translator script adds the sidebar language selector by default. Use this tweak when a custom language selector is in place.

Selectors

In this section, you can enter CSS selectors to influence the behaviour of the translator script.

Block selector

HTML documents, such as webpages, have block and inline elements. W3Schools has [a great page](#) about them. The translator engine treats them differently by default. Block elements become their own entry in the Workbench. If you need to treat an inline element as its own block element, just enter its CSS selector here.

Re-process after changes

The translation algorithm translates every element on the page once. When the text of an element is changed, but the element itself isn't, it isn't re-translated because at that point it may contain mixed-language content.

You can enter CSS selectors to force a translation update on change here.

WARNING: Using this feature can result in translated content, or partially translated content, being ingested as source content.

Selectors

Force specific processing on given elements based on their CSS selectors. If the element should be treated as a block element - making it a separate segment - enter in under "Block selector". If an element has dynamic content, enter it under "Re-process after changes". This runs the translation algorithm again whenever that element is changed.

Block selector

e.g. .block



Re-process after changes

e.g. #main-nav



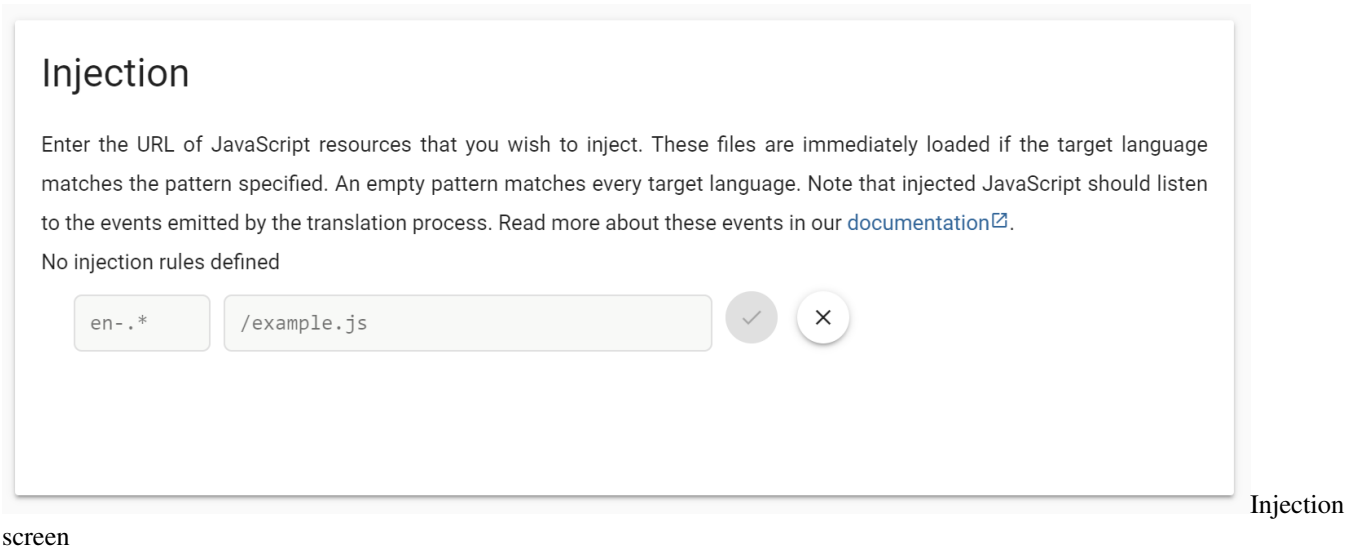
screen

Selectors

Injection

Here, you can specify JavaScript files that are injected to the website, even when the source language is selected. This way, you can make any change required both to the original and the translated websites. You can use this feature, for example, to customise the language selector or to add an overlay while the translations are being added to the website.

TIP: Write your JavaScript code to a Page content override making sure that the content type is set to `application/javascript`. Then, you can inject the temporary domain of this override.



Beta features

We recently added new features that aren't yet available on the Dashboard, but can be tried. Just drop us a line, and we can enable them for you.

- Custom flags for the language selector: By default, the same flags used on the Dashboard are used for the language selector. Now imagine you translate from `en-US` to `es-US`. In such a scenario, both flags would be that of the United States. Now, the flag can be replaced with any other flag. The English could become the flag of the United Kingdom or the Spanish flag could be used for `es-US`.
- Passive mode: With this mode enabled, the translator script won't show the language selector, and it won't persist the user's selection either. This allows the use of custom management solutions.
- Language selection based on path: If you have a page, like `example.com/fr/index.html`, the translator script can automatically select the French translations.

3.8.2 Per-language settings

In this menu, you are provided with the means of publishing your website on a pretty domain after translation tasks have finished.

The publish wizard has two options to make the site available to the web at large: serving domain mode and subdirectory publishing mode.

Domain settings

This screen opens by default after clicking on a *Target locale* in your list. Details of the locale and some settings are here:

- *Target language*
- *Temporary domain*: Similar to the Preview but works like the live site (useful if you have different snapshots). Used for Subdirectory publishing.
- *Status*: Lets you know when the locale isn't published yet.
- *Live domain*: If the locale is published it lets you know the domain.
- *Translation path prefixes*: When you publish in Subdirectory mode, the content is on the same domain as the source. As such the crawler may visit it and ingest the translated content as source. You can prevent this by entering the path prefixes that contain the translations here. They will be excluded from the project.
- *Serving mode*: Subdomain or subdirectory.
- *Override HTML lang attribute*: Every HTML page has an attribute that specifies the language of the document. By default, this is set to the locale code (like `en-GB`). It's advised by the [W3](#) to "keep the tag as short as possible. Avoid region, script or other subtags except where they add useful distinguishing information". To follow this advice, you can override the default using this field.

There are 3 tabs full of additional settings available as well:

Access control

You can add basic HTTP authentication to the proxied site in order to restrict access to it. Select the proxy modes that you'd like to limit access to and add a user by entering a username and password. You can add multiple users to the list.

Publish website wizard

Under this menu, you will be able to publish your website either in Serving domain mode or Subdirectory publishing mode.

STEP 1 - Serving domain

As a first step, you need to provide the domain name on which the translated website is to be published. The proxy will only publish a project on one domain name at a time, the question is which one should be the primary - naked domain (`example.com`) or subdomain (`www.example.com`)? Once that question is decided, you should then set up server-side redirection to the "main" domain (best practice prefers the `HTTP301 MOVED PERMANENTLY` status code with the `Location` header).

STEP 2 - Publishing mode

Serving domain mode (`fr.example.com`)

In the serving domain mode, the translation proxy will publish the translated site either on a subdomain of the original (the default behavior, such as `fr.example.com`), or on a completely separate naked domain (such as `example-fr.com`). In order to use this mode, you (or the client) will have to modify the DNS settings corresponding to the original domain - the four to ten records (four for subdomains, up to ten for new naked domains) that need to be inserted in your DNS settings are found under the *Verification* menu. These records will change as you enter or change the desired serving domain in the first step called *Serving domain*. They also change depending on the user who is currently logged in. On projects with multiple users, only the user who first got the record can finish publishing.

Subdirectory publishing (`example.com/fr`)

The alternative to subdomain-based publishing is to retain your own domain and publish the site as a subdirectory. I.e. the translated pages will appear under separate paths under the same domain as the one the project was created for (the original domain).

Due to the way the proxy works, this requires a reverse proxy configuration to be placed in front of the web server. A variety of load balancer/reverse proxy solutions are available on the market, with **nginx**, **CloudFlare** and **AWS CloudFront** being three of the most well-known solutions available. See the vendor documentation for the details of setting up a reverse proxy (do note that nowadays, reverse proxies are monumentally powerful network solutions, and discussion of all their features is beyond the scope of this introductory description).

Client-side translation


Client-side translation bypasses the proxy entirely, and conducts the translation process in the user's browser without passing data through third parties. This can be desirable for clients bound by legal or security restrictions.

To use Client-side translation, you need to create a new export (not a Work package), with the following settings:

- **File Format:** JS
- **Export:** All Entries
- **Unique segments only:** checked

Export bilingual file

File format Client side translation (JS, export only) ▼

Source language  English (United Kingdom) [en-GB]

Languages to export [fr-FR] ▼

Export range Project created ▼ , until Now ▼

Export all entries ▼

☒ Resources included

☒ Unique segments only


☐ Trim export to contain as few tags and whitespaces as possible (experimental, safe to use)

☐ Skip excluded and pending segments from export

☐ Split package by every _____ segments

☐ Copy source to target where empty (Wordfast Pro)

☐ Send XLIFF files to None ▼

 No external TMS configured!

[Configure XTM](#) [Configure XTRF](#) [Configure Dropbox](#)



EXPORT

Settings

for CST export

Once the export completes, it has to be published from the *Actions* column of the *Previous Export* screen, and the client must add the *loader script* to their website by inserting the following one-liner:

```
<script type="application/javascript" src="https://{{whitelabel-domain}}/client/{
↩️{projectCode}}/0/stub.js"></script>
```

<input type="checkbox"/>	Creation time	File format	File name	Target language	Word count	Status	Actions
<input type="checkbox"/>	June 03, 2020. 19:07:42	JS	dso.provingground.tech_ko-KR_1kht39i6_1591204054888.js		495		Actions
<input type="checkbox"/>	June 03, 2020. 19:07:41	JS	dso.provingground.tech_ja-JP_1kht39i6_1591204054740.js		495		Publish
<input type="checkbox"/>	May 18, 2020. 14:00:24	JS	dso.provingground.tech_ja-JP_1kht39i6_1589803221227.js		495	Published	

Publishing

a CST export

And constructing a language selector of their own, that adds the query parameter `__ptLanguage={{ locale }}`. Together, the two will enable in-browser translation of the site.

Should the need arise, the client can also download both the loader stub and the packaged dictionaries, and serve them

from their own webserver, so that contact with third parties is minimized.

STEP 3 [Selected publishing mode: Serving domain] - Verification

There are four CNAME settings that are required on your domain to enable publishing of your website. Each of the lines in the table that is displayed has a specific function:

ACCESS CONTROL

PUBLISH WEBSITE WIZARD

PATH SPECIFIC SETTINGS

✓

✓

3

4

Serving domain

Publishing mode

Verification

Summary

☒ CNAME
 ☐ TXT

Please set up the following DNS records under your domain for us to be able to serve the translated content via your serving domain. As a side effect, the Google Domain Verification process will provide access for proxytranslation@gmail.com to [Google Search Console](#) and an e-mail notification will be sent out to other Search Console users. Consider notifying the site owners in advance.

Type	Key	Value	Verified
CNAME	e4lwnmub3sps.www	gv-fblyuaasncptyo.dv.googlehosted.com.	✗
CNAME	www	ghs.domainverify.net.	✗
CNAME	6ywbfcfpy3h33ogglogzlrnqztnv3hvbvx3pf10.www	6ywbfcfpy3h33ogglogzlrnqztnv3hvbvx3pf10.domainverify.net.	✗

VERIFY

COPY RECORDS

BACK

NEXT

Serving

domain verification

CNAME 1 Allows mapping of the subdomain in Google AppEngine, enabling us to alter the datastream and translate the site. This row is computed from the serving domain entered in the first step called *Serving domain*, and needs to be added once per serving domain.

CNAME 2 Allows mapping of the naked domain in Google AppEngine enabling us to alter the datastream and translate the site. Either this or the previous record is required. We recommend adding both.

CNAME 3 The third line determines where the currently selected target language will be published. This defaults to the language code, but you are not obligated to keep it that way. This has to be entered separately for every target language you publish.

CNAME 4 The fourth line verifies ownership of the original domain. This is computed from the user's ID who is currently looking at the publishing page (i.e. different users will see different values, so one person should communicate this to the client and hit the **Verify** button), and needs to be added only once per serving domain as well. You can set the subdomain and domain where the translated site will appear.

These may appear in a different order.

After all the settings have been entered into the DNS records, there is short time while the changes propagate and are replicated across the world. This can vary wildly with the DNS and hosting providers, taking anywhere between one and twenty-four hours. It is recommended to wait out the twenty-four hours, as you will not be able to click on the Next button until all checks are passed.

HTTPS & SSL certificates

Additionally, if the original site was HTTPS, or the translated sites will be served over a secure channel, an appropriate certificate and its associated key will also be needed. Ideally a wildcard certificate (one certificate for all subdomain), but Extended Validation certificates can also be used, although they require more setup work.

Our support team can also provide you with the *Certificate Signing Requests* necessary for the generation of these certificates - this will also have the added benefit of not having to send the private keys over the wider web for recording in AppEngine.

Additionally, we have a *Managed Certificate* program, where the proxy handles SSL certification automatically for published websites. The *Managed Certificate* program has a cost of 50EUR (60USD)/proxied domain/**year** or 100EUR (120USD)/proxied domain/**three** years.

See the pertaining section of the documentation [here](#).

STEP 3 [Selected publishing mode: Subdirectory publishing] - CDN / Reverse proxy

We tried to provide you the absolute minimum example configurations required to achieve a workable reverse proxy using five popular web server/CDN systems. See the integration guide for them below:

- *nginx*
- Microsoft IIS
- Apache HTTPD
- *AWS CloudFront*
- *CloudFlare Workers*

✓

SERVING DOMAIN

✓

PATH

✓

PUBLISHING MODE

4

CDN / REVERSE PROXY

5

SUMMARY

☒ Custom prefix as subdirectory
 ☐ Language code as a subdirectory at depth 1
 ☐ Locale code as a subdirectory at depth 1
 ☐ Multiple locales as a subdirectory at depth 1

NGINX

MICROSOFT IIS

APACHE HTTPD

AWS CLOUDFRONT CDN

CLOUDFLARE WORKERS

```

location ~* ^/(fr) {
    resolver 8.8.8.8;

    set $xhost fr-fr-redacted.app.easyling.com;

    proxy_set_header X-TranslationProxy-Cache-Info    disable;
    proxy_set_header Server $xhost;
    proxy_ssl_name $xhost;
    proxy_set_header X-TranslationProxy-EnableDeepRoot true;
    proxy_set_header X-TranslationProxy-AllowRobots true;
    proxy_set_header X-TranslationProxy-ServingDomain $xhost;
    proxy_set_header Host $xhost;

    #old nginx
    # proxy_pass $scheme://$xhost;
    #new nginx:
    proxy_pass $scheme://ghs.googlehosted.com;
}

```

CANCEL

BACK

NEXT

CDN

/ Reverse proxy options

If you select an option, for example *Multiple locales as a subdirectory at depth 1*, it will only show you a different configuration as an example, but as we wrote previously, you need to set up your own reverse proxy or CDN, configured appropriately. You can also set up path prefix rules under the menu *Path Specific Settings*, and we will discuss these configuration options later.

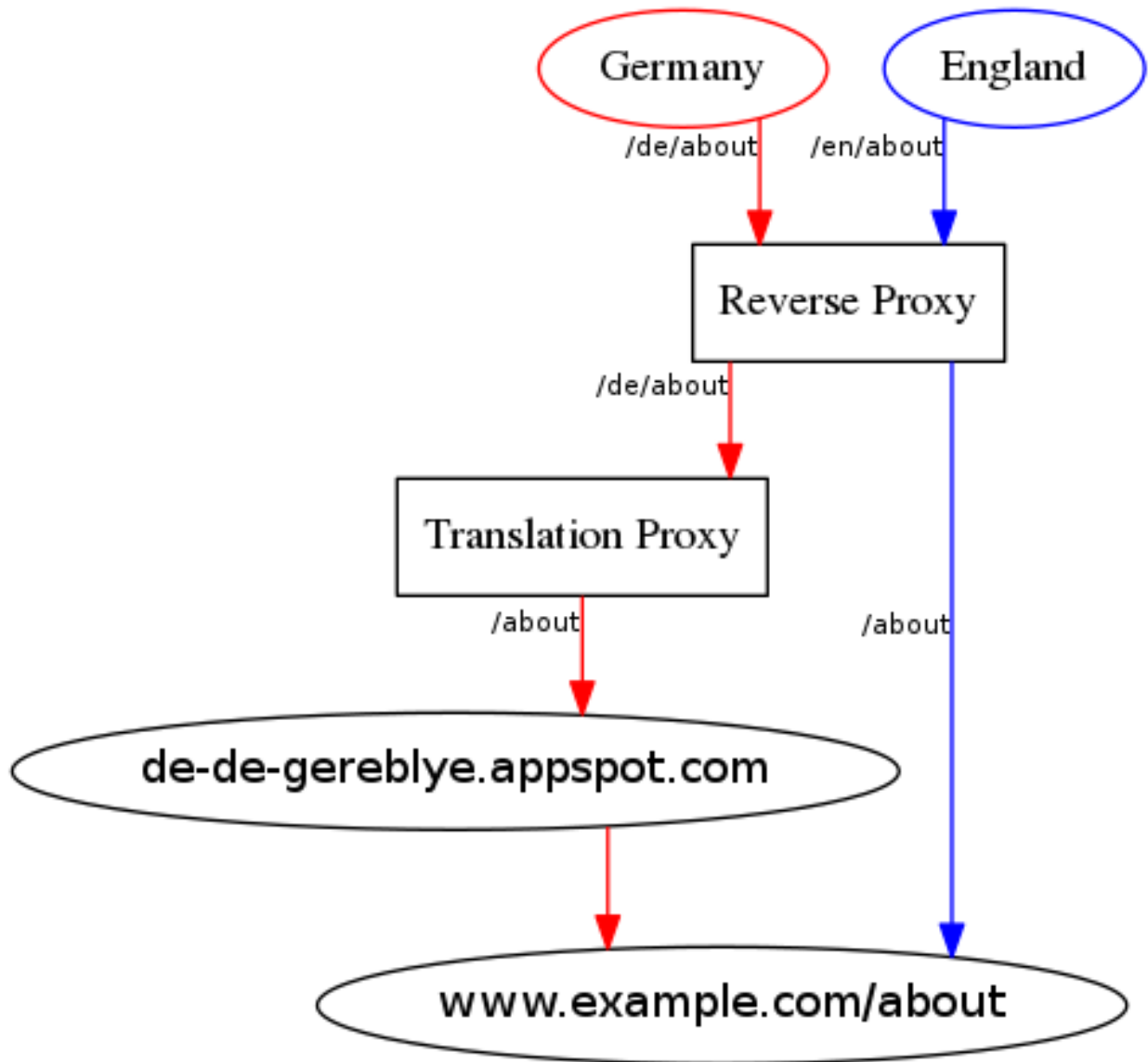
The goal of any reverse proxy configuration interoperating with a translation proxy domain requires that its configuration reflect the intent of the configuration examples.

Let's clarify these snippets with a more general, high-level explanation of a reverse proxy in operation.

Example:

Suppose that we know the following about a translation project about to be published using a reverse proxy.

1. The origin server domain is `www.example.com`.
2. Source language is English.
3. Translation exists for German
4. German serving domain: `de-de-gereblye.app.proxytranslation.com`



Reverse

proxy setup

The task of the reverse proxy, standing right at the beginning of the pipeline that will serve a client request, is to decide which target language is requested by the user and to ensure that the request is relayed to the appropriate domain that can respond with content in the appropriate language.

In our example scenario, the reverse proxy has to make one decision: is the user requesting a resource in English or in German?

Translation proxy

From our perspective, the most interesting case is when a user from Germany requests `www.example.com/de/about`, the reverse proxy decides that the target language should be served via Translation proxy. It relays the request to the Google Cloud, where it is resolved to what we call the **temporary serving domain**, defined as `de-de-gereblye.app.proxytranslation.com`. In the serving subdomain mode, this domain is hidden from the user by the DNS settings added. In subdirectory publishing, the reverse proxy hides the temporary domain.

You can see that `de-de-gereblye.app.proxytranslation.com` will – same as with subdomain publishing

– relay the request and all necessary request headers to the origin server, which will respond accordingly with source language content that the Proxy then processes on the way back and sends long to the client in a translated form.

Setting aside the exact details of that cloud translation pipeline, that's about it.

Source language request

Requesting the original content is a relatively straightforward process that we should nevertheless describe in brief for the sake of completeness.

If a user in England requests `www.example.com/en/about`. The reverse proxy strips the `/en` prefix, decides on the language to be served and relays the request to the origin server.

In this case, there is no more proxy mediation (that is, no Translation Proxy) between the origin server and the requesting client, so the server response is returned and the user can peruse the webpage in the original language.

Note that on the origin server, the `/en` prefix does not exist as part of the directory structure - it is a virtual prefix used by the reverse proxy to dispatch to different domains based on the target language.

If the origin server is capable of providing content in more than one target language, the reverse proxy should presumably do the same thing for each of those target languages. If a request for `www.example.com/de/about` can be fulfilled by the origin server alone, the reverse proxy will relay that request straight to the origin server (where it is assumed that the server backend will make the decision based on the HTTP request headers received).

STEP 4 - Summary

After configuring your publishing mode, in this last step, you will see a short summary. By clicking on the **PUBLISH** button, you need to accept all of the options in the *Disclaimer* popup in order to be able to publish your translated website.

If the publishing is successful, then the serving domain name will appear in the Live domain column.

Deactivate / Delete

There are two different option for disabling the translation, and based on the selected publishing mode, you can choose either **Deactivate** or **Delete**.

- **Deactivate [Selected publishing mode: Serving domain]:** In this case, the related entities from our database will be removed, and it would make an error page appear when the translations are requested, which means that the publishing is disabled by our side. When you select this option, the serving domain name will be visible under the column called *Verified domain*. Alternatively, if you or your client removes the DNS records relating to the projects (especially the `ghs.domainverify.net` or `ghs.googlehosted.com` records), the requests will no longer reach the proxy, and translations will cease to be served.
- **Delete [Selected publishing mode: Subdirectory publishing]:** Removing the target locale will disable the translated website and make all the translated segments disappear. However, translations are retained in case of deleted languages as well, which means that the language can be re-added and the translations will be immediately available.

Path specific settings

Roots (path prefix manipulation)

We gathered below a few examples for you, in order to show you the behavior of these configurations.

Publish translated content on www.example.com at directory depth 1

If your publishing mode is subdirectory publishing, then this value will change automatically from 0 to 1.

Translated path prefix (removed from original path, added to translated) /en/

Removed from original path means that the original / remote server will never receive this path prefix, it will be visible only in the client translated URL, as you can see below:

The screenshot shows the 'PATH SPECIFIC SETTINGS' tab of the 'PUBLISH WEBSITE WIZARD'. Under 'Roots (path prefix manipulation)', the 'Publish translated content on www.example.com at directory depth 1' is set. The 'Translated path prefix (removed from original path, added to translated)' is '/ca/'. The 'Original path prefix (added to original path, removed from translated)' is empty. A 'Path mapping tester' sidebar is open, showing 'Address on translated site' as 'http://www.example.com/ca/fr-fr/' and 'Address on original site' as 'http://www.example.com/fr-fr/'. Below the tester, the text 'Translated' is visible.

path prefix manipulation

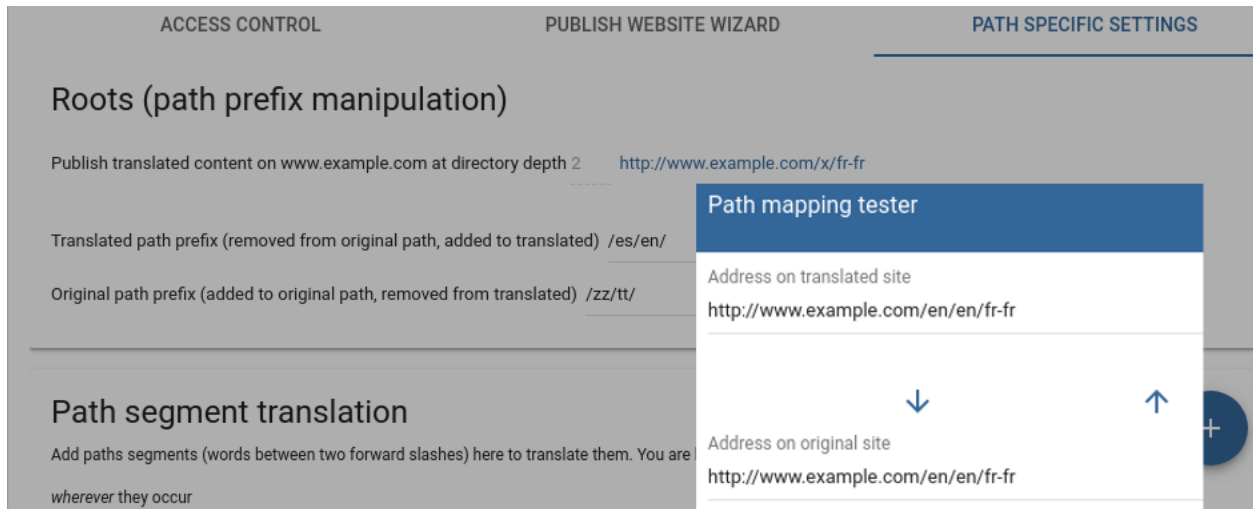
However, be aware that the directory depth number will increase according to the number of path prefix you describe in this field:

The screenshot shows the 'PATH SPECIFIC SETTINGS' tab with 'Publish translated content on www.example.com at directory depth 3' set. The 'Translated path prefix (removed from original path, added to translated)' is '/ca/en/es/'. The 'Original path prefix (added to original path, removed from translated)' is empty. The 'Path mapping tester' sidebar shows 'Address on translated site' as 'http://www.example.com/ca/en/es/fr-fr/' and 'Address on original site' as 'http://www.example.com/fr-fr/'. Below the tester, the text 'Translated' is visible.

path prefix manipulation with more than one directory

Original path prefix (added to original path, removed from translated) /en/

Using together Translated and Original path prefix options, they can be used to swap the root element of the link:



How

you can swap the root element

Path segment translation

With this option, you will be able to add paths segments (words between two forward slashes) here to translate them. You are limited to 50 entries at any time, but these will be translated *wherever* they occur, for example:

```
www.example.com/example --> www.example.com/translation
www.example.com/notexample/nottranslated/example/text --> www.example.com/notexample/
↳nottranslated/translation/text
```

3.8.3 Subdirectory publishing via nginx

As described on [their website](#),

nginx [engine x] is an HTTP and **reverse proxy** server, a mail proxy server, and a generic TCP/UDP proxy server

This makes it perfect for our purposes. We'll use a small subset of the features it provides to set up subdirectory publishing.

To get started, it's recommended to copy the settings provided in the Publishing wizard. These are partially project-specific. We'll use a project with the following details:

- Source domain: example.com
- Project code: redacted
- Translations exist for German
- We wish to publish them to example.com/de/
- Your white label is app.proxytranslation.com

With these options set in the Publish wizard, we get the following configuration:

```
location ~* ^/(de) {
    resolver 8.8.8.8;
```

(continues on next page)

(continued from previous page)

```

set $xhost de-de-redacted.app.proxytranslation.com;

proxy_set_header X-TranslationProxy-Cache-Info    disable;
proxy_set_header Server $xhost;
proxy_ssl_name $xhost;
proxy_set_header X-TranslationProxy-EnableDeepRoot true;
proxy_set_header X-TranslationProxy-AllowRobots true;
proxy_set_header X-TranslationProxy-ServingDomain $host;
proxy_set_header Host $xhost;

#old nginx
# proxy_pass $scheme://$xhost;
#new nginx:
proxy_pass $scheme://ghs.googlehosted.com;
}

```

With these details on hand, we can get started. Note that this guide assumes that nginx is already installed on your Debian-based server and that you are familiar with its command line. Canonical has a well-written guide for Ubuntu servers available [here](#).

Log into your server either as `root` or a user who can use `sudo`. If you aren't `root`, switch to that user via

```
$ sudo su
```

WARNING: At this point, you have complete, *unrestricted* access to the server. It's very easy to break it. Don't run commands that you aren't certain will do what you need to.

By default, the configuration file can be found at `/etc/nginx/nginx.conf`. We'll need to edit this file with our text editor of choice:

```
# nano /etc/nginx/nginx.conf
```

By default, the file looks something like this:

```

user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    # server_tokens off;

```

(continues on next page)

(continued from previous page)

```

# server_names_hash_bucket_size 64;
# server_name_in_redirect off;

include /etc/nginx/mime.types;
default_type application/octet-stream;

##
# SSL Settings
##

ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # Dropping SSLv3, ref: POODLE
ssl_prefer_server_ciphers on;

##
# Logging Settings
##

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

##
# Gzip Settings
##

gzip on;

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/
→xml application/xml application/xml+rss text/javascript;

##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}

#mail {
#     # See sample authentication script at:
#     # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#     # auth_http localhost/auth.php;
#     # pop3_capabilities "TOP" "USER";
#     # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#     server {
#         listen     localhost:110;
#         protocol   pop3;
#         proxy      on;
#     }
#
#

```

(continues on next page)

(continued from previous page)

```
#         server {
#             listen      localhost:143;
#             protocol    imap;
#             proxy        on;
#         }
# }
```

If you use this server to host websites too, not just as a reverse proxy then it will probably contain considerably more settings. In any case, scroll to the bottom and paste the configuration snippet that we provided. Save and exit.

NOTE: You can add further target languages by putting their code right below.

The configuration is in place, but nginx doesn't automatically switch them over. This ensures that you don't bring down your entire site by accidentally saving a syntax error. Verify that the file is correct.

```
# nginx -t
```

If no errors are printed, you should verify that your server can access the translation proxy. You can do so, for example, via `curl` and `head`. Don't forget to replace `de-de-redacted.app.proxytranslation.com` with your temporary domain.

```
# curl -Is https://de-de-redacted.app.proxytranslation.com/ | head -1
```

This should print that the status code is 200. If it doesn't, a likely explanation is that your server is missing Certificate Authority certificates and as a result can't connect to our service via HTTPS. Installing such certificates is beyond the scope of this article, but Tech Republic has a comprehensible tutorial on the topic [here](#).

Finally, you can reload the configuration without dropping connections via

```
# systemctl reload nginx
```

This ensures no downtime. With that, the nginx configuration is complete.

These instructions are up-to-date as of 24/02/2022.

3.8.4 Subdirectory publishing under AWS CloudFront

When you set up subdirectory publishing via CloudFront, you don't just set up a reverse proxy. CloudFront isn't just a simple web server solution after all. Instead of that, following this guide you'll set up a complete CDN in front of both your original and translated website. As such, you can also benefit from the other advantages that a CDN provides such as load balancing and global caching.

To get started, it's recommended to copy the settings provided in the Publishing wizard. These are partially project-specific. We'll use a project with the following details:

- Source domain: `example.com`
- Project code: `redacted`
- Translations exist for German
- We wish to publish them to `example.com/de/`
- Your white label is `app.proxytranslation.com`

With these options set in the Publish wizard, we get the following configuration:

```
Alternate Domain Names (CNAMEs): example.com
Origin Domain Name: de-de-redacted.app.proxytranslation.com
Behavior | path pattern: /de/*

Origin custom headers:
X-TranslationProxy-Cache-Info disable
X-TranslationProxy-EnableDeepRoot true
X-TranslationProxy-AllowRobots true
X-TranslationProxy-ServingDomain example.com

Cache Based on Selected Request Headers: Whitelist
- CloudFront-Forwarded-Proto
- CloudFront-Viewer-Country
- Origin
- Referer

Object Caching: Use Origin Cache Headers
Query String Forwarding and Caching: Forward all, cache based on all
```

With these details on hand, you can follow the following steps:

1. Log in to your CloudFront dashboard and head to *Distributions*. Or just use [this link](#).
2. Click *Create distribution*. You'll be navigated to the distribution creation form that you must fill in.
 - Origin domain: Use the one provided by the Publish wizard. In our case it's `de-de-redacted.app.proxytranslation.com`.
 - Protocol: HTTPS only is recommended, but Match viewer can work too.
 - Origin path: leave empty.
 - Name: enter a memorable name. We recommend `translationproxy-language`.
 - Add custom header: add the headers provided by the Publish wizard under *Origin custom headers*
3. Scroll further down to *Cache key and origin requests* and set it to *Legacy cache settings* and use the following settings:
 - Under *Headers* select *Include the following headers*
 - Select the headers provided by the Publish wizard under *Cache Based on Selected Request Headers*
 - Ensure *Object caching* is set to *Use origin cache headers*.
4. In the *Settings* section, just a bit further down, add the *Alternative domain name (CNAME)* as specified by the Publish wizard. In our case, it's `example.com`. Note that an SSL certificate will also be necessary for serving via HTTPS.
5. Click *Create distribution*
6. Navigate back to the *Distributions* section and select the newly created Distribution from the list. You'll be navigated to the details page of the Distribution that has multiple tabs.
7. On the *General* tab, ensure that an SSL certificate is available. If not, create and upload one.
8. On *Origins*,
 - Ensure that the `translationproxy-language` origin we created above is present
 - Click *Create origin* and enter the following:
 - Original domain: the source site, in this example it should be `example.com`.

- Protocol: HTTPS only is recommended, but Match viewer can work too.
- Origin path: leave empty.
- Name: enter a memorable name. We recommend the domain of the original site `example.com`.
- Click *Create origin* to save.
- **NOTE:** If you later wish to add additional languages, you don't need to create a completely different distribution, just add them here and follow the note under the following step too.

9. On Behaviors,

- Click *Create behavior* and use the following details
 - Path pattern: as specified by the Publish wizard - `/de/ *`
 - Origin and origin groups: select the `translationproxy-language` origin added previously
 - Click *Create behavior* to save
- **NOTE:** To add further languages, do this again with the appropriate Path patterns and Origins
- At this point, you should have behaviors set for every target language, but we'll also need one for the case when the visitor wants to see the source language site. The process to follow depends on the structure of the site.
 - If the original content is under the root, meaning that it is `example.com/path/to/page`, do the following steps:
 - * Click *Create behavior* again and use the following details:
 - Path pattern: enter the path you wish to move it to, like `/en/ *`.
 - Origin and origin groups: select the `example.com` origin added previously
 - * Click *Create behavior* to save
 - If the original content is already under a language folder the steps are slightly different:
 - * In the list of behaviors, there is one called the *Default (*)*. Find it and select it.
 - * The *Edit* button becomes available. Click it.
 - * At *Origin and origin groups*, select the `example.com` origin added previously
 - * Scroll down to the bottom and click *Save changes*

10. With that, the CloudFront configuration is complete.

These instructions are up-to-date as of 24/02/2022.

3.8.5 Subdirectory publishing via CloudFlare Workers

Cloudflare Workers is a serverless application platform. As it can work with HTTP requests, it can be used instead of a complex reverse proxy setup for subdirectory publishing of the translated website.

To get started, it's recommended to copy the settings provided in the Publishing wizard. These are partially project-specific. We'll use a project with the following details:

- Source domain: `example.com`
- Project code: `redacted`
- Translations exist for German
- We wish to publish them to `example.com/de/`

- Your white label is app.proxytranslation.com

With these options set in the Publish wizard, we get the following configuration:

```
addEventListener('fetch', event => {
  event.respondWith(handleRequest(event.request))
})

/**
 * Fetch and redirect or continue to Proxy
 * @param {Request} request
 */
async function handleRequest(request) {
  const redirectPaths = ['de']
  const proxyUrls = {
    'de': 'de-de-redacted.app.proxytranslation.com'
  };

  const url = new URL(request.url);
  const { pathname } = url;
  const [, topLevelDirectory] = pathname.split('/');
  if (topLevelDirectory && redirectPaths.includes(topLevelDirectory)) {
    // The request's URL must be overwritten to send it to the Proxy
    request = new Request(
      `${url.protocol}://${proxyUrls[topLevelDirectory]}${pathname}${url.search}`
    )
    request.headers.set('X-TranslationProxy-AllowRobots', 'true')
    request.headers.set('X-TranslationProxy-Cache-Info', 'disable')
    request.headers.set('X-TranslationProxy-EnableDeepRoot', 'true')
    request.headers.set('X-TranslationProxy-ServingDomain', 'example.com')
    request.headers.set('Host', proxyUrls[topLevelDirectory])
    return fetch(request);
  } else {
    // Regular request. Forward to origin server.
    return fetch(request);
  }
}
```

With these details on hand, you can follow the following steps:

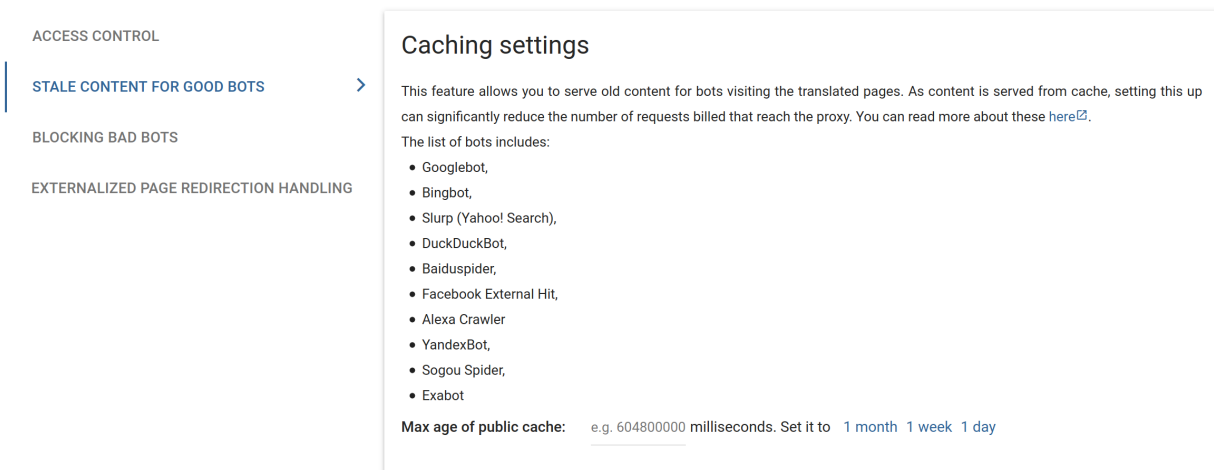
1. Log in to your CloudFlare dashboard and head to the *Workers* section.
 - If you've not used CloudFlare Workers before, you must choose a subdomain and a plan. For the purposes of this guide, we'll assume that the domain chosen is translationproxy-worker.
2. Click *Create a Service*
3. Fill in the *Service name* field with an appropriate name. translationproxy-domainname would be suitable in general so in our case, we'll just use that: translationproxy-example.
4. You don't need to change the starter, we'll overwrite it anyway, so you can simply click *Create service*. You'll be navigated to the settings section of the newly created Worker.
5. Click on *Quick edit*. You'll be navigated to the code editor.
6. Delete all the example code that CloudFlare provides and add the snippet we provide.
 - **NOTE:** If you wish to add more target languages, change the `redirectPaths` and `proxyUrls` variables accordingly.
7. Click *Save and Deploy*

8. Test it by opening the URL of the Worker on the translated path. In our case, that would be `https://translationproxy-worker.translationproxy-worker.workers.dev/de/`. It should load the translated site.
9. Head back to the settings section of the Worker and click on *Triggers*
10. Click on *Add route*
11. Enter your domain. In our example it is `*.example.com/*`.
 - **NOTE:** Only routes of hostnames configured on Cloudflare can be specified, so if your domain isn't yet configured, you must do so via the *Add site* button in the top bar. Adding further DNS records is likely required to do so.
12. Your environment will now respond to requests to `*.example.com/*`. You can verify that the translated site loads at, in our case, `https://example.com/de/`
13. With that, the CloudFlare Worker configuration is complete.

Make sure that [Rocket Loader](#) is disabled when using CloudFlare!

These instructions are up-to-date as of 04/03/2022.

3.8.6 Global settings



Global

settings screen

This menu contains publishing settings that apply to every target language at the same time.

Access control

This option is also present on a per-language basis but here it affects all languages. This can save you some work if you'd add the same settings to all of the languages anyways. Add username/password combinations and tick the modes that you wish to protect using [Basic authentication](#). The settings you add here and per-language are merged and both are applied.

Stale content for good bots

Bot traffic can significantly increase the maintenance costs of proxy sites. Some of this bot traffic is useful, e.g. the search engines. With this feature, you can serve a cached response to these bots and thus reducing the number of page requests they generate.

Note that with this feature enabled, bots won't be notified about changes to the site until the caches expire so their indices may be outdated for some time.

Blocking bad bots

There are bots, however, that aren't useful to the site owners at all. These can be blocked completely based on their User-Agent. Enter a Java regular expression to specify the agents that you wish to block. These will receive a 403: Forbidden response when trying to visit proxied pages.

Important: Do not enter 2 pipe symbols (|) one after the other. It will block every user and bring the whole translated site down. Consider the following regular expression `[Mm]alicious[Bb]ot|[Ee]vil[Ss]pider`. This matches "MaliciousBot", "" (the empty string) and "EvilSpider". As every user agent contains the empty string every user will be flagged as a bad bot and will be blocked.

Externalized page redirection handling

Using this feature, you can avoid duplicate content (and the SEO penalty that comes with it). Externalized pages aren't translated and by default shown as the source language. You can change this behaviour to redirection (either via 301: Moved Permanently or, 302: Moved Temporarily) or disable serving these pages (401: Access Denied or 404: Not Found)

3.8.7 Language Selector

As translation of a website is nearing completion, it is usually necessary to prepare a language selector to make the various languages more accessible. There are two options:

1. develop a custom selector in-house
2. use one of the proxy default selectors.

In all cases, the addition of a language selector will need to be done on the original server and then propagated to the translated domains through the proxy.

Proxy Solutions

To spare you the expense and time of developing a custom language selector, we have prepared two default solutions you can use. Each of these require the minimum possible intervention on the original site for deployment. Note, however, that seamless visual integration of the selector requires some fiddling with the CSS rules. The following two types of language selectors are available:

Sidebar

The sidebar is an easy-to-deploy-and-use language selector (with limited customization options) that doesn't require any changes on the original site beyond the addition of a single `<script>` tag to the source code. The language selector is a hovering sidebar which displays the flags of the selected target languages.

The sidebar is an easy-to-deploy language selector that displays flags for each specified target locale on the left or right side of the website. The horizontal orientation and display names are settable.

Set-up steps

On the Dashboard 2.0:

1. Click on the Language selector menu

2. Select the languages that you want to display in the selector
3. Select the “Sidebar” tab (default)
4. Select “Left Side” or “Right Side” according to your preference
5. Click save and show embed code

Note: You may also declare custom language names to be displayed by using the dropdown selector or choose “Language” to display only the language, but not the Country.

On the original site:

1. Finally: copy the embed code from the textbox and add it to the end of `<head>` tag on the original site. With the tag added to the original site, a reload should reveal the language selector.

Dropdown

The Dropdown is the more customizable option. One bit of additional adjustment in the original source is necessary, however: besides adding the `<script>` tag, a dedicated `<div>` tag is also necessary – since the dropdown is more closely integrated with the site structure (that is, it has to be positioned in relation to other elements), this `<div>` tag will be used to locate the designated location of the dropdown.

Setup steps:

On the Dashboard:

1. Click on the Language selector menu
2. Select the languages that you want to display in the selector
3. Select the “Dropdown” tab
4. Declare a unique class name for the `<div>` that will contain the language selector. The default is `el-dropdown` but you may specify any class name that does not clash with any other classnames in the original HTML. For the purposes of this tutorial, the classname will be `el-dropdown`
5. Select a display template. You may choose between “Language (Country)”, “Language” or entirely custom names according to your preference.
6. Click save and show embed code and copy the code.

On the original site:

1. Locate the position in the markup that you wish to insert the Dropdown
2. Extend the HTML markup with a `<div>` tag. Furnish it with the previously specified class. In this case, it would look something like this: `<div class="el-dropdown"></div>`
3. In addition, Add the embed code copied from the Dashboard to the `<head>` tag of your page.

Post-setup:

At this point, the language selector is displayed after reloading the page in your browser (the JavaScript referenced in the `<script>` should take care of downloading and rendering it). The language selector contains a menu button, a container and target language entries you can customize via CSS rules. The language selector is self-contained for convenience, so it uses in-line styling. Postfix your CSS rules with “!important” to make sure that they are applied.

RYO or Existing Selector

It is possible that site undergoing translation is also running (or was previously running) a different localization solution, in which case a language selector is usually already present. It is relatively straightforward to extend an existing language selector with the proxy-published target languages.

Since that language selector is also proxied, it is important to protect the link pointing to the original language site from being remapped in the process.

In order to do this, the `__ptNoRemap` special class can be used to indicate to the proxy that it should not change any links in the given tag.

A brief example of how the original site link can be protected from remapping with `__ptNoRemap`:

```
<div class="selector-container">
  <ul id="selector">
    <!-- PREVENT REMAPPING OF ORIGINAL LINK - this requires a full domain-->
    <li id="en-US" class="language selected"><a href="//www.example.com" class="__
    ↪ptNoRemap">English</a></li>
    <ul class="dropdown">
      <li id="de-DE" class="language"><a href="//de.example.com">Deutsch</a></
    ↪li>
      <li id="fr-FR" class="language"><a href="//fr.example.com">Francais</a></
    ↪li>
    </ul>
  </ul>
</div>
```

You may also wish to use the `translate="no"` attribute in places where you want the proxy to ignore the textual content of the HTML element.

Example

Depending on how involved it is, a language selector may have to react to user input/animate dropdowns/start XHR requests, etc. Its appearance and behavior depend on the number of languages, publishing method, site layout & design etc, which is beyond the scope of this manual. But to get you started with a suggestion, a commented minimal example in vanilla JS/CSS follows below, which is also available in a slightly modified form as a JSFiddle [here](#).

```
(function () {
  "use strict";
  /* The target locale code, a two-letter language code plus a two
     letter country code, separated by a dash, is added
     automatically over the proxy. Either this value or the HTML
     markup might have to be adjusted to be applicable on the
     original site */
  var lang = document.querySelector("html").getAttribute("lang") || "en-US";
  /* The usual "readystatechange" event listener works if
     you need to wait

     document.addEventListener("readystatechange", function () {
       if (document.readyState === "complete") init();
     })
  */

  init();
})
```

(continues on next page)

(continued from previous page)

```

function init () {
    selectDefault();
    setEventHandlers();
}

function setEventHandlers () {
    var items = document.querySelectorAll("div.selector-container li.language a");
    for (var i = 0; i < items.length; i++) {
        items[i].addEventListener("click", function (e) {
            selectLanguage(e.target.parentNode);
        })
    }
}

function selectDefault () {
    /* See if we can select anything in the language selector
       based on our locale code. */
    var base = document.querySelector("div.selector-container li#" + lang);
    /* ...and try to default to anything we've found */
    if (base !== "undefined") selectLanguage(base);
}

function selectLanguage (target) {
    /* 'click' event was triggered on any of the language selector
       entries. */
    var selected = document.querySelector("li.selected");
    var dropdown = document.querySelector(".dropdown");

    /* don't do anything if we would be switching to the same
       language */
    if (target === selected) return;

    /* SWITCH LANGUAGE.

       ...but we are only "swapping elements" in the menu (the
       code doesn't navigate in actuality). A target language
       domain or a CST query parameter could result in a page
       load.

       The "dropdown" approach showcased here is also an
       illustration of a frequent issue: when selectLanguage() is
       called, the selected target language is replacing the
       current one in the dropdown list. So, if you are particular
       about the target languages being listed in a set order, a
       hide/show approach might work better than DOM element
       swapping. */
    target.setAttribute("class", "language selected");
    selected.setAttribute("class", "language");

    selected.parentNode.replaceChild(target, selected);
    dropdown.insertBefore(selected, dropdown.firstChild);
}
})();

```

Below are the associated CSS rules.

```
div.selector-container ul {
  box-shadow: 2px 2px 8px rgba(0, 0, 0, 0.5) !important;
  list-style: none;
  margin: 0;
  padding: 0;
}

ul#selector {
  list-style-type: none;
  text-align: center;
  background-color: #f5f5f5;
  max-width: 100px;
}

ul#selector li {
  list-style-type: none;
  list-style: none;
  padding-top: 10px;
  padding-bottom: 10px;
}

ul#selector a {
  color: black;
  text-decoration: none;
}

ul#selector li.selected:hover a {
  color: grey;
}

ul#selector li.selected:hover + ul.dropdown {
  display: block;
}

ul#selector ul.dropdown:hover {
  display: block;
}

ul.dropdown {
  display: none;
}

ul.dropdown li.language {
  padding-top: 10px;
  padding-bottom: 10px;
}

ul.dropdown li.language:hover {
  background-color: chartreuse;
}
```

RYO selector + Client-Side Translation

If a project is published using Client-side translation, either an existing language selector needs to be used, or a new one developed to trigger translations after selection.

CST translation is activated when the target locale code is stored in the browser with the help of the `__ptLanguage`

query parameter.

In order to trigger language selection/translation in-browser, add this query parameter with the appropriate published locale codes as values. Note that in order for the method to work, URL navigation has to happen once, which means a full reload is required in order to change languages.

Let's modify the example above to suit the needs of CST:

```
<ul id="languageSelector">
  <li class="language selected">English</li>
  <ul>
    <li><a href="/?__ptLanguage=en-GB">English</a></li>
    <li><a href="/?__ptLanguage=de-DE">Deutsch</a></li>
    <li><a href="/?__ptLanguage=fr-FR">Francais</a></li>
  </ul>
</ul>
```

The value of `__ptLanguage` will persist across sessions, the JS file will rely on it until the user changes their mind about their preferred language.

Visiting `www.example.com/?__ptLanguage=ja-JP` will change target languages into Japanese and store the setting. Visiting `www.example.com` the next time around will result in the site being translated into Japanese automatically.

The default browser locale can be detected successfully to be a published target language, CST will attempt a locale-specific translation without having had a query parameter provided to it.

Although that is about it, it is useful to keep in mind a few things when setting up a language selector to work with CST.

Somewhat different from the proxy method, CST requires that we provide all locale codes explicitly, especially the **original** language. Since CST *stores* the user's selection in the browser's local storage, the query parameter is necessary in order to allow the user return to the original language version.


CST publishing does not require any proxy traffic, but the Workbench In-Context editing screen remains available. This can sometimes result in the Client-Side Translation getting mixed up with the proxy-based translations, so if you are using the proxy modes for your translation work while publishing with Client-Side translation, it is generally a good idea to ignore the language selector in Preview.

3.8.8 Link transformation preview

Link transformation preview

Using this tool, you can preview how the proxy will transform links in your website before actually going live or without interfering with the already-published website.

Select a language to import the current settings Select a language ▾

The current settings is same as the following target language  English (United Kingdom) [en-GB]

Domain

Provide the domain name on which the translated website is to be published.

Roots (path prefix manipulation)

Publish translated content on at directory depth

Translated path prefix (removed from original path, added to translated)

Original path prefix (added to original path, removed from translated)

Path segment translation (0)

Add paths segments (words between two forward slashes) here to translate them. You are limited to 50 entries at any time, but these will be translated +

wherever they occur

OPEN PATH MAPPING TESTER

[Link](#)

transformation preview screen

This feature allows you to preview the path settings that normally become available once you published a target language. You can use this to test URL translation, root depth or original and translation path prefixes before you publish.

3.8.9 Basic publishing: Client-side translation

Basic publishing is currently the recommended publishing method.

What is CST?

To summarize briefly, Client-side translation is a **publishing method** that lets you use your translations on the original domain.

As opposed to the proxy method, which involves standing between the user request and the original site, translating responses on-the-fly, Client-side translation does not require the proxy pipeline to serve content.

That is, instead of being processed in the Cloud, translation of the page will happen in the user's browser, using a JavaScript based dictionary file that can be exported from your project and embedded right in the original site's HTML source.

Pros & Cons

Basic publishing is suited for most websites be it small or large. It especially excels when it comes to websites and web applications mainly driven by JavaScript. As the translation engine runs in the browser, it can work on the DOM of the website as opposed to the proxy that can only work on the source code of the site. This means that Client-side translation works better out of the box. This is particularly useful in case of smaller websites where the engineering effort necessary can be almost zero, ensuring a fast turnaround.

The main drawback of Client-side translation is that the source content must be loaded before the translations can be replaced. In a nutshell, this means that the original text will be visible for a bit before the translations are put in place.

Publishing process

Client-side publishing consists of two steps:

Dashboard - Publish Your Translations

There are three things you must do on the Dashboard. First, head to the *Export* section. Under *File format*, select *Client-side translation (JS, export only)*. The rest of the options change to the correct values by default. You'll be notified via email when the process is finished. Once it arrives, go to *Previous exports*. Find the latest JavaScript export and in the action menu, select *Publish*. Finally, go to *Language selector*, select the target language(s) and click save.

Add stub.js reference to original site

Like embedding a video, including your published translations involves adding a single `<script>` tag to the original HTML source. The script tag looks like the following:

```
<script type="application/javascript" src="http://${PROXY_APP_DOMAIN}/client/$
↪{PROJECT_CODE}/0/stub.js"></script>
```

We have a dedicated integrators' guide that you can find [here](#).

Once the script is placed on the original site, a dictionary file of published translations will be downloaded by the site automatically and a language selector will be displayed. You can just select a new target language.

Published and possible

It's important to mention how the target languages in the selector are calculated. It uses what we call the *Published and Possible policy*. In order for a given target language to show up, it should be selected under *Language selector* on the Dashboard. This is what we refer to as *Published*. Additionally, you'll need a JavaScript export created and published, the *Possible* part of the mantra.

If at any point, a target language doesn't show up, it's likely due to one of these conditions missing. In this case, remember the *Published and Possible policy* or just reach out to our support team.

3.9 Third party integrations

3.9.1 Third party integrations

The proxy can integrate with quite a few third party services. The integration can simplify your translation workflow or the management of the source and target files.

These services can be configured either in the Dashboard 2.0 for the project or in the Account settings page. By default, the project level credentials are used with the account level as fallback. You can prevent the fallback using the “Disable fallback to account level credentials” checkbox.

Note that the cost of these services isn’t included in the proxy fee.

GeoFluent

[GeoFluent](#) is a machine translation engine that enables real-time translation of text (or audio). It can also detect the language of the source text. This can be especially useful when the source site is pre-localised and you need to ensure that you only pick up content in the source language.

To enable it, you’ll need to have your Account key and secret as well as the endpoint that the proxy must call.

Google Translate

I doubt [Google Translate](#) needs much of an introduction. Both the traditional dictionary-based and the new neural translation modes are supported as well as source language detection. You can read more in [Google’s Official Documentation](#)

To set it up, you must generate an API key in Google Cloud Platform ensuring that the “Accept requests from these server IP addresses” field is empty. Then, you can enter your key and enable Google Translate for your project.

SYSTRAN Translate

Aside from high-quality translations enabled by its industry-specific translation models [SYSTRAN translate](#) also supports the usual machine translator features such as source language detection.

It can be enabled by entering your API key. The default endpoint, `https://api-translate.systran.net/`, can also be overridden should you need to do so.

Microsoft Translator

Successor of the Bing Translation API, the [Microsoft Translator API](#) can be used for manual or automatic pre-translation, source language detection as well as suggestions on the Workbench.

To enable, simply enter your Subscription key.

Dropbox

The Proxy is capable of installing a Dropbox connector in your account, that allows it to export into and import from a specific folder in the account. The proxy root folder needs to be specified once, during account linking, after which the proxy will create new folders within based on the project code as needed.

During export, the option to submit to Dropbox will place the exported ZIP file into the `export` folder of the given project, allowing you to download and handle the file rapidly. After processing, the contents of this folder can be deleted freely.

The `import` folder is used to re-import translated content into the database. Once a file is placed in this folder, the project owner receives an email notification of the fact. From the Import dialog, they can then choose which files they want imported, which are then imported directly from Dropbox. By default, on switching to the Dropbox tab in the import dialog, only files added *after* the last import will be selected, making distinction easy.

To add a new account, click “Select Dropbox account”. You can then select an account that you added to any other project or you can log in with a valid Dropbox account and authorize the Translation Proxy Dropbox app to access your storage.

XTM

The proxy can export content in XLIFF format into XTM, creating projects as necessary, and listen for workflow completions to commence import. XTM has also integrated the preview functionality, allowing live verification of translations.

After configuring XTM access under the Account screen, submitting to XTM as an external system becomes available at export-time, or from the Previous Exports dialog. Upon submission, the proxy creates the relevant project within XTM and imports the XLIFF into the project. Afterwards, a listening service starts, periodically polling XTM for the workflow state of the project. When the project is set to completed, the proxy detects this during its next polling cycle, and imports the translated XLIFF back into the proxy project.

XTM has also integrated the proxy preview. Using information in the exported XLIFF, XTM’s online interface can display the translated page, allowing in-context verification of the translations.

Nota Bene: the live-updating preview does **not** mean that translations are streamed to the proxy for inclusion! You will still need to save the XLIFF file and import it (either manually or by completing the XTM project) to return the translations to the project!

XTRF

Similarly to Dropbox and XTM, the proxy can export content to XTRF. The workflow is very similar to XTM.

MemoQWeb

The proxy can also export content to a MemoQWeb instance and retrieve the translations. Additionally, when you open an XLIFF file from the proxy in MemoQ, you can see a live preview using the MemoQ plugin.

Nota Bene: the live-updating preview does **not** mean that translations are streamed to the proxy for inclusion! You will still need to save the XLIFF file and import it (either manually or through MemoQWeb) to return the translations to the project!

3.10 Workbench

3.10.1 Workbench

By clicking [here](#), you will be taken to the Workbench on a new tab. The Workbench is the working end of the proxy service - please visit the [Workbench section](#) of this manual for the details of how to use it.

4.1 Account settings

Here you can find the settings that are specific to your account as opposed to your projects. You can access these settings by clicking on your profile picture on the top right on the Dashboard 2.0 then selecting (as you probably guessed) Account settings.

4.1.1 Profile settings

You can edit the main details of your project as well as change your password.

- **Profile picture:** Click on the image to replace it with one that you upload. Note that it is visible to other users in the Sharing settings section so make sure to select something appropriate.
- **Nickname:** Similar to the Profile picture.
- **Show subscription details:** Allows you to hide your balance on the Dashboard 2.0.

4.1.2 Billing settings

Enter or change your billing details here. The fields are different based on whether you are an individual or a company so make sure to select the corresponding radio button before filling the details in.

Billing information

Legal status:

☐ Business ☒ Individual

Country*:

United Kingdom

Contact name*:

Proxy User

State/Province:

Glasgow

Send a copy of bills to:

alternative email

Address*:

13 George Square

City*:

Glasgow

Postal code*:

G2

information

Billing

4.1.3 Usage report

Wallet Report

By selecting the date interval and at least one project, you can export the wallet statistics of the selected projects, which will be generated by wallets. You can generate daily, and also add additional recipients. The following project list contains only the currently active projects. If you would like to generate a report, that contains all of the projects (active and deleted), please select the option Generate report for all projects (active and deleted)

▲ Exports are restricted to a 100 days. Move the end date of the report first to move the window. Please note that the start and end date are both inclusive.

From

To

Fri Jan 01 2021

Tue Jan 05 2021

☐ Generate report for all projects (active and deleted)

Select project(s)

<input type="checkbox"/>	Project code	Domain	Alias
<input type="checkbox"/>	baguette	example.com	

Additional recipients (optional)

example@example.com, example2@example.com

☐ Generate daily report (optional)

CREATE REPORT

report screen

Usage

126

Chapter 4. Account settings

Use this section to track your usage and expenses on multiple projects. Select a date range and at least one project. If you need information on the use of deleted projects, select “Generate report for all projects (active and deleted)”.

You can also add additional recipients and opt for a daily report. As well as the usual summary part (see below), the daily report contains the daily use of the projects (if there was any). This is particularly useful when trying to figure out why a project cost more than expected. You can find the date of the spend and what it was spent on.

Billing period	December 01, 2020 00:00:00 to January 05, 2021 23:59:59 (UTC)	
User id		
E-mail	proxy.user@example.com	
	Usage	Cost
Page views	4125	4.125
Cached page views	0	0
Unique source words added	725	1.45
Unique source words translated	10905	109.05
Total (without VAT)		114.63 EUR

Report

summary


4.2 Third party integrations

You can set up the same integrations that are available on a per-project basis here as well (detailed [here](#)). This has the benefit that you can access them from all of your projects.

The following integrations are available:

- GeoFluent
- Google Translate
- SYSTRAN Translate
- Microsoft Translator
- Dropbox
- XTM
- XTRF

4.3 Translation memory

Name	Source language	Target languages	Actions
Example for docs writing		   	



Translation

memory settings

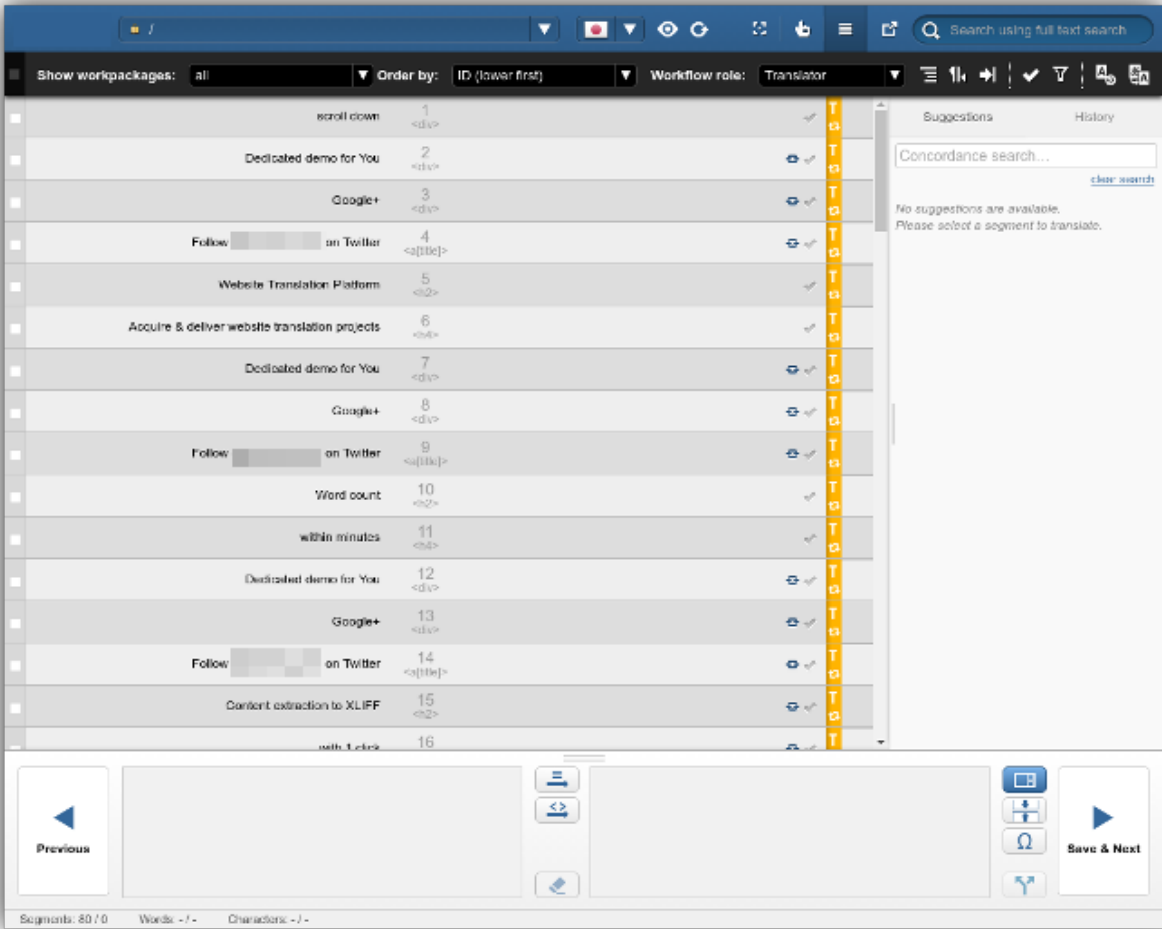
As mentioned in the [section about translation memories on the Dashboard 2.0](#), translation memories are linked to your account and *not* your project. This way, if you want to, you can have one gigantic translation memory for all of your projects. This is useful if you have multiple projects of the same topic. This page allows you to view and manage all of the memories associated with your account.

You can create new memories as well as change the name and target languages of your existing memories. Then you can use them by assigning them to the projects that you need them on.

CHAPTER 5

The Workbench

5.1 Introduction



The Workbench is a CAT tool in the cloud.

If the Dashboard is our take on the project management side of website localization, then the Workbench is the analogous CAT tool you can use without ever leaving the browser window.

In this chapter, we take a close look at all the various features available on the Workbench. The explanations pertaining to the various elements and functionality of the Workbench screen are grouped into thematically linked subsections. Use the sidebar to navigate to the subsection of your choice.

5.1.1 Opening the Workbench

You can access the Workbench anytime after adding at least one target language.

There are a few ways to enter it:

1. In the Dashboard menu, see the *Languages* part. If you hover over any of the target languages, a menu bar will be displayed to the right. Click on “Translate in List View”..
2. Click on “Manage segments” in Content menu...
3. Enter the Page list and hover over any entry to display a menu to the right. Click on “Translate in List View”...

.. and the Workbench will open in a new tab.

5.2 Moving Around

The Workbench has a single viewport, so every feature you’d need to navigate between pages and segments is always just a couple of clicks away.

You may switch between pages, search for text, filter for segments based on a variety of metadata, such as Approval state, containing block element, translation source and so forth. This section deals with the various ways to do that.

5.2.1 Workbench Page List

A segment is tied to the specific page it was found on. You can use the Page list dropdown right next to the logo to get a list of all pages currently within the scope of translation.



Page

list

You can click on any page entry in the list to visit that page and get an overview of all segments associated with that page. Websites can get very large with a huge list of pages - use the search field embedded in this dropdown to locate a specific page (you may use regular expressions with the usual format of including them between two slashes like this: `/[regular expression]/`).

There are three options at the top of the dropdown that bear special mention:

Show All Entries

Most of the time, you will want segments displayed for a specific page, but you may also use this option to get an overview of all segments across all pages.

WARNING! Only **List View** is available in this view, all other View buttons will be unavailable!

The All Entries list doesn't flood your browser with every last segment all at once: segments will be loaded in batches of 500. The Workbench will automatically fetch a new batch of entries as you scroll down.

Show Pending Entries

By default, Scans will pick up new entries in "Approved" state, which in this case means "Approved for Translation", immediately available for translation. You can change this default behavior, by going into the Dashboard and changing it in [Advanced settings](#) to either Pending or Excluded.

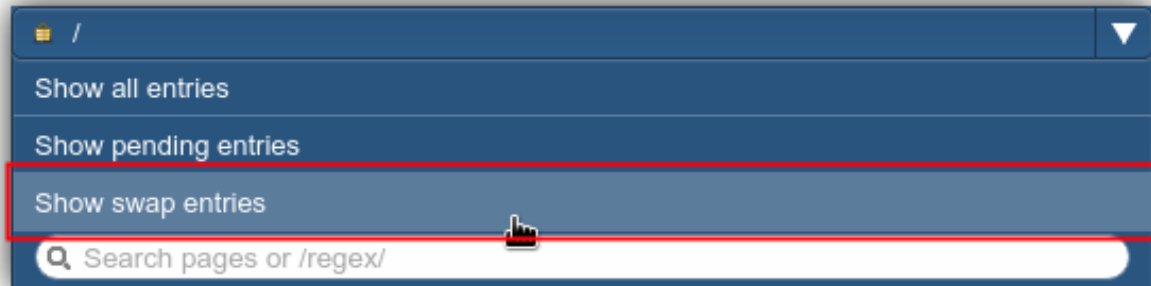
By clicking on "Show pending entries" in the page list, the list of all entries that are currently waiting for Approval is displayed. In their current state, They will not be included into exports unless the relevant option is selected at export-time, and will not appear for translation unless filtered for specifically.

Project or backup owners, or users with the Customer role can move these into either one of the other two states, by **approving** them or **excluding** them entirely from the scope of translation.

Show Swap Entries

Swap entries are those segments that have had the “EL_swap” class added to their enclosing tags on the source site.

They are special in that they are added to the Workbench without processing their tags. They will be displayed verbatim, allowing you to edit the source content markup directly. That content will be sent as-is by the proxy for each request.



Swap

entries

Take caution when editing swap entries. All responsibility of rendering them successfully *and* safely is delegated to the requesting browser.

Filters

There is a comprehensive assortment of filters available. Click on the Filters icon in the toolbar to get a an overview of all available filters:



Filters Icon

Use the checkboxes to define your filtering settings in the dialog and click on “Set Filters”.

You can filter for the roles that edited segments, their status and the date range they were last modified. Both the before and after dates are optional. If “after” date is omitted, it means the filter should display segments that were modified until the “before” date. If “before” omitted, it means the filter should display segments that were modified after the “after” date. Providing both will set a filter that displays segments in a given date range where both after and before dates are inclusive. In general, everything that modifies the translation of an entry (translation, pre-translation etc.) is considered to be a modification. There are a couple of exceptions that aren’t currently tracked:

- a segment is confirmed or its workflow status changes
- a segment is locked



Filters

Dialog

The dialog will close and a new element will appear in the toolbar, indicating that user-defined Filtering settings are currently active, and the segment list is updated accordingly. Click on the “X” to disable filtering. You may also click anywhere else on the toolbar indicator to open the Filters dialog again and fine-tune your settings.

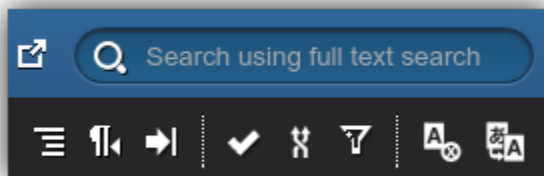


Filtering

Indicator

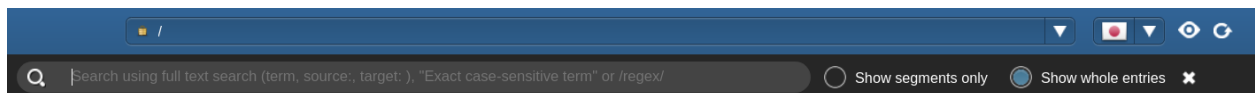
These filters work with the various types of metadata associated with an Entry (such as currently assigned workflow role, enclosing block element in source, approval state or source of translation), **not** content - use the **Search** functionality to filter for source or target text.

5.2.2 Searching for segments



Search field in corner

Use the full-text Search field in the upper-right hand corner of the screen to search for segments. When you click on the Search field, the Workflow part of the black menu bar will be superimposed with a search field right above the Viewport.



Search

field

Normal full-text search and regular expressions are supported. The normal search can be a bit misleading, as it is a forward-from-word-boundary search. If, for example, you'd like to find "translation" in a source segment, a search query for "translatio" will turn up all segments containing any words that start with that exact set of strings (and likely end with ns, or nese, perhaps). Searching for "ranslation" only, however, will return zero matches.

Use regular expressions to search between word boundaries: to extend the previous example, a query for /.ranslation/ will show all segments that contain any character (except newline) exactly once, followed by the literal string ranslation. The same as with the Page list, a string is interpreted as a regular expression if you enclose it between slashes.

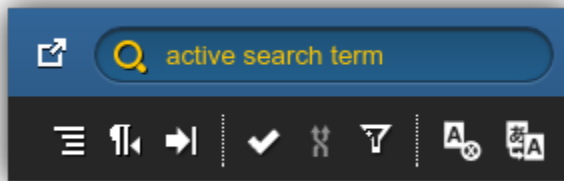
You may select between displaying segments or whole entries using the radio buttons next to the Search field.

Closing the Search Field

It is important to remember that the search field is also a **filter**: as long as it is active, segments will be filtered based on its contents regardless of being in All entries view or on a specific page.

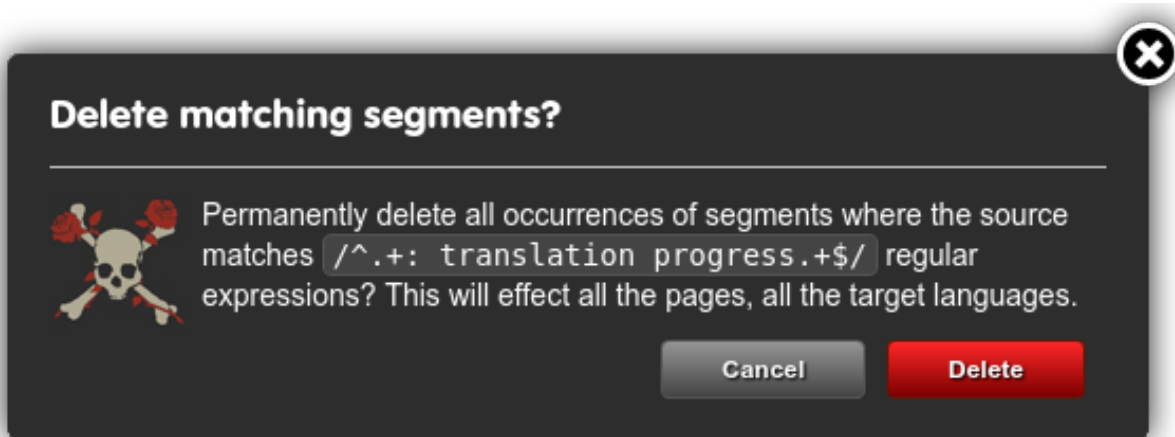
If you wish to restore full view of segments, clear the search field and send an empty search. Use the close button next to the Search display options to close the search field.

If a search string is still present, it will be preserved and displayed in the upper right corner Search field in orange, like this:



Search with string

Deletion of Segments



Delete

segments

You might be wondering what this has to do with navigation, but executing a regex search reveals another feature of the Workbench you might ordinarily look for someplace else: that of deleting segments. If you click on the Magnifying glass icon while the regex search is active (a bottom-facing triangle will indicate availability), the dialog above will open.

The non-discoverability of this feature is premeditated. Skulls & Bones warnings would generally apply to **any** situation where the words “delete” and “regex” are found in the same sentence. That being said, to give you a measure of peace of mind, deletion of segments is not as final as we seem to make it out to be: when TM Freeze is **disabled**, you can re-add segments anytime by Scanning or visiting in Preview the page that contains the deleted segments.

But that is new words added each time. So, if for no other reason, be careful about deleting segments in order to avoid unnecessary expenses. Buyer, beware!



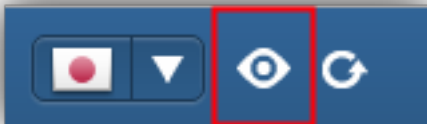
Mallard on Deletion

The

5.2.3 Preview

By clicking on the “Eye” icon, you can visit the temporary domain to check your translations in their original context. In the “All Entries” view, this function is disabled if no segment is selected - without picking a segment, the Workbench has no information on which page to show you. Otherwise, if a Page view is open, the selected page will be loaded.

If you select a segment in All Entries view, the Preview proxy will open on the page where it was seen by the crawler for the first time.



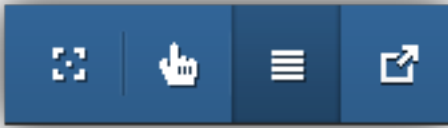
Preview Icon

The icon on the Workbench will take you to the Preview mode, but there are a few different Proxy modes available besides that. See [here](#) for details.

There is more than one way of looking at content on the Workbench, and the default, the List View is only one of them. In this section, we go over the various in-context ‘Views’ you can access from the Workbench and use to edit your translation while making sure that it is behaving exactly as it should in the original context.

5.3 Views

5.3.1 List View



List View Icon

While not necessarily the most impressive, the List View is certainly one of the most useful views on the Workbench. You can use it to go over each segment being translated and edit, filter and search for any subset.

The List view provides various features to use with each entry. The currently selected Entry will be highlighted in yellow, hovering will causes the Entry under the cursor to highlight in blue. The presentation of Entries is clear and simple, yet there are a variety of features you can use with each. Let's take a detailed look at an entry and see what each part of a line does.

Anatomy of an Entry



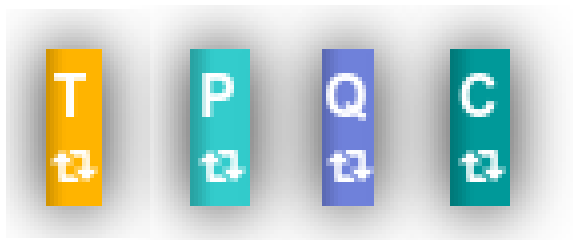
Segment

- Select Checkbox:** check this box to select the entry or segment in question. You may use the “Bulk Actions” icon to batch process selected entries (i.e. confirm, exclude or approve them)
- Source Entry:** contains the text that was found within a given block element on the source site. You may set the text direction using the “Align source text” icon in the toolbar. Otherwise, the Source Entry is not editable.
- Entry No. + Containing Block Element:** use the “Go to Segment” icon to jump to a segment with a given number. This number is assigned to entries in order of arrival (new entries can be found at the bottom of the list). Additionally, the tag that contained the text on the original site is displayed below it (useful when you want to identify a segment in the HTML source).
- Target entry:** The translation, provided by a variety of sources: Manual Editing, Translation Imports, Machine Translation or Translation Memory.
- Lock Segment:** Prevent any changes from influencing the current content of the Target Entry. Especially useful when you want to run batch processes on your segments, but you want to exclude an entry from the scope.
- Comment on Segment:** Use this icon to add comments to an Entry either as a note-to-self, or as part of a collaborative translation effort. All comments have a checkbox next to them, allowing you to mark them as settled.
- Chain link:** indicates that the segment is repeated verbatim (102% match) in the current view. Click on the icon to jump to the next repetition.
- Confirm Tick** click on the tick to Confirm the segment for the current workflow and send it to the next stage. Confirmed entries remain editable as long as they are unedited by the next workflow role.
- Workflow Status Indicator:** Display the current workflow state of the segment. Note that this might differ depending on which user in which workflow role is currently looking at the segment.

10. **Flag:** Display current translation source.

Workflow Tags

Depending on what Workflow role you're currently in, the following Workflow tags will be displayed for each segment, influencing their availability for editing:



Workflow Tags

T - Translator

P - Proofreader

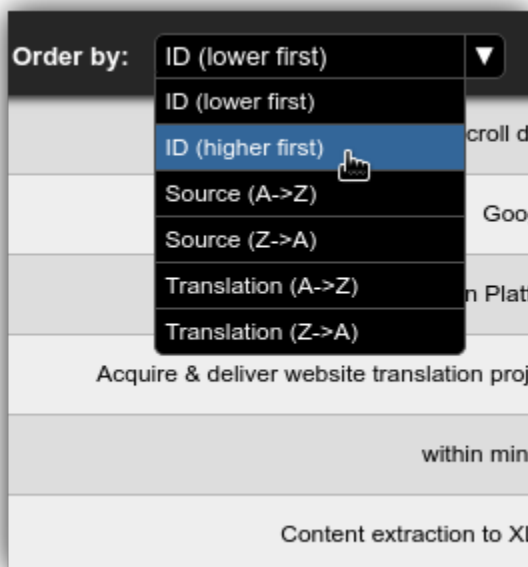
Q - Proofreader 2 (Quality Check)

C - Customer (or Client)

See the section on Workflow roles for a detailed description of the various workflow roles.

Flags

Ordering Segments in List View

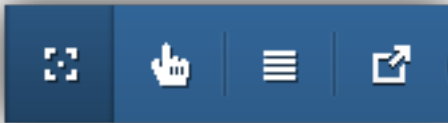


Order By Dropdown

Use the “Order by” dropdown to alphabetize the target or source entries or reverse the order of segments based on their Workbench ID. The following Ordering methods are available from the dropdown:

- **ID, lower first** (default)
- **ID, higher first** (use to quickly navigate to new segments)
- **Source A → Z**
- **Source Z → A**
- **Translation A → Z**
- **Translation Z → A**

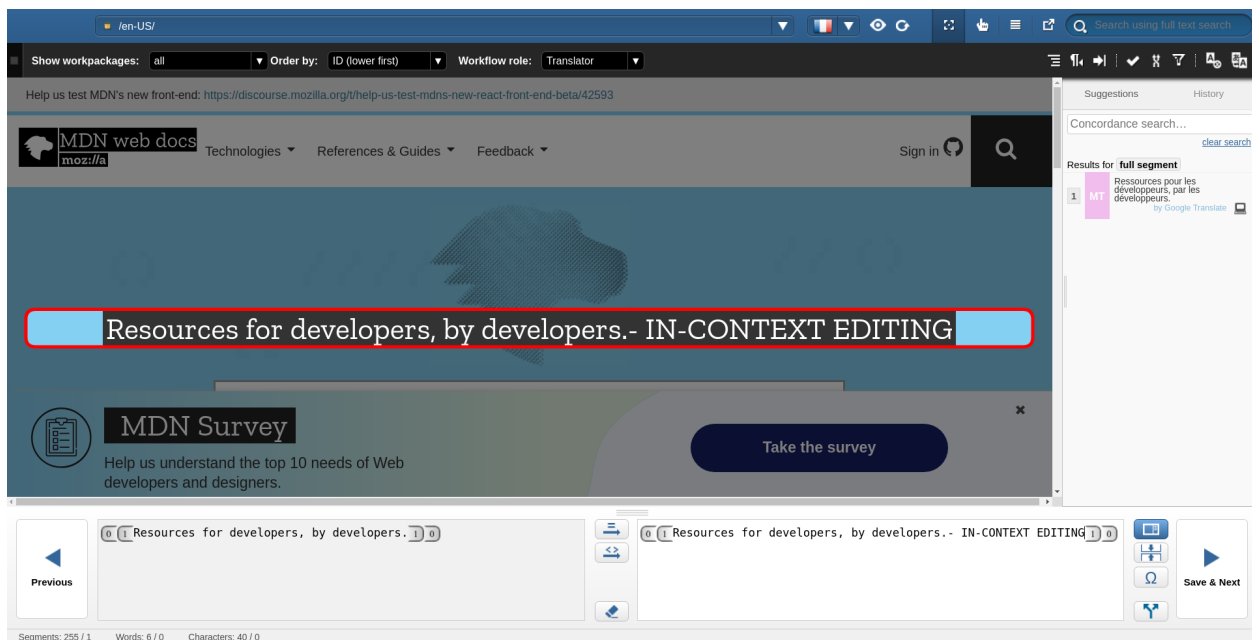
5.3.2 Highlight View



Highlight View Icon

The highlight view is a true in-context editing view that makes the Workbench popular with Translators, and *the* solution to the problem of adequate context during website localization.

By selecting an Entry in List view and clicking on the Highlight View on the Workbench, you will be shown the text on the original webpage



Highlight View

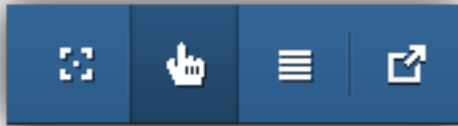
You may click on any part of a website and have a highlighting frame appear around that segment. At the same time, the editing box below will jump to the segment in question, where you can add/edit your translations in-place.

Really, the Highlight View is simplicity itself with very little in the way of hidden gotchas. Select a page, point & click, and translate away!

But keep in mind that while you are using the Highlight View with a page, links will be unclickable - use Free-click View to navigate on the original site from within the Workbench.

NOTE The Highlight view is a wonderful tool we are very proud of, but don't forget that much of the textual content of a website is not clickable. Check the other modes and the Preview to make sure that everything is covered!

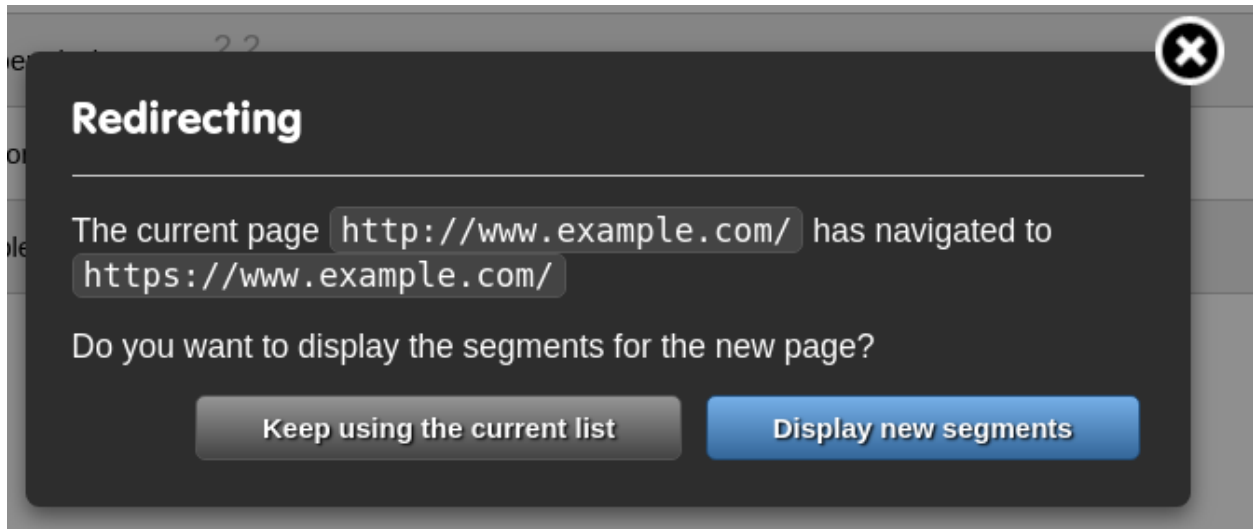
5.3.3 Free-Click View



Free-click View Icon

The Free-Click View is much like the Highlight View, but it allows you to use the links on a site to visit other parts of the website. Alternate between Free-click view and Highlight View to translate segments as you explore the website.

Free-click view will offer to reload your segment list based on which page you are on. The following dialog will be displayed.

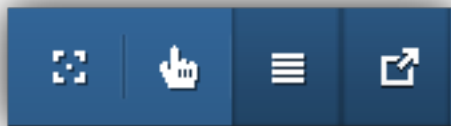


Reload

Segment List

Highlight View will only work if the segment list is that of the current page.

5.3.4 Pop-out View



Pop-out View Icon

The Workbench is a single-viewport application, and the Pop-out View is a feature meant to allow you to have the Highlight View and the List View on your screen **at the same time**. Select an Entry from the list and click on the Pop-out view icon to open a Highlight View in a separate pop-up window.

Modern browsers block popup windows by default, so you will most likely have to **enable popups** for the Workbench for this feature to work.

5.4 Translation of Segments

We are getting to the primary purpose of the Workbench, which is of course, translation of text. In this section, we'll talk about the different ways of doing that.

It bears mentioning right off the bat that the Workbench is not meant to selfishly yank you out of your CAT tool of choice. As the proxy supports exporting segments in the industry-standard XLIFF format, you can always employ any tool of your preference, SDL Trados and MemoQ, for example, being two major players in the field. The choice is yours.

Nevertheless, you will find that the Workbench is closer to where the website really is - up there in the cloud, and therefore immensely useful during the editing process. You'll see it truly begins to shine when your XLIFF files have done their first round-trip to and from external CAT tools.

There are multiple ways of translating content on the Workbench:

- manual Translation using the Target editor
- using Translation Memories
- using Machine Translation
- importing translated XLIFF files

The following sections provide an overview of the translation methods listed above.

5.4.1 Editing segments

Editing of segments happens in the Target editor located on the bottom of the Workbench:



window

Editing

You can use the Highlight View or the List View to select a segment, and the editor will reflect your selection. The Source entry is displayed to the left, the translation (Target) to the right. Only the target is editable.

Editing Window

Editing translations on the Web means going around/avoiding/leaving intact the forest of HTML tags that the text is usually embedded on. The Workbench abstracts away these markup details to ease working with text (the first and foremost task of translators), but doesn't, and will never attempt to hide the fact of their existence.

Tags are represented as numbered grey widgets around certain parts of the text, which you can use as a yardstick to place your translations in the appropriate tag context without having to worry about what those tags actually do.

Two things:

- **You can Drag & Drop the Numbered Tags**
- **You can NOT delete them**

Adding translations is otherwise straightforward text input. Untranslated entries will contain the Source text as a placeholder until such time as a change is made to the contents of the Target, at which point both the List View and the Segment contents are updated.

There are a few smaller buttons in various parts of the Target editor. In the middle, you'll see these three buttons:



Source-and-Target Edits

These contain editing functions that relate the contents of the Source and the Target in different ways. Use the top “**Equal**” button to copy the Source contents to the Target, rendering them identical. You can use the middle, “**Tag**” button to copy the tag structure of the Source to the Target. The lower “**Eraser**” button will delete the translation and restore the placeholder.

There is another set of four buttons to the right:



Target Editing Buttons

They all deal with the Target. In descending order, these are:

1. **Toggle sidebar**: use this to hide the Suggestions & History sidebar above the button.
2. **Preserve Whitespace**: prevent trimming of whitespaces in the segment
3. **Insert non-printing characters** There are a number of non-printing characters you might need during a translation project, especially if you are dealing with languages that have a Right-to-Left writing direction, such as Arabic. See the section on RTL conversion for further details.
4. **Split segment from group**

Saving changes

Click on “Save & Next” to save your work on the given Entry and jump to the next segment. You can also use Ctrl+Up or Ctrl+Down to do this. This is an explicit, although not strictly necessary action, as any edits are **automatically saved** upon leaving a page, or otherwise after 60 seconds of inactivity on the Workbench. Any navigation generated by the Workbench itself will also trigger a flush of all unsaved segments.

5.4.2 Automatic Translation

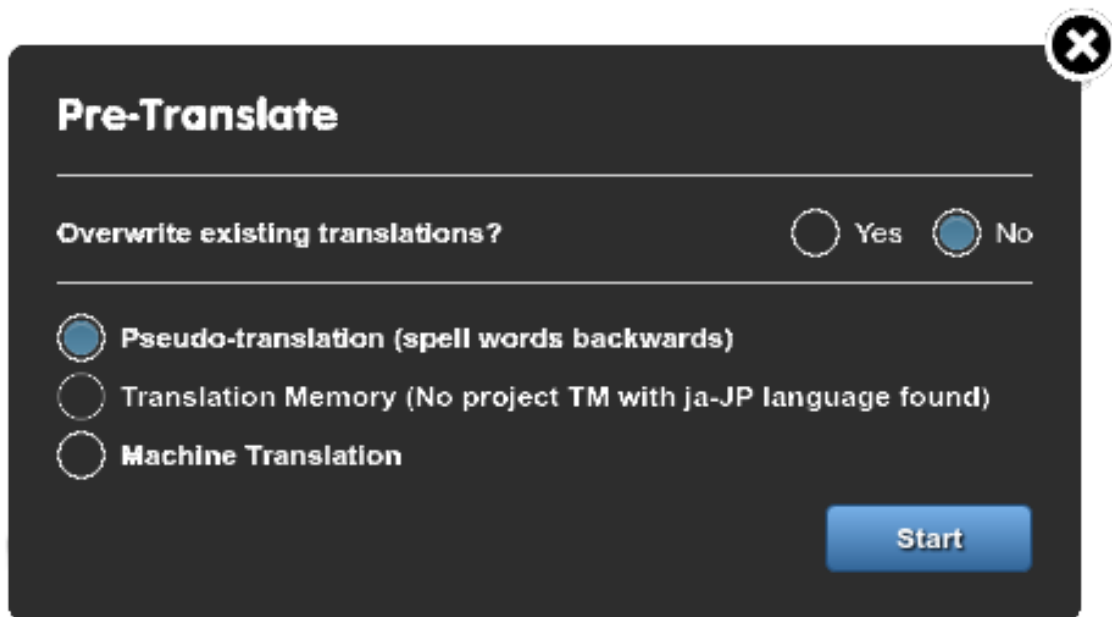
There are a number of Auto-translation features that you can employ to ensure quality on the proxy.

The option to run pre-translation or set it up to run automatically is also available on the Dashboard. Here, we discuss the options available after clicking on the “Pre-translate” icon in the toolbar:



Pre-translate Icon

The following dialog is displayed:



Translate Dialog

Pre-

Pseudo-translation

This function is admittedly not translation in a meaningful sense, but string-reversing each and every word of every segment on the spot is very effective during a demo. Use it, then go to Preview to demonstrate to the client that website localization is as painless as using CAT tools in tandem with the Proxy. Good for the wow!-factor. You can also use pseudo-translation to test the various editing features available on the Dashboard and the Workbench.

Translation Memory

If you have a populated Translation Memory on a project, you may use its contents to translate segments. Use this feature to translate your content with a preset match threshold.

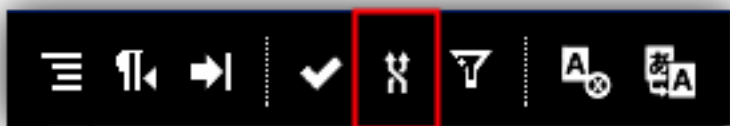
Machine Translation

You can choose to Machine Translate the currently selected batch of entries using one of the available MT APIs (Google Translate, Bing Translate, iTranslate4u and GeoFluent).

Translation Memory and Machine pre-translation are both reproductions of the options accessible on the Dashboard, with the added functionality of being able to control which specific group of segments should be targeted by the process.

5.4.3 Search & Replace

The Workbench has a Search & Replace feature you can access by clicking on this icon in List View:



Search & Replace Icon

It always operates on the currently listed set (or subset) of editable segments. The dialog that opens will look like this:

Search

& Replace Dialog

Both ‘Search in target’ and ‘Replace in target’ values are required, while ‘Filter by source’ is optional. You can use either simple strings or regular expressions as search terms (simple strings are the default, check the ‘Regex’ option next to the field to use a regular expression). The function is case-sensitive.

Choose ‘Test run (no changes will be made)’ to check how the operation would affect your translation. To see the entries affected by replacement, check the ‘Preview changes on a few segments first’; the segments will appear in the Preview area (Preview can’t be used without Test run). You can also start replacement right away, without preliminary checking.

Once the operation is ready, you will receive an e-mail notification with a detailed report on the entries affected.

Please note that the replacement is done in the database, not in the TM. If you want your TM to reflect these changes, you need to run ‘Populate TM’ from the Dashboard.

Keep in mind that **the operation cannot be undone!**

5.4.4 History

The proxy keeps tabs on what happens to each Entry in the project timeline, and the Workbench displays these tabs in the sidebar. You can use these to access previous editing states of an Entry. Here follows a short description of the

History functionality.

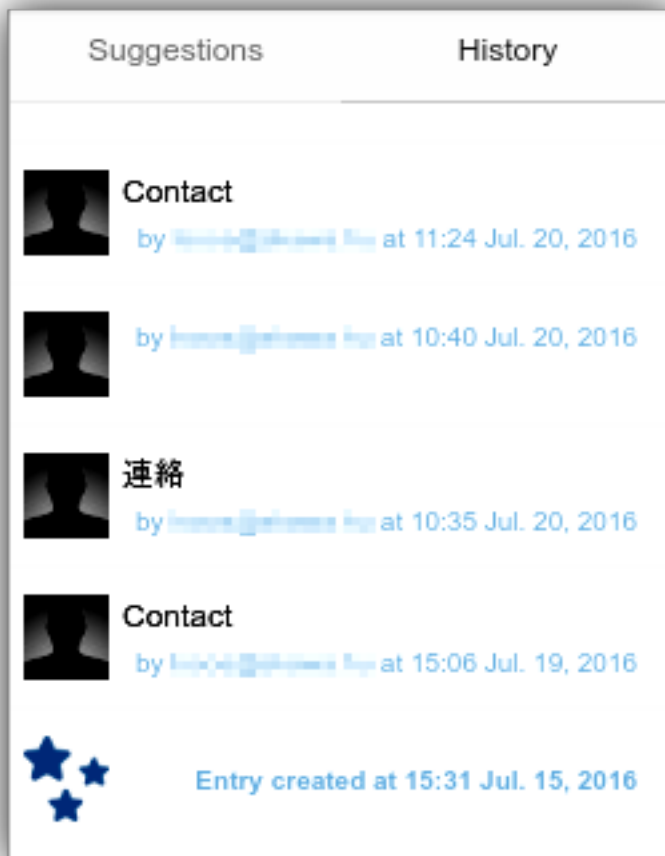
Translation Memories

If you select a segment, TM suggestions will be displayed for it in the sidebar tab labeled “Suggestions”. Click on any one of them to add it as a translation for the given Entry.

A Search field is also provided that you can use for concordance lookups.

Segment History

Whether a result of a manual or an automatic edit, each saved state of an Entry will be saved with a username and a timestamp in the Entry history. You can access it in the sidebar tab labeled “History”.



Entry History

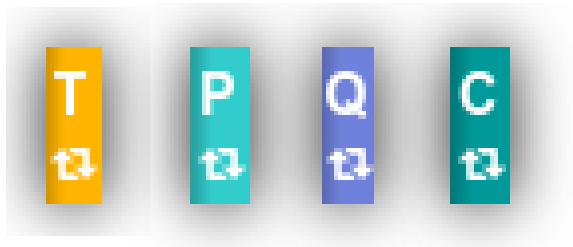
This means you don’t have to worry about ever losing translated content as a result of manual edits - you can always restore a previous state of an Entry by selecting the Entry in List View or Highlight View, going to History and copy & pasting a previous state of your choosing.

5.5 Collaboration

Collaboration are a must with website localization. If you have used the Dashboard's Sharing settings to invite other people into the project, granted them the appropriate editing rights, the list of segments in the Workbench will look a bit different to each user.

5.5.1 Workflow Roles

As mentioned previously, roles are predominantly a project management feature associated with work on the Workbench. To reiterate, there are four different roles:



Workflow Roles

T - Translator (default)

P - Proofreader

Q - Proofreader 2 (Quality Check)

C - Customer (or Client)

There are four different workflows on the proxy you may employ. You may set these on the Dashboard.

- Simple Translation Workflow (**T**)
- Translation + Proofreading (**TP**)
- Translation + Proofreading + Client Approval (**TPC**)
- Translation + Proofreading + Quality Check + Client Approval (**TPQC**)

Each setting will activate the necessary roles, which the Owner or Backup Owner may assign to any project participant. By default, only the Translator role is required. Owners have access to all workflow roles.

Workflow Roles in Action

Use the Workflow role dropdown in the toolbar to switch between the available Workflow roles:



Workflow Role Dropdown

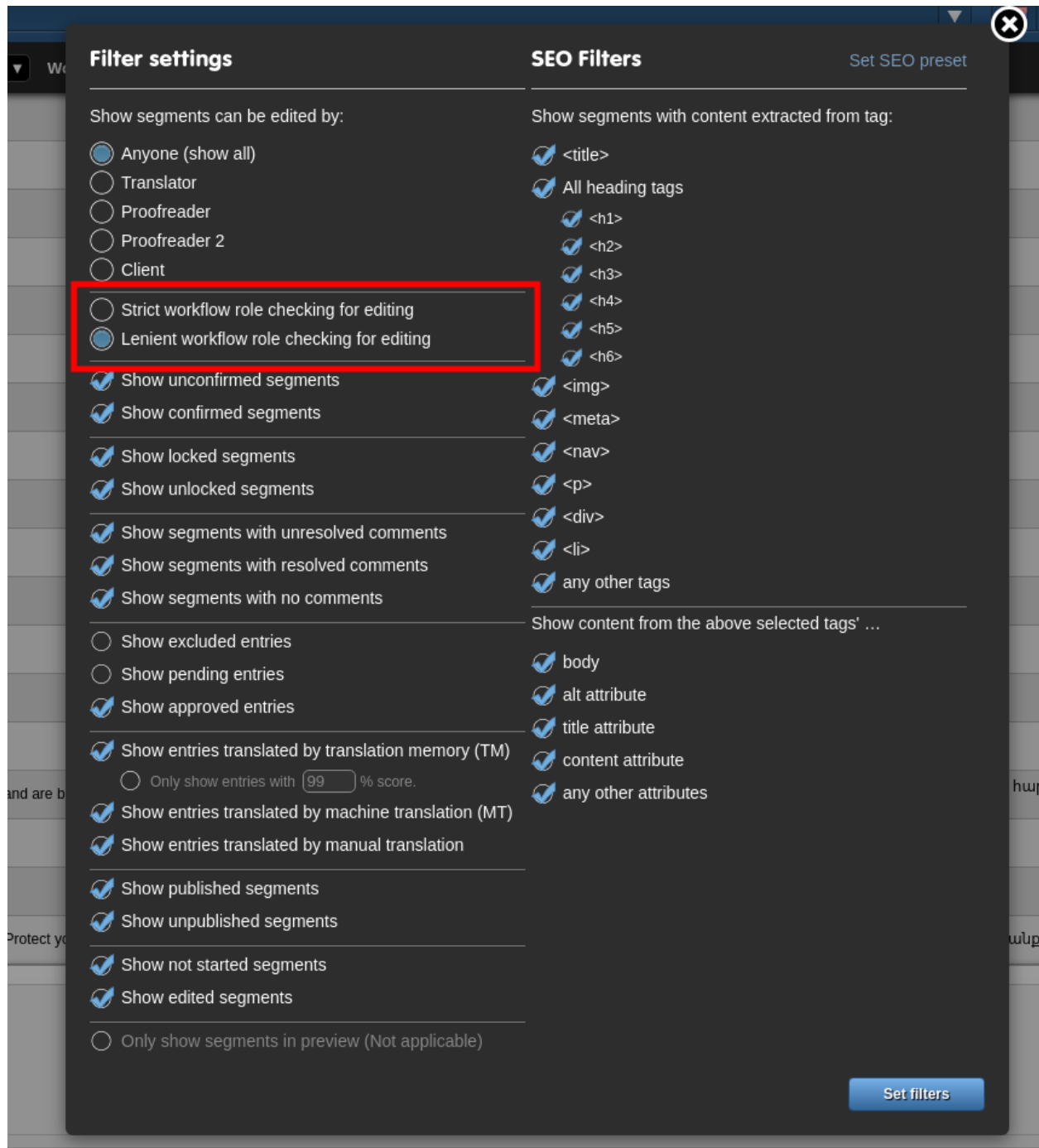
Take TPQC, the workflow with the most participants, for example.

1. Each **approved** segment is assigned to the Translator role.
2. When finished with a segment (either through manual edits, automatic translation or via XLIFF importing), the translator clicks on the Confirm tick to declare that segment cleared for that phase and send it to the next role, the *proofreader*.
3. The *proofreader* (and much everyone else) may use Filters to display only those segments that are assigned to that role. He takes the segments sent by the Translator, edits them, changes their wording as required. When finished with an entry, the *proofreader* clicks on the Confirm tick, sending the segment along to *Quality Check*.

And so on. This cycle is then repeated until a segment (more to the point, *all* segments) reach the final workflow role, that of the Customer, who approves translated entries.

A few things to keep in mind:

- Each Role has access to the lists of **upstream** roles.
- Only Owners, Backup Owners and Project Managers have access to **all** roles.
- By default option *Lenient workflow role checking for editing* is enabled, which means you don't need to switch between workflow roles in order to be able to edit segments that belong to different roles. If you are a *Proofreader2* for example, you will be able to edit and/or confirm segments that belong to the **upstream roles** (*Proofreader*, *Translator* in this case).
- If you select option *Strict workflow role checking for editing*, then the entries/segments belonging to another role will be **greyed out**, and you could only edit those segments if you have access to that role, so that you can select it.



Workflow

role checking options

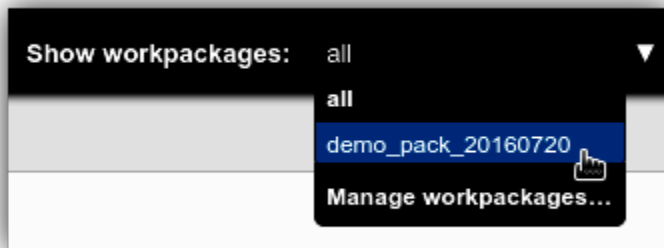
- A segment remains available for editing after Confirming it just as long as it is not touched by the next Workflow role. If you ever mistakenly Confirm an entry, you may, so to speak, **reclaim** it for some more work before the next role can get to it.

And that's about it!

5.5.2 Work Packages

If Workflow Roles is a method of grouping your users, then Work Packages are a method of grouping segments. See the Dashboard chapter for the details of how to generate them.

Use the Work Package dropdown to select a Work package, and the Workbench will display only those Entries that belong to that Work Package (note that an Entry may belong to more than one Work Package!). The dropdown looks like this.



Work Package Dropdown

The only default entry in the dropdown is “All”, which means disabling Work Package based filtering on the Workbench. As new Work packages are generated, this list will automatically update after refreshing the window. The dropdown will always contain the names of the 100 latest Work Packages.

Clicking on “Manage workpackages...” will take you to the Dashboard where you can tend to your existing Work Packages or generate new ones.

5.6 Cookbook

5.6.1 Search & Replace

Tutorial 1: Fix Spelling (String Replacement)

A simple use case of the *search & replace* feature is the old chestnut: the differences between British and American spelling rules.

This could come up whenever you are working on both the `en-US` and `en-GB` locales, or if two translators, each on a different side of the pond, forgot to coordinate their spelling.

Let’s say you have the following targets with German as a source language:

```
The world's No.1 donut
Vanilla donut
Chocolate-chip donut
Doughnut miss it!
```

and so on. Replace all instances of the word `donut` with the `doughnut` variant by following the steps below.

1. Click on the Search & Replace icon



Search & Replace

1. Fill in the `Filter by source` field to work exclusively on those entries that contain the given string in the source language. This example is about the spelling of “donut”, so you would enter the German original, `Krapfen`, to limit the search. Rest assured: source entries are **never** changed.
2. Enter the word that you’d like to replace in the `Search in target` field, in this case, `donut`.
3. Enter the replacement, `doughnut`, in the `Replace in target` field.

That’s all there is to setting things up, the rest is about making sure your changes will not cause any problems.

Search and Replace

This Search & Replace function operates on the set of editable segments currently listed, and the changes are saved in the database (not in TM). Once the operation is ready, you receive an e-mail notification with a detailed report on the entries affected. Please note that the operation cannot be undone!

Details

Replacement scope: `http://www.example.com/` Target locale: `de` Workpackage: `all` Affected entry count: `5`

Filter by source ☐ Regex

Search in target ☐ Regex

Replace in target

Preview

Preview unavailable

To preview the results of the replacement, tick in 'preview changes' and perform a search

☒ Preview changes on a few segments first
 ☒ Test run (no changes will be made)
 Cancel

Settings

Before Preview

1. Click on **Preview!** to see your changes applied to a subset of segments.
2. If contents of the Preview area look good, uncheck the Preview checkbox.
3. (Optionally)do a Test run by clicking on “Go!”
4. Uncheck the `Test run` checkbox and click on “Go!” to really apply the replacement.

Depending on the number of segments, the process can take some time to finish. You will receive an e-mail for both the Test and Live modes, containing a list of proposed (Test) or applied (Live) changes.

Tutorial 2: to-do (Regex Replace)

6.1 Project maintenance

The translation project of a website is a continuous task, as new content is regularly added to the site. So uploading the initial translation is not the end of the project, but the beginning of a new phase. This phase is practically a repetitive cycle of the following activities, many of which can be automated:

- checking the site for new content
- extracting new content for translation
- translation of the new content
- uploading the new translation
- and, of course, error fixing, if needed

6.2 Automation possibilities

Crawls can be set to be carried out automatically at daily, weekly or monthly intervals, depending on how frequently content is updated on the website. This is called **scheduled crawl**. If you turn on this feature, changes will be checked and retrieved for translation at the specified intervals, and an e-mail notification will be sent to all project participants. The new content is available right away for translation in the online editor interface, and you can also download an XLIFF for translation.

Please note that this check is technically content extraction, so it has an associated cost of EUR 2 per 1000 words.

You can enable this option in **Dashboard 2.0 > Crawl > Crawl wizard**.

Progress bar: Type (1), Scope (2), Pages (3), Fine-tune (4), Summary (5). Current step: Fine-tune.

Configuration tabs: ORIGIN (SOURCE) SNAPSHOTS, COLLECT RESOURCES, LIMIT TO/EXCLUDE, **RECURRENCE**, SESSION COOKIE.

Recurrence

With recurrence you can schedule a crawl to run at specified intervals.

🕒 Current crawl: non recurring ▼

The crawl is going to start ~Now

Buttons: BACK, NEXT

Scheduled

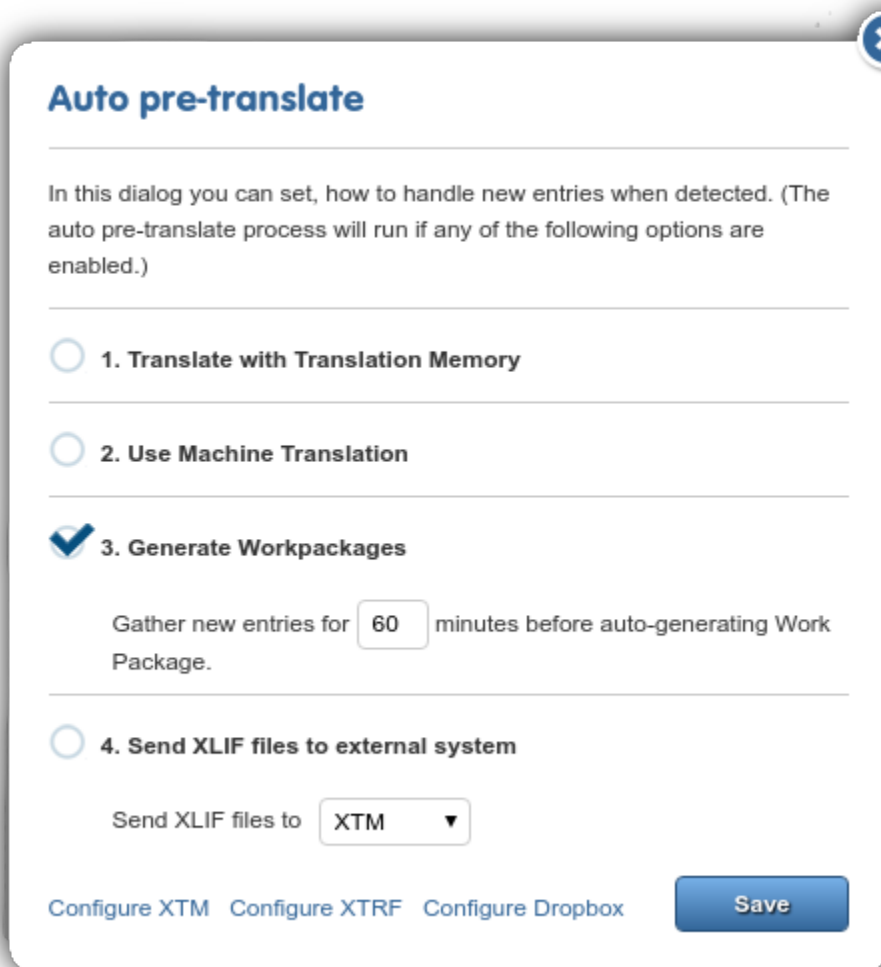
Scan

The process can automatically extract new content, apply the associated translation memories and machine translation services, prepare a work package and the XLIFF export. This is particularly useful for fast-moving sites where content arrives quickly and time spent untranslated needs to be minimized (possibly at the expense of real-time human oversight). Read more about [automatic pre-translation](#) [here](#).

6.3 Automatic pre-translation

The proxy can automatically pre-translate incoming content without user intervention or oversight, feeding its translation engine from saved translations (from the translation proxy Translation Memory) above a certain confidence level or using configured machine translation engines (Google Translate, Bing Translate, iTranslate4EU, and GeoFluent, currently).

If new content is encountered, and at least one source is configured, a user-configurable timer starts counting down. Content is collected during this timer, and automatically translated using the configured sources. At the end of the configured window, any content that cannot be translated with the assigned sources (no matches of the desired confidence are found in the TM or the MT-engine returns no translation) is packaged into an auto-generated Work Package and exported into an XLIFF file (being pushed directly to XTM, if an integration is configured). The resulting export can then be translated in any external system and imported back normally.



Auto pre-translate

In this dialog you can set, how to handle new entries when detected. (The auto pre-translate process will run if any of the following options are enabled.)

☐ 1. Translate with Translation Memory

☐ 2. Use Machine Translation

☒ 3. Generate Workpackages

Gather new entries for minutes before auto-generating Work Package.

☐ 4. Send XLIF files to external system

Send XLIF files to

[Configure XTM](#) [Configure XTRF](#) [Configure Dropbox](#) [Save](#)

ities

Automation possibil-

6.4 Caching And The Proxy

6.4.1 Overview

Nearly all browsers today implement local caches to accelerate page loading and prevent unnecessary requests from being sent out to the network. However, the operation of these caches is tied to the presence of certain headers on the page, such as `Pragma` and `Cache-Control` - based on their presence and the values communicated in these headers, the browser (and various systems, such as CDNs) may make a decision to intercept the request and serve up certain content without requesting it anew from the server.

Normally, the Proxy simply forwards these headers, much the same way it does with any other. The option to override their presence and values exists (see the [Path Settings](#) option on the Dashboard), but by default, they are left unmodified, in the spirit of minimum invasion. This is not always desirable, however, as a site without such cache headers will remain uncached in the visitors' devices, and each visit to the page will result in another request that is billed.

6.4.2 Inspecting Cache Headers

You can investigate how well a site may be cached using the Developer Tools in most major browsers. In Chrome, for instance, the DevTool can be summoned using by pressing the F12 key (or Alt+Cmd+I on under Mac OS), and after refreshing the page, the Network tab can be used to browse traffic associated with the tab. By selecting any entry in the list, you can view its details, in particular, the request and response headers. To tell whether or not a given resource will be requested again, you need to look at the “Response Headers” section, and look for the keys `Cache-Control` and `Pragma`.

If you see `Pragma:private` and/or `Cache-control:no-cache`, it is safe to say that the given resource will not be cached and each visitor will result in another hit. Files like this will likely prove resource drains if the site receives large amounts of traffic. On the other hand, `Pragma:public` and `Cache-Control:public, max-age=\d+` (where `\d+` means at least one digit, or more) are good signs in that these files will be stored on the client’s device after the first request, and will not be requested until `max-age` seconds have elapsed since the last load, and will save resources in the long run. Of course, this also means that visitors may be seeing an “outdated” version of the resource for a limited time before their caches expire and are reloaded.

There is a bit of a gray area when seeing `Cache-Control:must-revalidate`: this directive allows the cache to make the final decision based on its own algorithm, and the response **may** be stored, but not necessarily. When seeing this it is good to prepare for potential increased traffic, as the browser cache may or may not retain these responses.

6.4.3 What to take away

If you’re experiencing consistently large numbers of page views on a given project, it is often a good idea to suspect caches (as opposed to search bots, which cause transient spikes). In such cases, you should inspect a few pages using the DevTool, and determine whether or not the site is set up to take advantage of browser caches.

If it turns out be the case that the site is not set up to use caches, the best course of action is to notify the owners - perhaps they have a good reason for it. In any case, these headers should be added to the source, so that they provide consistent values across all languages. Alternatively, you can use the aforementioned Path Settings dialog to force the Proxy to override the cache headers and make the site cacheable, at the risk of diverging from the original, even if only for a short time.

7.1 Localising Wistia video embeds

7.1.1 What is Wistia

Wistia is a leading business video hosting platform. It specializes in creating and hosting marketing-oriented videos, with a particular focus on embedded videos playing on the website directly.

7.1.2 Why you should localise videos

Basically for the same reason you're localising your website: increased reach by speaking to people in their native tongues.

Just like how you already use the Translation Proxy to translate your website, if your marketing team creates a localised version of your marketing videos, it would be prudent to replace the existing video on your translated website, so any foreign visitors are greeted with tailored materials speaking their language.

7.1.3 How to localise embedded Wistia videos

Note that this guide assumes you have access to the proxy project so that you can configure the platform to make the necessary changes.

1. Open the website and go to the page where the original video is found.
2. Press **Ctrl-U** to open the source code (use **Line wrap** in Chrome).
3. Press **Ctrl-F** and find `wisita_async`. Here you'll find an ID for the given video. It'll look like `wistia_async_sourcecode0`. You'll need this later.
4. Go to **SETTINGS -> Path prefix overrides** on the Dashboard. Click on the relevant path. This video is in the root in our case so you need to click on `/`.

5. If the video is on a page where nothing has been replaced yet, it is not added to the paths yet either. In such case click on the **ADD NEW** button, enter the URL and click on the **ADD NEW PATH** button. and then click on the now listed path.
6. Select **CONTENT SEARCH & REPLACE** and click on the **ADD NEW** button.
7. Enter `text/html.*` in the Content-types field. It is worth adding the `.*` to the end as the server might send other info as well.
8. Paste the code selected from the source code (`wistia_async_S0urceC0de`) to the **Search** field.
9. Paste the code of the new video (`wistia_async_C0deT0Repl4ce`) to the **Replace** field.
10. Leave the **Regexp** checkbox disabled.
11. The **Target languages** field is optional so you can leave it empty.
12. Click on the **ADD** button.
13. Click on the **SAVE ALL** button.
14. Go back to the website and enter `/?test=0` to the end of the URL. The URL is now changed by the query with something that probably does not change content served and if the translated page had been cached somewhere, it cannot be used to serve the content for us so the translated video should be displayed on the site (otherwise you would have to wait until all caches are cleared). You should see the replaced video now on the website.

7.2 Whitelabel Setup

Note: This article will use variable values, that will change for everyone. These variables are written here in the UNIX style of `${VARIABLE_NAME}`. When providing us information, simply replace the entire construct (not just the name itself) with the data in mind. There is also “global” variable to keep in mind: `${APP_DOMAIN}` refers to the domain chosen in point #3 to serve the translation proxy on.

1. A €200 monthly fee: the translation proxy under your own brand name is special service, offered to customers who cater not only to one or two clients, but put their weight behind the punch and open up whole new markets with our proxy solution.
2. At least a yearly commitment to the whitelabeled brand.
3. A one-time setup fee of €200.
4. A custom domain name: you will need a place to serve the translation proxy (as well as any previews) on. Generally, our clients settle on `app.${yourdomain}.com`, but we can use practically anything that comes to your mind - the only limitation is that we are unable to serve the proxy on a naked domain (for instance, `yourproxy.com`). Just keep in mind that once you settle on something, and we set up your branded version of the proxy, it becomes fixed, so your decision is final.
5. Two logos: one goes on the Dashboard, the other goes on the Workbench. They should be transparent PNGs, ideally, but we can use other file formats as well. However, their dimensions are fixed: the Dashboard logo is set at 200x62px, while the Workbench logo needs to be 109x44px.
6. An SSL certificate: the translation proxy uses encrypted channels to communicate on, and for that, we require an SSL certificate to be made out for the domain name of your choice, and any subdomains it may have - the translation proxy uses your “app domain” to serve previews until they’re published, so your certificate must be a so-called “wildcard certificate”. This is a type of SSL certificate that is valid not only for `app.yourdomain.com`, but also `*.app.yourdomain.com`, needed in order to ensure SSL coverage of all project-specific preview subdomains (the names of which are created by combining an arbitrary locale code and the *randomly generated* project code). Certificate issuers are likely to request a Certificate Signing Request (CSR) for the certificate, which we will have to provide.

In order to generate a CSR for you, you'll need to provide a few pieces of data related to your company, which need to be incorporated into the certificate. Please provide the following by replacing the fields (this should look fairly familiar to your IT department) with the appropriate data.

```
1. `countryName_default = ${COUNTRY}`
2. `localityName_default = ${CITY}`
3. `streetAddress_default = ${ADDRESS}`
4. `postalCode_default = ${ZIP}`
5. `0.organizationName_default = ${COMPANY_NAME}`
6. `organizationalUnitName_default = ${ORG_UNIT}`
```

1. The final step is to configure your DNS servers; and if you use Google Apps for yourdomain.com, the setup process will require someone with Google Apps admin rights as well. You will need to add the following CNAME records to enable the translation proxy on your domain:
 1. \${APPENGINE_KEY}.\${APP_DOMAIN} CNAME \${APPENGINE_HASH} - these values will be provided to you.
 2. \${APP_DOMAIN} CNAME ghs.domainverify.net
 3. *.\${APP_DOMAIN} CNAME ghs.domainverify.net
2. In order for the translation proxy to be able to send emails from under your domain, you will need to provide authorization to the email service. This is done by adding a couple of specialized DNS records:
 1. \${SELECTOR}._domainkey.\${APP_DOMAIN} TXT \${DKIM_KEY} - these values will be provided to you as they are domain-dependent.
 2. v=spf1 include:sparkpostmail.com ?all; or if you're already using an SPF record, add include:sparkpostmail.com just before the last operator.
3. Finally, if you want, you can specify the following information to customize the white label experience (this is completely voluntary and can be changed at any time):
 1. name: Name of your branded product
 2. greeter: Person signing the greeting emails for new users
 3. team: Team name
 4. greeter address: Email address of the person sending the greeting emails
 5. greeter display name: Name of the person sending the greeting emails
 6. noreply address: Email address used for automated emails
 7. noreply name: Display name used for automated emails
 8. quote and wordcount signer: Person signing the quote and word count emails

Once all seven main points are settled, and we have the logos and the SSL certificate with us, we'll set up the white label version for you, and you'll be ready to start cracking your target market wide open.

7.3 JS/JSON/XML Translation

In this section, we describe the process of translating content in JavaScript files and dynamic (JSON or XML) responses.

7.3.1 General

Translation of HTML is mostly automatized over the proxy. But websites rely on many additional resources besides the document itself, such as JS libraries, CSS stylesheets, webfonts, dynamic requests and images. Not all of these resource types require translation, but JSON and XML responses frequently do. Such responses can also be very inhospitable to the translator and proxy specialist.

One of the problems is detection: proxy crawls/analyses do not operate in a browser-like environment. There is no headless browser or VM running in which a page load could be initiated or JavaScript evaluated for content detection purposes. Inherent complexity is another issue: the enormous diversity (to put it charitably) of web technologies in use nowadays prevents reliable automation of such a process.

But, though JS/dynamic content can slip under the radar at first, the proxy can easily translate it with some help.

7.3.2 Finding Content

An early investigation will reveal content that is unavailable to the crawler by default, and will save you trouble (of having to deal with untranslated content as late as in the review phase, for instance).

X-proxy

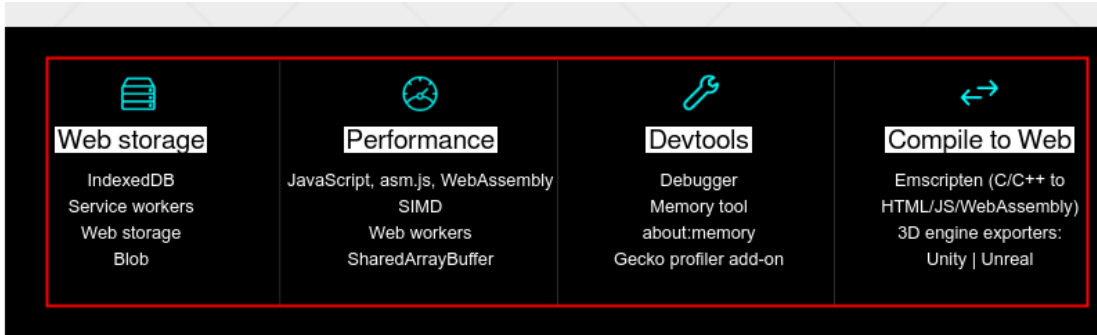
To check how a website is doing over the proxy, open it in the **X-proxy** mode, a specialized Preview mode available through the Dashboard page list. Click on Preview in the hover menu while holding `Ctrl` to open it.

Note that you need to *add at least one target language* and *select it in the left side menu* to access the preview.

The X-proxy replaces text it recognizes with x-es. Though not too impressive visually, it is an excellent research tool. It lets you *home in on undetected content*. To utilize it most effectively, combine it with your browser's DevTool.

Major browsers such as Firefox and Chrome allow you to do full-text search on (almost) all resources/requests used during and after page load. In Chrome, for example, you can press `Ctrl + Shift + F` to start a full text search in the DevTool.

The following screenshot demonstrates how the X-proxy can make untranslated/undetected content obvious:



XXXXXXXX XXX XXXXX

XXXXXXXX xx xxx XXX xxxxxxxxxxxxxxxxxxxx! Xx
 xxx xxx xx xxx xxx xxxxxxxxxxx xxx xxx
 xxxxxxxxxxx xxxxxx xx xxxxxx xxxxx. Xxx xxx xxx
 xxx xxxxxx xxxxxxxxxxx xxx xxxxxxxxxxx xxxxxxx xx
 xxx xxx xxx xx xxx xxx, xx xxx xxx xx xxxxxx.

Xx'xx xxx xxxxxxxx x xxxxxxxx xxxxxxx xx xxx xxx
 xxxxxx xxx xxxxxxxxxxx xxxxx xx xxx xxx xxxxxx
 XXXx xxx xx xxx xxxxxxxxxxx.

Xxxx: XXXXXXXX xxxxx xx xxx xxx xxxxx xx
 x xxxxxx xx xxx xxxxxxxxxxxxxxxxxxx xxx
 xx XXXX, XXX, xxx XxxxXxxxx. Xxx
 XXXXXXXX Xxxx xx x xxx xxxxxx xx xx xx
 xxx xxxxxxx xxx xxx xxxxxx.

Xxxx xxxxxxx xxxxxx

xx xxx Xxx

Xx xxx xxx x xxxxxx xxxxxxxxxxx (xxx xxxxxxx xxxxxxx
 xxxxx xx Xxx), xxx xxx xxx xxxxxxxxxxx xxx xxx xxx xxx
 xxx xxx xxx xxx xx xxx Xxx, xxx xxxxxx xxxxx
 xxx xxxxx xxx [XXXXXXX](#) xxx — xxx xx xx
 XXX xx XxxxXxxxx xxxxxxx, xxxxx xxxxx XXXX
 xxxxxxx (x.x. xxxxxxx xxx X/Xxx xxxxx Xxxx, xx
 xxx xxxxxxx xxxxxxx) xxx xxxxxxx xxx xxx
[xxx.xx](#), xxxxx xxx xx xxx xx xxx Xxx.

Xx xxx xxxxxxx, xxx:

- [XXXX XXXXXXXXXX](#) xxx xx xxxxxxxxxxx
 xxxxxxx xxx-xxxx xxxxxxx.
- [XXXXXXXX xxx XXXXXX](#) xxx xxxxxxxxxxx xxx
 xxxxxxx.
- [XXXXXXXX XXXXXXX](#) xxx x xxxxxxx xx
 xxxxx xxx xx xxx xxxxxxx.

XXXXXXXX

Xxx x xxx xx xxx xxx xxxxxxx, xxx xxx [XXXXXXXX](#)

proxy example

Having removed all known text from the equation, you are free to concentrate on the “untranslated” parts. Findings will naturally be site-specific, but there are some familiar, recurring scenarios:

1. **content is in `<script>` tags:** this is the simple case, as there isn’t even a distinct resource to mark as translatable. text/html pages are translated by default, and JS content in them can be annotated right away.
2. **content is a string in a JS file:** aside from the necessary annotation process, you’ll also need to ensure that the resource in question is marked as translatable.
3. **content is requested dynamically:** dynamic content can be tough to dig up. Many DevTools don’t support full text search in XHR response bodies. If content is in plain sight on a webpage but the DevTool is not reporting any of it, then the content could have arrived via an XHR request. Check the “XHR” section in the Network tab

after a reload. Aside from the inconvenience of locating them, dynamic request endpoints can be annotated and marked in the same way as JS files.

4. **content is on an external domain:** this scenario requires some work. External domains require separate, but linked projects (add to this that you also have to ensure that URL references are mapped well, which can be difficult in a JS file), and the resources have to be marked and annotated on those projects to be translated.
5. **content is in an image:** though not strictly connected with the topic of JS translation, an “honorable” mention goes to natural language content *as image data*, also frequently revealed by the x-proxy.

There are many ingenious ways in which webpages encode content, and the proxy has various levels of support for all these schemes (usually involving a combination of features). When in doubt, feel free to contact support for advice!

7.3.3 Marking Resources

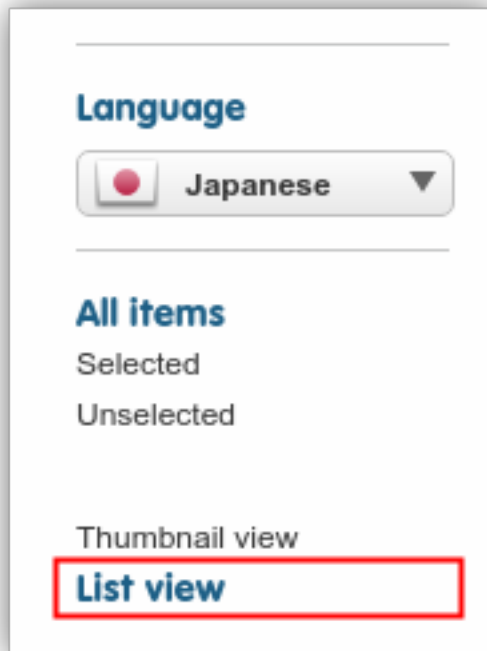
You can mark a resource as translatable manually on the Resource screen or using a prefix in Advanced Settings.

Manual

All collected resources are listed in Discovery > Resources and Content > Resources. All “pages” and files that are not of the `text/html` content type will go here.

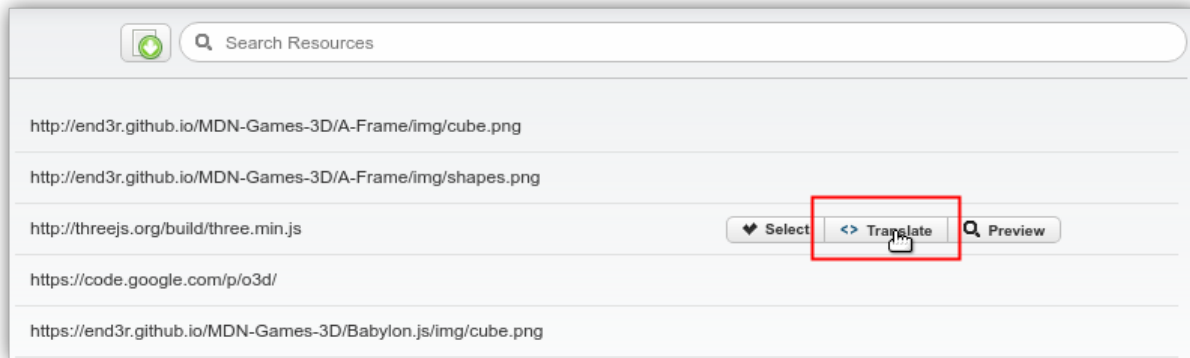
What sets resources apart from pages is that by default, they have no associated *source entries* or translations. Marking a resource as translatable means declaring that it does have translatable text that can be stored as source entries (and accessed via the Workbench).

So, to mark a resource as translatable:



Switch to List View

1. switch from thumbnail view to list view
2. click on the “Translate” button in the hover menu of a resource



Translate

Resource

The resource is moved to the page list and from that point onward, it can be opened on the Workbench. Note, however, that we have not yet told the proxy what and how to translate in it.

Prefix

You can do the marking via prefixes. Go to Advanced settings to find the “Mark Multiple Resources as Translatable” text field. Copy & paste the prefixes of your choice and click on Save to apply your changes.

Mark multiple resources as translatable

Mark multiple resources as translatable using URL prefixes.

```
https://www.example.com/wp-content/plugins/
http://www.example.com/wp-content/plugins/
```

Mark

Multiple Resources

Note that in the screenshot above, HTTP and HTTPS prefixes are handled separately, a recommended practice for sites that support both kinds of traffic. Prefixes are treated as simple strings by the proxy when matching them against a pathname. You are free to add as many of them as you like.

This feature is made available because cherry-picking resources for translation is not always feasible. For instance, versioned URLs are liable to create new resources on a project whenever a file is updated on the original site (the proxy keeps these URLs separate by default), but the new resources are not marked automatically.

You will recognize those cases where you want to apply the exact same translation rules and process to a set of URLs that differ in minimal ways. A resource prefix will let you do this without having to mark things one-by-one as they come.

7.3.4 Annotating JS/JSON

Picking up JSON/JS/XML content *wholesale* would be both costly and unwieldy. When you have successfully identified the source of content and earmarked it for translation, the last major task is to annotate those parts of it that really want to translate. **JS/JSON paths** and **Xpaths** can be used for this purpose.

JS/JSON paths

Go to JSON/JS/XML processing on the Dashboard 2.0, and select the JSON path tester tab. This tool can be used to mark specific string in JavaScript or in JSON as translatable. You can more about it [here](#)

Xpaths

Xpaths for XML AJAX responses in a similar way as JS/JSON paths do for their respective content types. For example, `/ajax-response/component[1]/text()` `html` assumes that the first `<component>` node contains translatable HTML markup.

Due to space constraints, we decline to reproduce a full Xpath tutorial in this documentation, and direct the reader's attention to the many tutorial resources available online. The W3Schools [summary of Xpath syntax](#) serves as a good starting point.

Limitations

Content extraction

A simple content extraction crawl takes care of JS content in the source of an HTML document. But in many cases, such content is requested **as part of a page load** or **via user action**. The same limitation still applies, however: the page is not available to the crawler in a way that would allow for such “interactive” requests to start.

The solution is to extract content **via Preview**. Open the page in Preview mode and go through the required user actions to trigger all necessary events. The proxy will take care of extracting content from the affected dynamic responses (provided that your setup is correct).

Note the influence of **TM Freeze** on this approach: you need to disable it temporarily for Preview-based content extraction to work.

String Concatenation & Computed Values

JS translation cannot be used with values such as `var string = "You have " + getCartItems().length + " items in your basked"`). In these cases, you either have to forego translation or change the content so that no *computed expression* is present among the concatenated elements.

This implies that those instances of string concatenation where *no* token of the expression is computable are supported. This is indeed so, and provided that the appropriate tweak is enabled in [Advanced settings](#), the proxy can perform string concatenation *upfront* and handle the resulting string as a whole.

7.4 Resource Translation

Resources are binary content found on the sites, suchg as images, PDFs, CSS and JS files, etc. Please note that the content of these resources is not extracted for translation, so you have to translate / edit them separately.

7.4.1 Replacing Images with Localized Versions

You have the option to replace images, downloadable files and other resources with their localized version.

1. Please make sure that you see the appropriate target language selected on the left side menu.
2. Navigate to the original image in the Resources view of the Content (or the Discovery) section.

3. Hover over the thumbnail image. A green ‘+’ icon should appear.
4. Click this icon. It marks the image as ‘localizable’, i.e. a candidate for replacement.
5. Select the replacement image and upload it. It will immediately replace the original image, and the new image will show up on the translated site.
6. Check the Preview to see if the image was replaced properly.
7. Repeat these steps for all images and all target languages that need localized versions.

Please note that you can replace only 1 resource, for 1 target language at one time.

7.4.2 Extraction Issues with data-image Attribute

Images might be inside a “data-image” attribute in the source code. In such cases you have to set the “data-image” attribute up to be translated as a link in the Advanced Settings screen. The `` attributes are recognized by default as URLs, while data-* attributes are not, as they can contain any sort of data, and must be configured manually to tailor the behavior to the current project’s needs.

The image might also be drawn from a subdomain. In this case you have to create a second project for the subdomain, mutually link the two projects together, and add the resource in the subdomain’s project. In this case you have to publish the two projects together, in order to preserve the mapping.

7.5 HubSpot Forms

The proxy supports translation of HubSpot (or similar) forms via a combination of [project linking](#) and [JS translation](#).

Method #1 marshals a combination of advanced proxy features. It is entirely **hands-off** from the site maintainer’s perspective, no change on the original server is necessary (which is a rather frequent constraint).

Method #2 relies on injected JS and HubSpot to provide separate, localized forms for each target language. Compared to #1, it is a clean and simple approach.

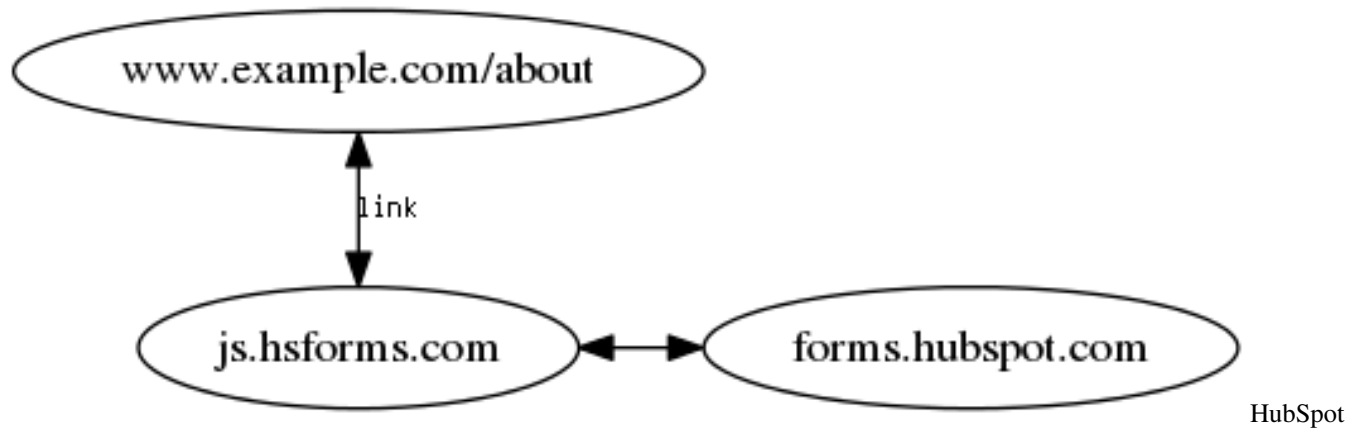
7.5.1 Method #1: Proxy

The proxy approach traces the structure of the main and form domains via linked projects. Affected JS resources/endpoints are overridden and the responses marked as translatable.

Project Creation & Setup

HubSpot uses several external domains to drive a form. You will see <https://js.hsforms.net/forms/v2.js> referenced in the page source. This file itself references <https://forms.hubspot.com>, which is where the translatable form contents are coming from.

Domains used by a HubSpot form are related to the main project and each other in the following manner:



Projects

Assuming that `example.com` is already set up, at most two additional projects are required:

1. `js.hsforms.com`: creation of this project is *optional*, though not complicated. At the time of writing, visiting the landing page results in a 403 Forbidden page (but this is not a problem).
2. `forms.hubspot.com`: this URL redirects you to `https://developers.hubspot.com/`. To create a project for it, **disable Redirect checking** in the Add Project dialog. Click on **Advanced settings** to reveal the option and uncheck the checkbox:

Create new project

Website URL 

forms.hubspot.com

English (United Kingdom) [en-GB] ▼

☐ Process pages in source language only ([requires Google language detection API](#))

Project display name/alias

☐ Start Discovery along with project creation

Page limit

10

Advanced settings

☐ Check if the website URL redirects to another domain 

(0) LIMIT TO RULES

(0) EXCLUDE RULES

You can set **limit to** rules. 

A new limit to prefix rule.

ADD

Provide custom SRX file (optional) 

Browse...

No file selected.

CANCEL

CREATE

Redirect checks

Don't forget to add every target language of the main project to each project you create.

Link Projects

Open each project in a separate tab and link each project according to the [section on Project Linking](#). The result should be a chain of projects leading from `example.com` to `forms.hubspot.com` with `js.hsforms.net` as an intermediary.

Alternative: Search & Replace

The `js.hsforms.net` project is not, strictly speaking, necessary. Its true purpose is merely to expose a slightly modified version of the `/forms/v2.js` script. If its URL is referred to in a way that makes it possible, you can *sidestep* the domain using a combination of [Search & Replace](#) and a [page content override](#). The setup steps for this are as follows (done on the main project):

1. **create a path override** for the exact URL where the form is present (the diagram above shows `/contact` as an illustration).
2. **add a Search & Replace rule:** replace `https://js.hsforms.net` (a regex matching both HTTP and HTTPS versions of the same URL) with the empty string. This turns the reference to `/forms/v2.js` into a relative URL, pointing it toward a page content override that is to be created in a moment).

Overriding `v2.js`

This resource contains a crucial variable called `urlRoot`, which has to be **remappable** over the proxy. However, it is set via a *computed expression*, which is unsupported by the proxy for reasons discussed in [the section on JS translation](#), so an override and a small change is unavoidable (regardless of the presence/absence of the intermediate project). Follow the steps below to create the override:

1. visit `https://js.hsforms.net/forms/v2.js` and **copy & paste** the contents of the JS file.
2. use the DevTool or an [online pretty printer](#) before pasting the code. Though optional, it is highly recommended that you do this (such minified code is cumbersome to work with as it is).
3. create a PCO for the `/forms/v2.js` pathname in Page modifiers > Content Override. The response code default is 200, and the Content-Type header is `application/javascript; charset=utf-8`. We'll return to Cache-Control and Pragma later, after setup is complete.
4. Add the following line to the top of the PCO:

```
var HUBSPOT_URL_ROOT = "https://forms.hubspot.com";
```

1. Search for `this.urlRoot`. It is set in a line similar to the one below:

```
o ? this.urlRoot = "https://f.hsforms" + e + ".net/fallback" : null != a ? this.  
↪urlRoot = "" + a : this.urlRoot = "https://forms.hubspot" + e + ".com";
```

1. Add the following line after it to make it use the “accessible” value:

```
this.urlRoot = HUBSPOT_URL_ROOT
```

1. Use the *Mark multiple resources as Translatable* text field in Advanced settings. Simply add the pathname prefix of the PCO to the list:


```
/forms/v2.js
```

1. URL Translation & HTTP/HTTPS Finally, add the following JS path to the list of translatable paths in Advanced settings:

```
"HUBSPOT_URL_ROOT" url
```

Open the PCO link over any one of the proxy preview domains to test it. If all projects are correctly linked and you followed the setup steps correctly, the `HUBSPOT_URL_ROOT` variable will hold an appropriate proxy-mapped domain (and consequently `this.urlRoot` will be set to the same value).

Form Contents

Set up the HubSpot content endpoint as translatable on the project for `forms.hubspot.com` [according to the JS translation section](#). In summary:

1. locate the form request using the DevTool and add it to the “Mark multiple resources as translatable” list of prefixes. For any given HubSpot form, translatable content will usually be associated with a prefix similar to the one below (it will also have a `callback` query parameter).

```
/embed/v3/form/{numericId}/{formId}
```

1. use the JSON path tester tool in Advanced settings to process the response. HubSpot forms come in a response format called JSONP or padded JSON (a function call such as `hs_request_0` with the form data passed to it as argument). It is not necessary to prefix JS paths with `"json"` in this case.
2. use the x-proxy to test your JS paths, and use Preview for content extraction.

Publishing & Caching

All projects need to be published together in all target languages. Note that you don’t need to publish on a subdomain of the original server: you are free to proxy the German version of `forms.hubspot.com` through `hs-de.mydomain.com`, for example.

Once setup, translation and publishing is complete, you are free to set an appropriate Cache Header on your page content overrides (either on the PCO itself or [on a prefix-basis](#)) to reduce page request costs.

7.5.2 Method #2: HubSpot & Injected JS

You can rely on HubSpot to localize the form property names after [cloning your form for each target language](#), and rely on a little JavaScript to drive the forms on the client-side for each target language. This approach is cleaner than the one described above, as long as you don’t mind having a separate form for each target language.

The proxy sets the `lang` attribute of the `<html>` tag to the appropriate locale code on each target language domain, which you can use for branching. The code below demonstrates one example of how such code could look in practice:

```
var lang = document.querySelector("html").getAttribute("lang");

var HSFormId = {
  "en-US": "English9-bb4c-45b4-8e32-21cdeaa3a7f0",
  "fr-FR": "Frenche9-bb4c-45b4-8e32-21cdeaa3a7f0",
  "de-DE": "Germane9-bb4c-45b4-8e32-21cdeaa3a7f0"
};
```

(continues on next page)

(continued from previous page)

```
// refer to the HubSpot documentation for further customization at
// https://developers.hubspot.com/docs/methods/forms/advanced_form_options
hbspt.forms.create({
  portalId: "portalId",
  formId: HubSpotFormId[lang]
})
```

7.6 Page requests, CDNs and Caching

The number of page requests sent when a page loads predicts monthly costs on a project. The goal of this page is to provide an in-depth description (and an upfront summary) of the matter and offer suggestions for cost reduction.

7.6.1 Summary

Page requests (in the strict technical sense of an **HTTP request**) are a primary concern over the proxy as they form the basis of monthly costs. Since each HTTP request that hits the proxy is billed, it is useful to consider various ways of reducing the number of hits.

The three methods discussed here, **URL remap prevention**, **use of CDNs** and **public caching** have the potential to dramatically reduce monthly costs (and can be combined according to need).

7.6.2 Page Views vs. HTTP requests

The difference between a *page view* as generally seen and a *page request* as understood as part of pricing is crucial to keep in mind.

Views

Website owners/maintainers usually focus on tracking the number of *page views* (via Google Analytics, for example). This is a useful metric to gain insight about the visitors to a site and make various predictions/business decisions based on user traffic.

Such analytics count *end user visits*, where we generally expect one additional page view to show up in our Google Analytics View each time a user loads a page on the site.

Requests from bots of search engines and request where Javascript is not run (e.g. a JS-disabled browser, HTTPtrack or command-line tools such as `cURL` and `wget`) do not result in a “visit”.

The proxy pricing system **does not** refer to this understanding of a page request.

HTTP Requests

Refreshing the page with a devtool’s Network tab openend, or using the Content Breakdown section of www.webpagetest.org for a page shows you that a modern webpage heavily relies on multiple resources (HTML, CSS, JS files, images, fonts etc.) over several domains to construct the unified whole shown to the user. By *page requests*, then, we mean the number of distinct **HTTP requests** for such resources.

It is this sense that the pricing system uses the term. The proxy is a technical solution to process and translate HTTP requests between the visitor and the original site, and any HTTP request that has to be relayed between the user and the server is counted as 1 page request.

This is regardless of the type of content: whether HTML, an [XHR/AJAX request](#) or a static resource, it will be counted as a page request if the proxy has to process it.

It is easy to see now why understanding the number of requests going into a page load is important – they act as a multiplier on the number of **page visits** and become an important predictor of monthly project costs.

The ideal case, of course, is 1 page request per 1 user visit (meaning that only the HTML document has to be translated and served). Although this ideal case might not be attainable in all cases, it very often is.

See the next section for a suggested manual approach of evaluating a page load.

In this section, we'll go over the various ways you can *reduce the number of requests* and consequently, project costs.

7.6.3 Possible approaches

Optimization means preventing HTTP requests from reaching the proxy. In practice, there are three general approaches:

1. prevent URL remapping for non-localized resources
2. ensure that “auxiliary” content such as CSS and JS files and images is served from a CDN
3. caching intra-domain resources

Remapping

The proxy billing system is only concerned with those requests that are forced to go through it, so the simplest way of preventing an HTTP request from going through the proxy is to prevent it from being re-mapped from the original site to the translated domain entirely.

If `www.example.com` is being translated into German and published on `de.example.com`, any URLs in the source that point to the original server (that is, are intra-domain) will be **remapped** to refer to the translated domain over the proxy.

But useful exceptions should be made. For example, images that are not localized do not get mapped by default, which means that the image will be downloaded from the original `www.example.com` site instead of the TL domain, naturally preventing an HTTP request from going through the proxy. Although a tweak exists in Advanced settings to force images through the proxy, you should consider the cost implications of this tweak before turning it on (and look into the subsection on caching to offset increased costs).

The `__ptNoRemap` HTML class is handled specially by the proxy. If this class name is detected, the `href` or `src` attribute of the given element will not be mapped to the proxy (avoiding the request cost).

For example, on a project for `www.example.com`, an anchor tag such as

```
<script src="https://www.example.com/client.min.js"></script>
```

would be remapped by default, and it would look like this if published on `de.example.com`:

```
<script src="https://de.example.com/client.min.js"></script>
```

The `__ptNoRemap` class disables this default action, so the script `src` is not remapped even if the page is opened over the proxy:

```
<script src="https://www.example.com/client.min.js" class="__ptNoRemap"></script>
```

Used in a systematic manner, this change has to be applied on the origin server, which is a potential downside if you don't have source/admin access. It bears mentioning that the class is reported to have solved one-off problems by being search & replaced into targeted spots in the page source.

CDNs

The simplest way to prevent an HTTP request to the project domain is to offload it from the original server, too. Content Delivery Networks are servers that are capable of providing source (non-localized) content in a reliable way across the globe.

Public Caching

(Note: not to be confused with Source and Target caches!)

Many static resources need no processing whatsoever by the translation proxy, and become cost overhead if funneled through it. But it is also often the case that it is impossible to avoid having these request go through. What to do in these cases? Enter public caching.

HTTP supports what is called a `Cache-Control` header that can be added by the server to a response. The content of this header instructs public caching nodes on the network on how to store a static copy of the resource for a time.

A cache (usually affecting a given geographical area or specific network pathway) will serve the resource to visitors until the “Time-To-Live” of the cached resource expires. This TTL is defined by the value of `max-age` in the `Cache-Control` header, and its current value is tracked in the `Age` header.

Until such time as this time is up, requests coming in from a place served by the cache will not, in fact, reach the proxy. After `max-age` expires, the cache will re-request the original for another `max-age` term - during that time, however, a cache can serve a multitude of requests without having to burden the original server (& proxy for it).

Declaring a `max-age` of 86,400 on the image `/about-us/logo.jpg`, for example, broadcasts on the network that for the duration of one day, any public caching node should feel free to cache the resource – it’s up to them, but if possible, they should not re-request it until then.

This way, the caching/serving/trafficking burden evens out on the network, and many of the repeating requests can be avoided (i.e. an intently browsing user might load the same resource over and over again, but most of those requests will be served from a cache).

Keep the following important points in mind:

1. Caching naturally introduces **update delays**. A user getting cached content will have to wait for it expire before seeing a new version.
2. Frequently updated HTML pages (such as those with a newsstream) should not be targeted for caching, and URLs serving **dynamic resources should never be cached!**
3. Public caching works best with **static resources** (JS, CSS and images) and **versioned URLs**.
4. It is **impossible** to reach a cached resource coming in from the server side. Neither the original site, nor the proxy can tell a node to throw a cached version away. Only expiry will do that. This is architectural, the HTTP protocol does not provide a method to send cache invalidation notifications to arbitrary nodes. The upshot is that while it might seem sensible (from a cost-reduction perspective) to add as huge `max-age` values as possible, this is highly recommended **against**. Unless the URL of the given resource is versioned to allow for updates anytime, users may end up “walled out” by a cache storing an image for weeks, for example.

A measure of carefulness is advised!

Cache-Header Setting

If you find that very sensible defaults are coming in from the original server, that is very good news, but it is also a fact that such is not always the case.

To adjust `Cache-Control` on the proxy-side, go to Dashboard > Path settings to override headers on a URL or prefix-basis. See the [Path settings](#) documentation to learn more about overriding/fine-tuning Cache-Control headers.

7.6.4 References

See [HTTP caching](#) in Google's Web Fundamentals on public caching.

See [RFC 7234](#) for the full technical detail on HTTP Cache-Control.

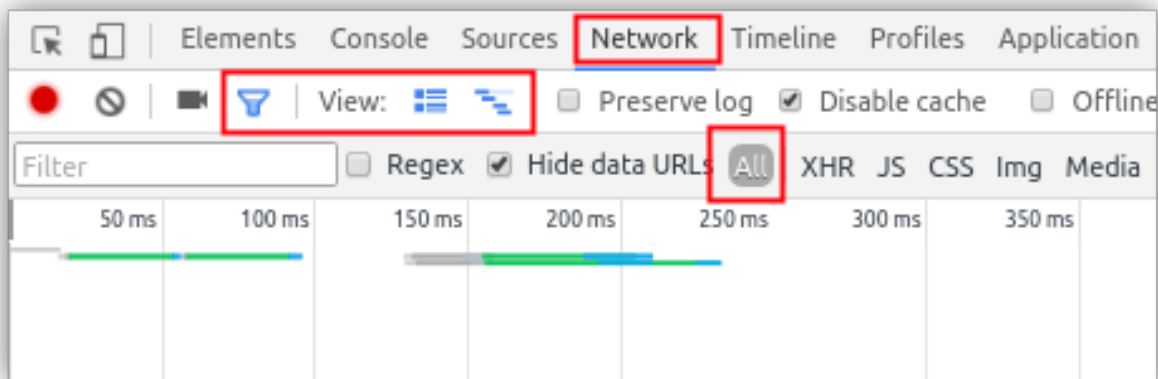
7.7 Page Request Evaluation

In this recipe, we go over one possible method of evaluating a page for the number of *HTTP requests* that it sends to the project domain when it loads.

7.7.1 Opening the DevTool

Investigation of the number of requests can be done on the original site. With the site open and selected in a tab, press **F12** to open the DevTool for that specific URL. By default, the DevTool will be docked within the browser screen, but you can also unlock it into a separate window.

1. Click on the Network tab.
2. Select "All" to track all requests.
3. Enable "Use large request rows" (to the right of "View" in the toolbar).
4. You can leave the "Hide data URLs" option on



Chrome

DevTool Settings

At this point, since you have opened the DevTool after the site has already loaded, the Network tab will be completely empty.

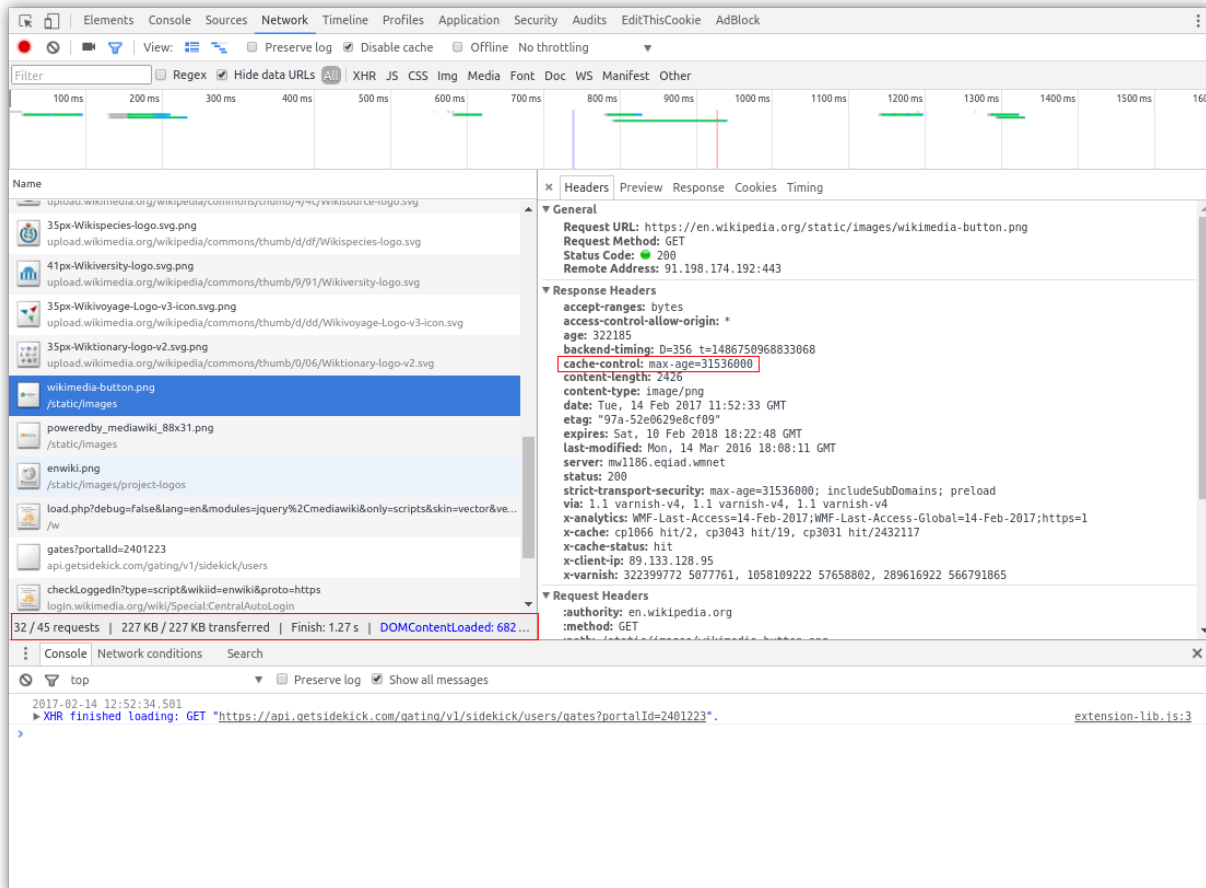
5. Refresh the site to start logging requests.

You will see a flurry of activity as the site reloads. After it has finished, you can start analyzing the various requests in the list.

7.7.2 Filter Requests on the same Domain

6. Enable "Regex" for filtering

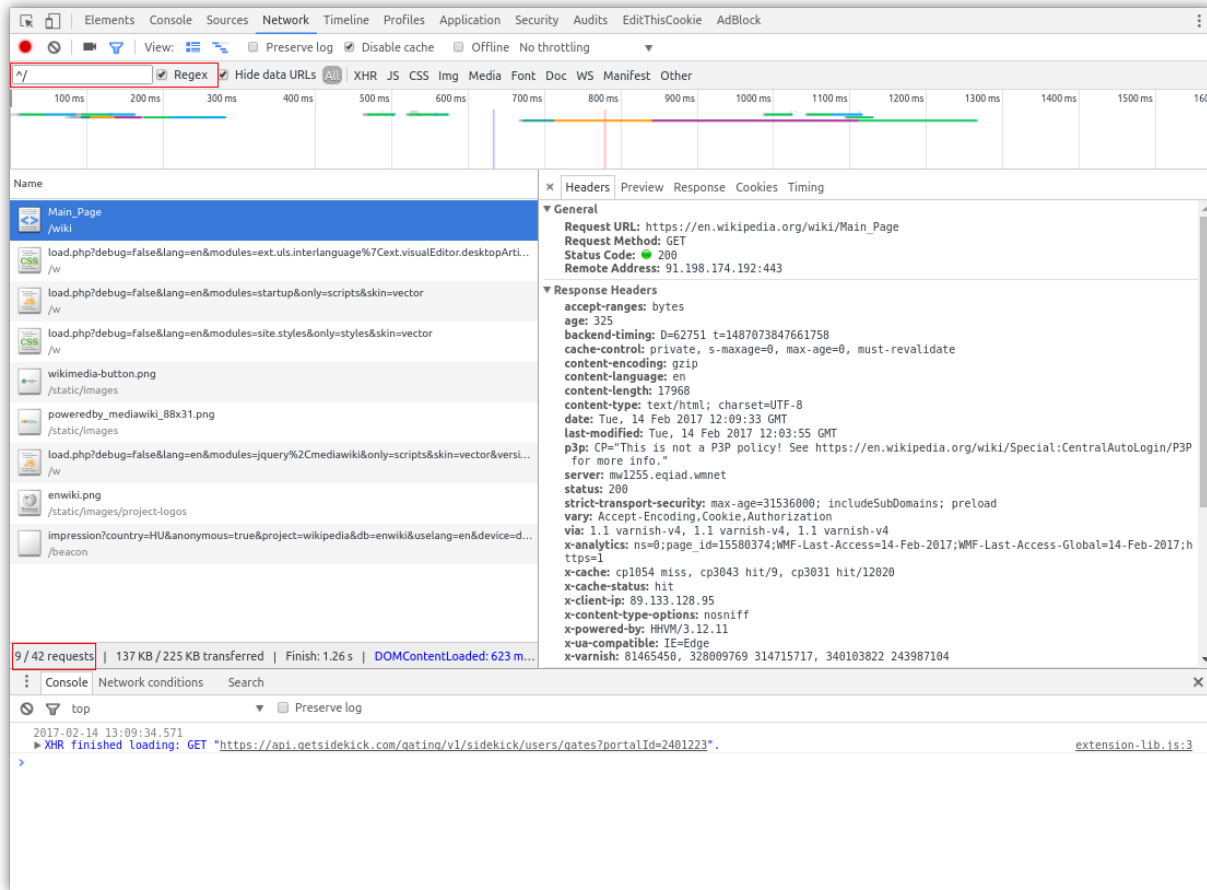
You are free to ignore all requests that go to external domains that will not be part of any linked projects. If you have enabled the “large request rows” option, the DevTool will helpfully list the paths below the resource names.



Request

Locations

The DevTool provides detailed information, but not all of it is needed in this case. As soon as you have the full list in view, you can remove from view those requests that go to external domains. The easiest way to do this is to add the “^/” regex to the filtering options on top.



Domain

Filter

At this point, the first relevant statistic becomes available at the bottom of the request list - the number of currently displayed requests / total number of requests will be a good first indication of the number of requests that *potentially* have to go through the proxy when the page is served.

7.7.3 Considerations

What Cache Headers are present?

If you click on a request in the list, a new sidebar will appear with information about that particular request. For evaluation purposes, the Header tab, and the `cache-control` directive is especially important.

`cache-control: private` or `cache-control: no-cache` indicate a request that the original server expressly states to be non-cacheable - usually, it is not a good idea to change this haphazardly. It is better to count these requests as necessary for the construction of the page.

Is the resource static?

PNG/JPG images, JS and CSS files are those resources that tend not to change rapidly. You can override the Cache Headers of such Resources in Dashboard > Path settings - with the effective result being that the burden of serving that content is offloaded to independent caching nodes on the global network.

Re-caching happens for each cached entity after the duration of the `max-age` directive passes. Using `max-age`, you declare a time-frame during which you will enlist the help of the network to serve the content in unchanged form. It can be used to fine-tune the time that you'll allow a specific cached instance of a resource to persist.

NOTE While Cache Header overrides work in the overwhelming majority of cases, there is no “law” to force caching nodes to respect them: consequently, the pace at which various Resources are cached/re-cached on the global network is, to a degree, arbitrary.

While technically possible, making Document resources cacheable requires careful consideration, but as little as 10 to 60 minutes can be very useful. Consider that in most cases, the landing page receives the most page requests. Consequently, allowing it to be cached with a controlled `max-age` value means considerable savings on the proxy (with the caveat that any changes will take at most the time declared for `max-age` to propagate across the network).

Does the requested resource change often/constantly according to context?

Dynamic Resource

XHRs/AJAX calls/dynamic content cannot be cached without rapidly running into many problems on the published site. It is better to say that they simply can not be cached. This also applies to those requests that are sent throughout the user session on the page after loading it (and in lieu of hard data, it is very difficult to forecast the number of dynamic requests a given user will start).

Salient examples are search field handler scripts, web-shop endpoints, PHP scripts, backend endpoints and other similar sources that give wildly varying responses based on the parameters sent in the requests.

Frequently Iterated Resource

If a site is undergoing development, for example, it is usually not a good idea to add Cache Header overrides (certainly not overly long ones). This would in effect delay propagation of any changes by the value of `max-age`, resulting in syncing problems between the original site and its translated counterparts.

7.7.4 Overriding Cache Headers

Evaluation of the number of requests is most useful when estimating the monthly cost associated with serving a site. For overriding cache headers on the Dashboard, see the [Path Settings](#) section of this documentation. Enable the public caching tweak in Dashboard 2.0 → *Advanced settings* to facilitate further speedups on the proxy side.

7.7.5 Example Scenario & Conclusions

If the following information is available:

```
* The original site has 50,000 monthly user visits
* A single target language sub-domain is expected to receive about half of that, 25,
  ↳ 000
* Each page uses between 50-70 requests to build
```

Using the consideration points above, it is determined that most of those requests can be counted out and the rest cached for 24 hours. The site will require 3 non-cacheable requests from a user coming in from a location where there is no caching node on the way.

From this you will be able to conclude the following:


```
* 2 requests are necessary, so 25,000 * 3 = 75.000 expected page requests
* BUT: consider also the number and visit frequency of revisitors - each time a
↳ cached entity's max-age (24 hours in this hypothetical scenario) expires, that
↳ resource has to be re-cached, which will increase the number of requests going
↳ through the proxy with a certain amount (although this amount is usually
↳ negligible).
```

With the appropriate Cache Headers, Google's geographically specific Edge Cache, the public internet, the various ISP caching nodes, and at the other end of the process, the user's browser cache will participate in offloading the page request from the proxy's translation pipeline.

7.7.6 Conclusion

Armed with this knowledge of an overarching view of requests that will have go through the proxy, you will be able to provide accurate estimates for the monthly costs of the proxy

7.8 Staging Domains

Use Staging Domains to change the origin server to use in Preview or crawls.

Website maintainers like to test any changes they make on a development or staging server before unleashing them on the Live site. The same staging server that is in place on the site can be used over the proxy as a testing ground for any translatable updates. Add a Staging Domain option to extract and use data from that domain staging server in the various proxy modes.

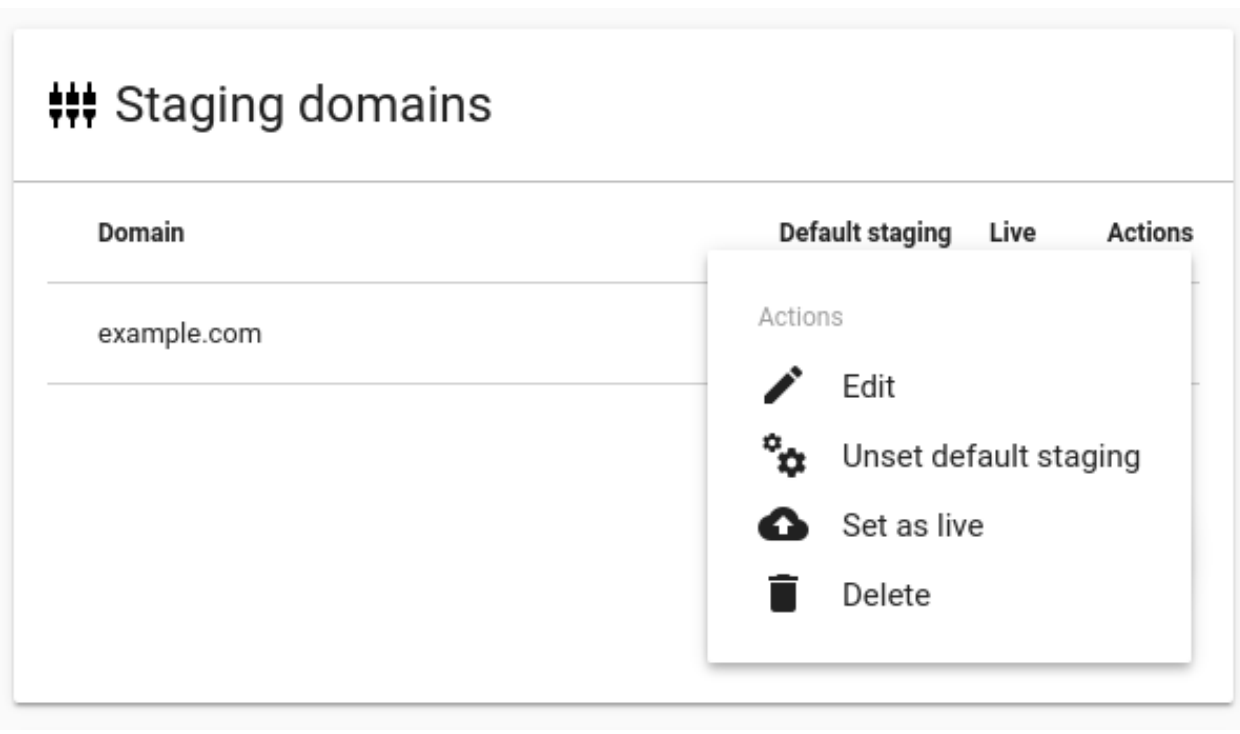
If the project URL is `example.com`, and a staging server exists at `dev.example.com`, you can enter that URL into the Staging Domain field and click on "Add Staging Domain". The domain is added to the menu below the text field automatically and enabled as default.

All requests going through the proxy (regardless of initiator, such as a user session in Preview or a content extraction Scan) will be mapped to that domain, regardless of the original project URL - the project domain might be `example.com`, but the Preview will be displaying content from `dev.example.com`.

Very important: Translations are propagated after extraction as usual, but this comes with an important warning: *previous translations* will only show up appropriately if the path structures of the original site and the staging server match 100%. If this is not the case, then a new project page is created in the project.

7.8.1 Default and Live Default Staging domains

Click the three-dot action button to reveal four options. *Edit*, *Set/Unset default staging*, *Set/Unset as live*, and *Delete*.



Staging

Domain Example

Click on “Make Live default” to enable the staging domain for the live published site. A tick icon will be displayed next to the staging domain when this option is enabled.

The same goes for *Default*, which applies to all other proxy modes (such as the Highlight View in the Workbench, the Preview domain or the X-proxy). Click on *Delete* to remove a Staging Domain.

Naturally, only one staging domain can be designated as Live or Preview default at one time, but it is possible to enable one domain to be both.

7.8.2 When to Use Staging Domains?

Content Decoupling

The staging domain feature of the proxy is beneficial when updates to the original site are regularly tested on the staging server first.

Staging domains let you make that same content available through the proxy for translation work without disturbing the translation quality over the published domain in the meantime.

The idea is to extract content from the staging domain and begin translation work early – by the time the changes on the original site are moved from the staging domain to the main one, all target language entries will have their translations ready.

Domain Name Changes

An alternative use of the staging domain feature is when the original site changes domains. As you know, project addresses cannot be changed once a project is created. However, migrating an entire project due to a simple name change might not be a very optimal solution.

In these cases, you can use the Staging Domain option to set up the new address as a staging server on the project. By following up on the name change in the Publishing settings, you can transform both the origin and the published domains to the new address (but note that the project address will remain the same, however).

7.9 Left-Right conversion

It depends on the site...

1. Best case scenario:

```
html {
    direction: rtl;
}
```

```
<html dir="rtl">
```

If it looks mostly OK, then all is left just some minor CSS fixes:

- flipping images (in carousel / slider)
- list elements' bullet is defined using a background image
- If text's align is defined explicitly (e.g. using WP's text editor) inline, then each and every element must be overridden using `!important`

1. Bootstrap and framework alike:

- some framework might provide a dedicated RTL css file [bootstrap-rtl.css](#)

1. If this is not case, each and every element must be positioned individually

7.9.1 Mixed content within text

- When it comes to actually render the text (numbers), the direction is determined by a couple of rules. Please read [this](#) for the details.
- ***As a rule of thumb: during the translation (from LTR language to RTL), don't use your CAT tool to change the order of the numbers and text where the translation is with latin characters. Phone numbers like 1-800-123-1234 should be left in this order.***
- To make sure numbers are rendered properly at the end, a Left-to-Right (LRM) must be inserted before every number. The dash between numbers split them, so LRM must be inserted after them again. [Click to read more on how to insert these LRM characters.](#)
- The same holds for parenthesis, etc. Make sure you understand the rules; sometime LRM must be inserted *after* the closing parenthesis

```
<p style="direction: rtl;">
<span>(TTY/TDD) 711 </span>
</p>
```

```
<p style="direction: rtl;">
<span>&lrn; (TTY/TDD) 711</span>
</p>
```

```
<p style="direction: rtl;">
<span>#8234; (TTY/TDD) 711#8236;</span>
</p>
```

- As an alternative, LRE character can be inserted *before* the sequence, terminated by a PDF mark.
- To edit the XML (XLIFF) directly, use [Sublime](#), available on Windows, OS X and Linux as well. It works very well with Regular Expressions. It's a robust way to insert these marks using their unicode symbol, such as `‎`

Further readings:

- http://dotancohen.com/howto/rtl_right_to_left.html

7.10 SSL Certificates

The proxy has the ability to proxy HTTPS pages, but to do so, it must be provided a certificate and private key matching the URL. Otherwise, it will be unable to identify itself as a valid server, and the browser will abort the connection for security reasons.

Our support team can assist in deploying an HTTPS site by providing a CSR (Certificate Signing Request) to generate the appropriate certificate if the required information is provided, or the client can prepare the certificate themselves.

Additionally, a certificate is required to provide a branded proxy instance.

7.10.1 The protocol

HTTPS (also called HTTP over TLS, HTTP over SSL, and HTTP Secure) is a protocol for secure communication over a computer network which is widely used on the Internet. HTTPS consists of communication over Hypertext Transfer Protocol (HTTP) within a connection encrypted by Transport Layer Security or its predecessor, Secure Sockets Layer. The main motivation for HTTPS is authentication of the visited website and protection of the privacy and integrity of the exchanged data.

In its popular deployment on the internet, HTTPS provides authentication of the website and associated web server with which one is communicating, which protects against man-in-the-middle attacks. Additionally, it provides bidirectional encryption of communications between a client and server, which protects against eavesdropping and tampering with and/or forging the contents of the communication. In practice, this provides a reasonable guarantee that one is communicating with precisely the website that one intended to communicate with (as opposed to an impostor), as well as ensuring that the contents of communications between the user and site cannot be read or forged by any third party.

Historically, HTTPS connections were primarily used for payment transactions on the World Wide Web, e-mail and for sensitive transactions in corporate information systems. In the late 2000s and early 2010s, HTTPS began to see widespread use for protecting page authenticity on all types of websites, securing accounts and keeping user communications, identity and web browsing private. (Courtesy of Wikipedia)

7.10.2 Issuing an SSL certificate

Issuing universally accepted certificates is restricted to the so-called “Root Certificate Authorities”. However, root authorities will generally delegate their powers to “Intermediate Authorities”, who will sign certificates as requested by the end user (provided ownership of the domain can be verified). When requesting a certificate for your white-label installation or HTTPS site, you will most likely interact with an intermediate authority, by providing them an encrypted configuration file (the Certificate Signing Request), which the provider consumes to produce a cryptographically signed certificate.

Using a CSR has several distinct advantages over generating your own certificate at the issuer: since the the translation proxy support crew is in control of the final product, we can tailor the request to generate the certificate we need from you; and since the private key remains safe with us, you do not need to take special precautions when sending it. However, we require some information to be provided to us beforehand:

1. `countryName_default = ${COUNTRY}`
2. `localityName_default = ${CITY}`
3. `streetAddress_default = ${ADDRESS}`
4. `postalCode_default = ${ZIP}`
5. `0.organizationName_default = ${COMPANY_NAME}`
6. `organizationalUnitName_default = ${ORG_UNIT}`

Issuing a certificate consists of several predetermined steps. First, a cryptographic key pair is generated, one half public, the other private. Then, a file indicating the domain the certificate is to be issued for, as well as various data about the entity holding the domain (generally, you or your client) is created. This file is then combined with the public half of the key, and signed with the private half. The resulting encrypted file is then handed to the issuer, who verifies the information contained within, and if successful, encodes the information into a public certificate, signing it with its own private key. This is the file that needs to be provided to our support team.

7.10.3 Providing us with a certificate

Following the previous phase, you are now in possession of a cryptographic certificate, and possibly its private key (if you elected to create your own certificate instead of requesting a CSR from us).

If we provided you with a CSR file, you need to send only the certificate. On its own, the certificate is not viable - it requires the private key to be useful, thus, it can be sent via email safely. The private key remains safe with us, and we will use it to upload the certificate to Google AppEngine, from where it will be available to authenticate the proxied site to the browser, and enable HTTPS for the translations.

If you elected to forgo the CSR and generate your own certificate, we will need the associated private key as well (and any passwords used to lock the private key). In this case, however, care must be taken to prevent the certificate falling into the wrong hands: the certificate and its keyfile, or the keyfile and its password must never travel together - if the email carrying both is intercepted, the malicious third party can use it to impersonate your site! Either use two separate emails, or even two different channels (email and Skype, for instance) to provide us the key and password and the certificate. Once we receive the files, we will again upload them to AppEngine after decrypting the keyfile, at which point it will be available for use with the proxied site.

N.B.: When selecting an SSL provider, bear in mind that Google only accepts certificates that are either self-signed or signed by a publicly trusted root Certificate Authority. One important point that needs to be highlighted is that the root CA for CloudFlare's Origin certificate system is *not* publicly trusted. Thus, we are unable to make use of Origin certificates generated via CloudFlare's system - in this case, please contact us for a CSR file for use with a provider of your choice.

7.10.4 SSL Manipulation Commands

Converting private keys to RSA

When uploading the keys into AppEngine, the file must be in RSA format. To verify, the beginning of the file should be

```
-----BEGIN RSA PRIVATE KEY----- .
```

If you read

-----BEGIN PRIVATE KEY-----,

you need to convert it, using the following command:

```
openssl rsa -in key -out key.rsa.key
```

Extracting Certificate and Private Key Files from a .pfx / PKCS#12 File (includes both the certificate and the private key)

- export the private key: `openssl pkcs12 -in certname.pfx -nocerts -out key.pem -nodes`
- export the certificate: `openssl pkcs12 -in certname.pfx -nokeys -out cert.pem`
- create RSA key / remove passphrase from the key: `openssl rsa -in key.pem -out server.key`

Check if a given key matches a certificate (or CSR)

By running these commands on the keyfile and certificate, you can verify that the key used to generate the certificate matches the one you have on hand. If the two outputs match, so do the keyfiles. But if not, your client may have used their own private keys to create the certificate, which you will have to obtain before forwarding it to us.

- `openssl rsa -noout -modulus -in privateKey.key | openssl md5`
- `openssl x509 -noout -modulus -in certificate.crt | openssl md5`

Alternatively, if you have the CSR as well, you can use the following command to obtain the checksum of the CSR's key and verify that the CSR you have on hand was used to generate the certificate.

- `openssl req -noout -modulus -in CSR.csr | openssl md5`

7.11 Proxy modes - X, P, live

7.11.1 X-proxy (testing, JS bugs, JS fix domains , CORS)

The X-proxy is great for testing. You can spot content that does not get picked up by default, and make your configurations to your project, and check for success.

There are a couple of situations, when the X-proxy comes in handy:

- Testing regular expressions, for example on e-commerce sites.
- Testing JSON (JavaScript) and XML translation.
- Just browsing through a site, for evaluation purposes.

An example X-proxy URL: `https://de-de-{project_code}-x.app.proxytranslation.com`

The X-proxy can be accessed from the pages list under Content (or Discovery) by clicking on the Preview button in the hover toolbar, while holding down the Ctrl/Cmd button, or you can just replace the -p for a -x in the normal preview's URL for the same effect.

7.11.2 P - Preview

The standard proxy mode to view the translated website before publishing. However, the preview can be used for a couple of other things:

- Cookie header extraction to get behind logins
- Visiting pages manually, to ingest content

An example Preview-proxy URL: `https://de-de-{project_code}-p.app.proxytranslation.com`

7.11.3 C - Channel

7.11.4 Live serving mode

After publishing the website, the proxy serves content on the chosen domain.

7.12 HTTP Headers

7.12.1 What are HTTP Headers?

HTTP header fields are components of the header section of request and response messages in the Hypertext Transfer Protocol (HTTP). They define the operating parameters of an HTTP transaction. The header fields are transmitted after the request or response line, which is the first line of a message. Header fields are colon-separated name-value pairs in clear-text string format, terminated by a carriage return (CR) and line feed (LF) character sequence. The end of the header section is indicated by an empty field, resulting in the transmission of two consecutive CR-LF pairs. ([Wikipedia](#))

7.12.2 Headers and the proxy

The Proxy strives to be as transparent when it comes to headers as possible. Therefore, we forward the majority of headers added to any incoming request, with a few exceptions where the presence of said header could cause undesirable operation in the original server.

Additionally, The Proxy also adds a few specialized headers both to requests to the remote server and responses to the client. The presence of these headers **SHOULD NOT** cause erroneous behavior in the server. Request headers contain additional information on the client viewing the site and the language being served. This can be used to provide customized content. Some Proxy-specific request headers are:

- `X-TranslationProxy-Translating-To: ja-JP`: This gives the language of the translated version the client is browsing.
- `X-TranslationProxy-Translating-Host: jp.eveonline.com`: This header contains the domain under which the Proxy is serving the translated site.
- `X-TranslationProxy-Originator-IP: 192.168.168.192`: If enabled, this header contains the IP address of the requester, which may be hidden by CDNs and other proxies.
- Future headers of the X-TranslationProxy containing other metadata
- `User-Agent`: This header is somewhat special, in the sense that it's not specific to The Proxy, rather, it is sent by almost all browsers to identify themselves. The reason it finds a place in this article is that it can be used to identify Proxy-served requests. Google AppEngine modifies this header when sending requests, in a way that ensure no further tampering before the header reaches the original server, by adding `AppEngine-Google`;

(+http://code.google.com/appengine; appid: {{application-name}}) - this can be used to whitelist the Proxy.

7.12.3 HTTP Headers and Security

Due to its nature as a proxy, and the fact that Google URLFetch uses a diverse range of addresses allocated randomly, it is easy to see that proxied requests may be caught by Web Application Firewalls, Anti-DDOS software, or even security provider companies. When a project is launched, it is often a good idea to contact the client and have them notify any contracted security providers, or make the necessary changes to firewalls and block-lists.

The easiest way to identify proxied requests is to read the `User-Agent` header, and locate the above-mentioned pattern - the application ID is added by Google at the last possible moment, thus, it can be a trusted indicator (along with the other headers, if needed) that the request was initiated by the Proxy. Security providers can be advised that the appearance of these headers is normal and should not be construed as an attack/phishing attempt.

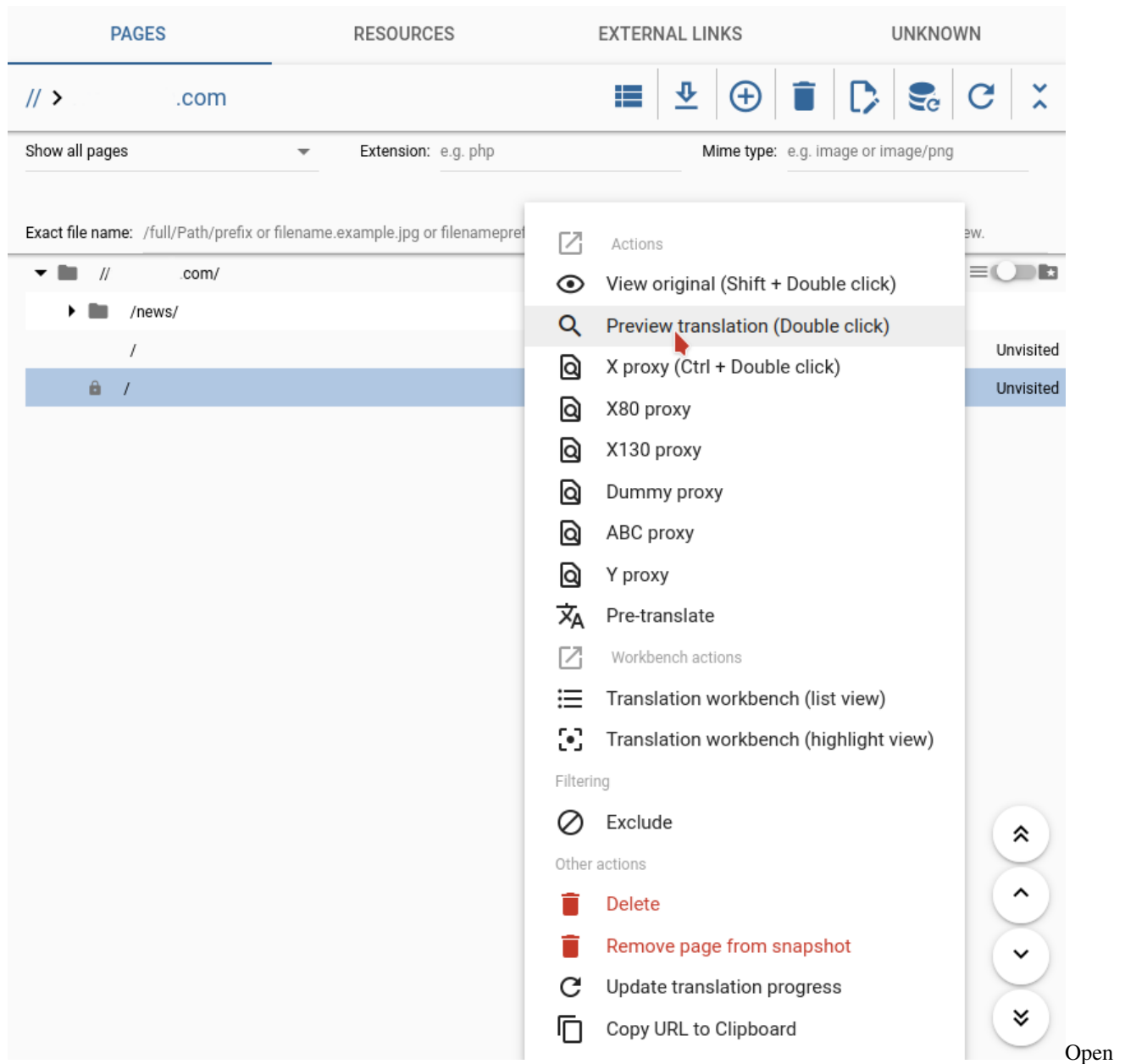
7.13 Secure Login - Passing Cookies to Scans

To scan content behind secure login, please follow this procedure:

1. Open the Dashboard 2.0 and navigate to the Pages list
2. Find the page with the login, right-click on it and select **Preview** (you'll need at least one target language on the project for **Preview** to be selectable!).

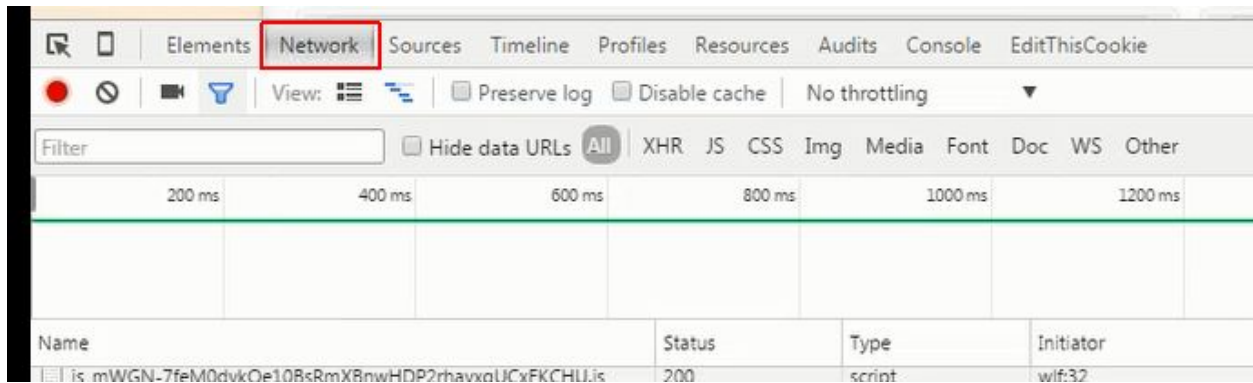
OR

1. Go to the **Preview** of the front page (the “/”, the first one on the Pages list). It will give you the front page through the proxy.



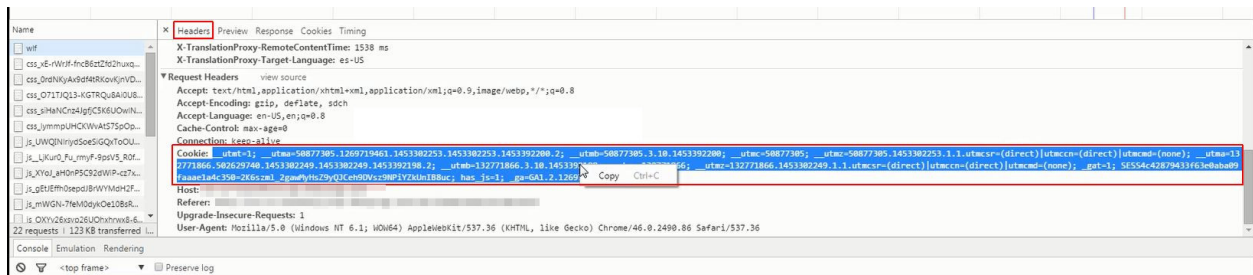
the Preview

1. Go to the address bar and type in the URL of the login-protected page.
2. Enter your login details.
3. Open your browser's DevTools from the Menu (F12 on Windows).
4. Go to **Network** and reload the page.



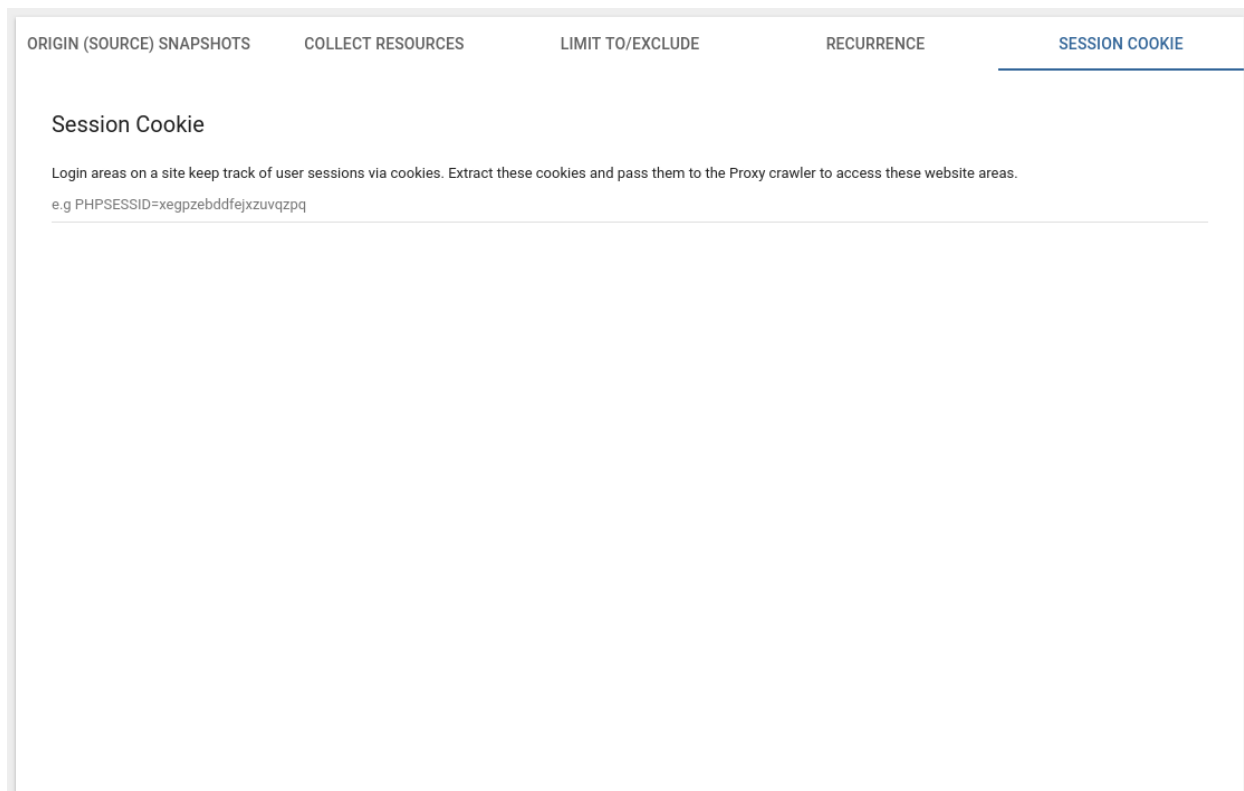
Getting the cookie

1. Scroll up to the first item and click on it.
2. Under headers scroll to the cookie header (among request headers), and copy the entire header.



Cookie header

1. Pass it to the Proxy: go back to your project and start a new crawl in the Crawl Wizard. Proceed as usual to step #4 (Fine-tune), then paste the contents of the cookie you just copied to the **Session Cookie** tab.



Passing

the cookie to the proxy

1. configure the rest of the crawl and launch it as usual.

7.14 Site Search

If a site includes a search function of any kind (ex.: product search in a webshop), visitors expect to be able to search for things in the language of the website – to input any term into the search fields and receive appropriate results immediately.

A pretty standard example

Let's say that a visitor uses the search field of `www.example.com` to search for the word “**product**”. Upon pressing enter, they are navigated to `https://www.example.com/search?q=product`. The server will detect and use the value of the `q` parameter and runs whatever server-side search mechanism it uses to search for this term, assemble a list of results and construct a result page to send to the browser.

Because the server is the one, who does the actual searching, we call this approach *server-side search*.

This introduces a potential issue

On a proxied site, however, we run up against a problem: the original server does not know about the translations. Requests are automatically relayed to the original server. If a visitor were to type “**produkt**” on the German domain (resulting in an URL navigation to `https://de.example.com/search?q=produkt`), that is the same as relaying a search query with German language content to the original site.

Not possessing a *German language index*, the response is certain to contain 0 results. The fact that the proxy is CMS-agnostic and that it generally doesn't require that translated content be shared with the original server also means that this same content will not be available to the indexing/search software that is running on the original.

Luckily, the server-side approach isn't the only option, when it comes to having some search functionality on a website!

Recommended solution

The fact, that proxied pages themselves are *publicly available for indexing by search engine bots* comes in handy. A search engine that supports site-specific queries can provide localized search via **client-side** AJAX requests.

There are two aspects to this kind of solution that you should consider:

On the one hand, such a site search has to be coded in JavaScript in the form of an override (an example with a detailed explanation follows below), and depending on how ornate/feature-laden the original's search functionality is, complexity of the override implementation can vary from the relatively simple to the astonishingly complex.

On the other hand, familiarity with the various indexing-related conditions of your chosen vendor is also important (we provide pointers to some documentation for Bing, since it is the vendor used in the example). In our experience, it is not at all unusual for search engines to act leisurely in their pace. Remember that a site can only be indexed after it is published over the proxy.

On the same note, the [publishing method](#) you use can also have a bearing on the way search & indexing will work, so a case-by-case analysis is a must. Note that the Client-Side Translation publishing method is *not compatible* with a site search integration of the type described here.

7.14.1 Third-party Integrations

Bing

Bing's Web Search API can be used for site search integration purposes over the proxy. In order to access Bing's API, you will need to purchase an API key. See the details on [Microsoft Azure](#) on procuring one, and see the [pricing page](#)

for a detailed description of the various service tiers.

Consult the [Webmaster Tools Documentation](#) concerning general usage-related matters. You can submit a domain for indexing in Bing [here](#). Submission of specific URLs is possible [here](#), but this feature is limited to root domains at the time of writing: until such time as this limitation is lifted, you might only be able to use this targeted indexing feature with subdirectory publishing.

Bing: Verification of subdomains

If you want to track the indexing of your subdomains in Webmaster Tools and do SEO tracking, you'll need to verify ownership of the target language subdomains. The available methods are [described here](#).

The XML-based approach is the simplest to implement over the proxy (it can be done without having to apply any changes on the origin): create a temporary [Page Content Override](#) with the contents of `BingSiteAuth.xml` (content type should be `text/xml`). This exposes the authorization XML over the proxy domains. From then on, you are free to add the target language subdomains in Webmaster Tools and verify them one-by-one.

7.14.2 Part II: Example

In the rest of this documentation page, let us detail how a simple site search integration could be implemented as injected JavaScript over the proxy using [v7 of the Microsoft Web Search API](#).

Page Modifiers

Customized JavaScript can be injected into the `<head>` tag of each page over the proxy. This capability is the entryway for a search override. To add injected JavaScript you'll find an editor in Page Modifiers > Javascript editor, where you can type or copy & paste Javascript code. After saving the modifier, it will show up in the page source over the proxy after a refresh (but note that cache settings might get in the way of an instantaneous update on the live domain!).

Site search functionality is frequently displayed to the visitor in the form of an input field and a button. We'll go with this familiar scenario and use the following minimal webpage for this tutorial:

```
<html>
<head>
  <script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>
</head>
<body>
  <script>
    function showResults() {
      document.querySelector("#results")
        .setAttribute("style", "display:block;");
    }
  </script>
  <div id="search">
    <form id="form" action="javascript:showResults();">
      <label for="#input">Search:</label>
      <input id="input" type="text"/>
      <button id="button" action="submit">Submit</button>
    </form>
    <div id="results" style="display:none;">
      <div class="result-item">
        <h2><a href="#">Result title</a></h2>
        <p>Result summary</p>
      </div>
    </div>
  </div>
</body>
</html>
```

(continues on next page)

(continued from previous page)

```
    </div>
  </div>
</div>
</body>
</html>
```

First steps

`showResults()` simulates the original search functionality of a site. The goal of our search integration is to prevent the original from running and replace it with a client-side dynamic request. A site search integration is no different from any other client-side code, and the skeleton that we described previously on the [Page Modifiers documentation page](#) is a safe starting point:

```
(function () {

  "use strict";

  $(document).ready(overrideSearch);

  function overrideSearch () {
    ...
  }

})();
```

We use jQuery's `$(document).ready()` to wait until the DOM is ready to be manipulated. The IIFE wrapper isolates our modifier from the global namespace, and we never leave the house without `use strict`.

We could define a callback on `document.onreadystatechange`, but this approach suffers from potential problems: we'd be setting our modifier on a publicly accessible property of `document`, exposing us to the possibility that some other script might inadvertently redefine it (or we could be the ones doing the redefining).

We would also end up having to pool different page modifiers in one function definition, which goes against the principle of separation of concerns. For these and similar reasons, vanilla JavaScript's `document.addEventListeners` or jQuery are a better fit.

With basic scaffolding in place, we move on to the nitty-gritty of search. The integration needs to do three things:

1. override search elements on the page
2. send search request based on user input and handle response
3. build the results page based on the response

We take each of these responsibilities in turn to see how they can be implemented.

Overriding Elements

Having prepared this basic scaffolding, we consider the contents of `overrideSearch()`. We would like to attach our event handlers to all search-related elements:

```
function overrideSearch () {
  $("#form").on("keydown", overrideInput);
  $("#button").on("click", overrideClick);
}
```

Next, we implement the callbacks. For the input field, we ensure that the user is not disturbed by the modifier needlessly, which only steps in if the Return key is pressed:

```
function overrideInput () {
  if (event.keyCode === 13) {
    event.preventDefault();
    sendRequest($("#input").val(), renderResult);
  }
}
```

The button is straightforward to override. The use of `this.previousSibling` is featured here as a suggested alternative to selectors.

```
function overrideClick(){
  event.preventDefault();
  sendRequest(this.previousSibling.value, renderResult);
}
```

We are done with the override part. The error message `Uncaught ReferenceError: sendRequest is not defined` should appear in the console if we try to search at this point. The default event might have been a navigation or a dynamic request of the original site's own, but we prevent it from executing and call the function `sendRequest` instead, which we'll implement next.

Sending the Request

We assemble a GET request using the search term passed to `sendRequest` and send it to the Bing endpoint. For example:

```
function sendRequest (term, callback) {
  $.ajax(createRequest(term)).success(function (resp) {
    callback(resp);
  })
}
```

`createRequest` is an important part of this process, where various API-related parameters to construct and return the request are sent. Using Bing, it could look something like this:

```
function createRequest (term, offset) {
  return {
    beforeSend: function (xhr) {
      xhr.setRequestHeader("Ocp-Apim-Subscription-Key", config.API_KEY);
    },
    error: function (xhr, error, thrown) {
      console.log("Error during request!");
    }
  },
  url: config.API_URL,
  type: "get",
  data: {
    q: "site:" + config.DOMAIN + " / " + term,
    count: 10,
    offset: offset || 1
  }
};
};
```

`beforeSend` sets the **Ocp-Apim-Subscription-Key** header, which contains the API key to authenticate the request. We introduce Bing's **site-specific search** feature into the request by prefixing the value of the `q` property with

“site:”(which works the same way as on [bing.com](https://www.bing.com/)).

It is also at this point that the code should handle the publishing method you use. If you publish your project in a subdirectory of the original domain, then in addition to the “site:” prefix, you would also have to add the target-language directory prefix (e.g “/ja/” or “/de/”). If you are publishing in multiple languages, you can use the `lang` attribute of the `html` tag (the value of which is always the current locale) to make this part of your code target-language specific.

A variable called `config` was also introduced to hold search-related config parameters in one place. This variable can go to the top of the modifier:

```
var settings = {
  API_KEY: "nmtcxylkj56lkjmnj3mg782nmvf23gz", // example only!
  API_URL: "https://api.cognitive.microsoft.com/bing/v7.0/search",
  DOMAIN: location.host
}
```

If the API key is valid and version-compatible with the endpoint, Bing responds with a JSON to each request, details of which can be found [here](#). The `webPages.value` property of this response is particularly important, a JSON array containing the first batch of search results.

Note that `settings.DOMAIN` is set to `location.host`. Encoded in this fact is the assumption that `location.host` (the site in which the code is running) is to be used for searches and it is already indexed.

During development and testing, however, no proper search result might be available yet. For dev purposes, it can be useful to change `DOMAIN` to e.g `en.wikipedia.org` temporarily to receive proper search results. The use of a mock responses is also an option.

Search engine APIs have many features that you can provide in your implementation beyond this example, but we will not discuss all the possibilities in this tutorial.

We pass the response to the final component of our override, the DOM handler, `renderResult`. We already mentioned it in passing to the `sendRequest` calls.

Building the Results

Astute readers will notice that there is hardly any site-specific information in the code above (practically none, except the element selectors). We’d be inclined to think that plugging a `form` and `button` selector into the functions and changing the `API_KEY` will get us on our way on any similar site – and to the extent of element overrides and request handling, that might very well be the case. `renderResult`, however, does not enjoy the benefit of generality.

The DOM handler is responsible for *displaying* search results on the proxied site in a way that seamlessly integrates with the original appearance, making this part of a site search integration heavily dependent on the specific context of the website that we are dealing with.

In order to work out the implementation for our example webpage, we turn our attention from `#search` to `#results` and research the structure of a search result item.

In the same breath, we can add the basic structure of `renderResult` right away. The bare minimum functionality is to clear the result list and add the one that was received and passed as argument.

```
function renderResult(response) {
  $("#results").empty();
  if (typeof response.webPages.value !== "undefined") {
    for (var hit in data.webPages.value)
      $("#results").append(createResultItem(data.webPages.value[hit]))
  }
}
```

`createResultItem` constructs an element for a result item using information provided by Bing, but copying the element structure from the original, which has a huge benefit: existing CSS styles will apply to the search integrated results automatically. Inspecting the search results produced by the original server, we learn that a search result item looks like this:

```
<div class="result-item">
  <h2><a href="#">Result title</a></h2>
  <p>Result summary</p>
</div>
```

Knowing this, we can implement `createResultItem` in the following simple fashion:

```
function createResultItem (result) {
  return $("

").attr("class", "result-item")
    .append($("<h2>").attr("href", result.url).text(result.name))
    .append($("<p>").text(result.snippet))
};


```

Among the available options for creating DOM elements (`document.createElement`, string concatenation, etc.), the jQuery-based approach is clear and concise. Many bells and whistles can be added to a DOM handler, such as a pager, for example, which would require that we handle offsets (only hinted at by `createRequest` in this tutorial) and add Previous/Next buttons, etc.

Our tutorial, however, does not extend to those details and we conclude it here.

Notes

Some frequent, but non-essential complexity of client-side search is omitted from our example:

- offsets (a Bing-specific term) are only alluded to. Any search API will support requests for the next batch of search results for the same search query. This is usually exposed to the end user in the form of a pager that, when clicked, sends the same request but increments the `offset` by 10. Such a feature causes both `renderResult` to manage the concept of a pager and `sendRequest/createRequest` to have to keep tabs on the current `offset`.
- a search field is also generally available on all pages on a site, not just the search results page. This means that an integration needs to ensure that the user is redirected to the search results page and that query parameters are appropriately handled. Besides using the values of the input field (a user-driven query after page load), a search integration usually has to be able to extract `produkt` from `https://de.example.com/search?q=produkt` to start a default search on the search results page when it loads for the first time. This is usually not difficult, but as we've said previously, much depends on specific circumstance.
- a search integration also has to be prepared to display a "No results." search result page if the search engine returns no hits. Such natural language has to be exposed in a way that can be translated using the proxy.

Code

We repeat the code from the discussion above in its entirety. In summary, when injected into the example webpage, it will override both the search field and the button to request search results from Bing via an AJAX call and then displays those search results in-place.

```
(function () {
  "use strict";
```

(continues on next page)

(continued from previous page)

```

var config = {
  API_KEY: "nmtcxylkj56lkjmnjn3mg782nmvf23gz", // example only!
  API_URL: "https://api.cognitive.microsoft.com/bing/v7.0/search",
  DOMAIN: location.host
};

$(document).ready(overrideSearch);

// OVERRIDE
function overrideSearch () {
  $("#form").on("keydown", overrideInput);
  $("#button").on("click", overrideInput);
};

function overrideInput () {
  if (event.keyCode === 13) {
    event.preventDefault();
    sendRequest($("#input").val(), renderResult);
  }
};

function overrideClick(){
  event.preventDefault();
  sendRequest(this.previousSibling.value, renderResult);
};

// REQUEST
function sendRequest (term, callback) {
  $.ajax(createRequest(term)).success(function (resp) {
    callback(resp);
  })
};

function createRequest (term, offset) {
  return {
    beforeSend: function (xhr) {
      xhr.setRequestHeader("Ocp-Apim-Subscription-Key", config.API_KEY);
    },
    error: function (xhr, error, thrown) {
      console.log("Error during request!");
    },
    url: config.API_URL,
    type: "get",
    data: {
      q: "site:" + config.DOMAIN + "/" + term,
      count: 10,
      offset: offset || 1
    }
  };
};

// DOM
function renderResult(response) {
  $("#results").show().empty();
  if (typeof response.webPages.value !== "undefined") {
    for (var hit in response.webPages.value)

```

(continues on next page)

(continued from previous page)

```
        $("#results").append(createResultItem(response.webPages.value[hit]))
    }
};

function createResultItem (result) {
    return $("

").attr("class", "result-item")
        .append($("<h2>")
            .append($("<a>").attr("href", result.url).text(result.name)))
        .append($("<p>").text(result.snippet))
};

})();


```

8.1 Architecture

8.1.1 Modularity

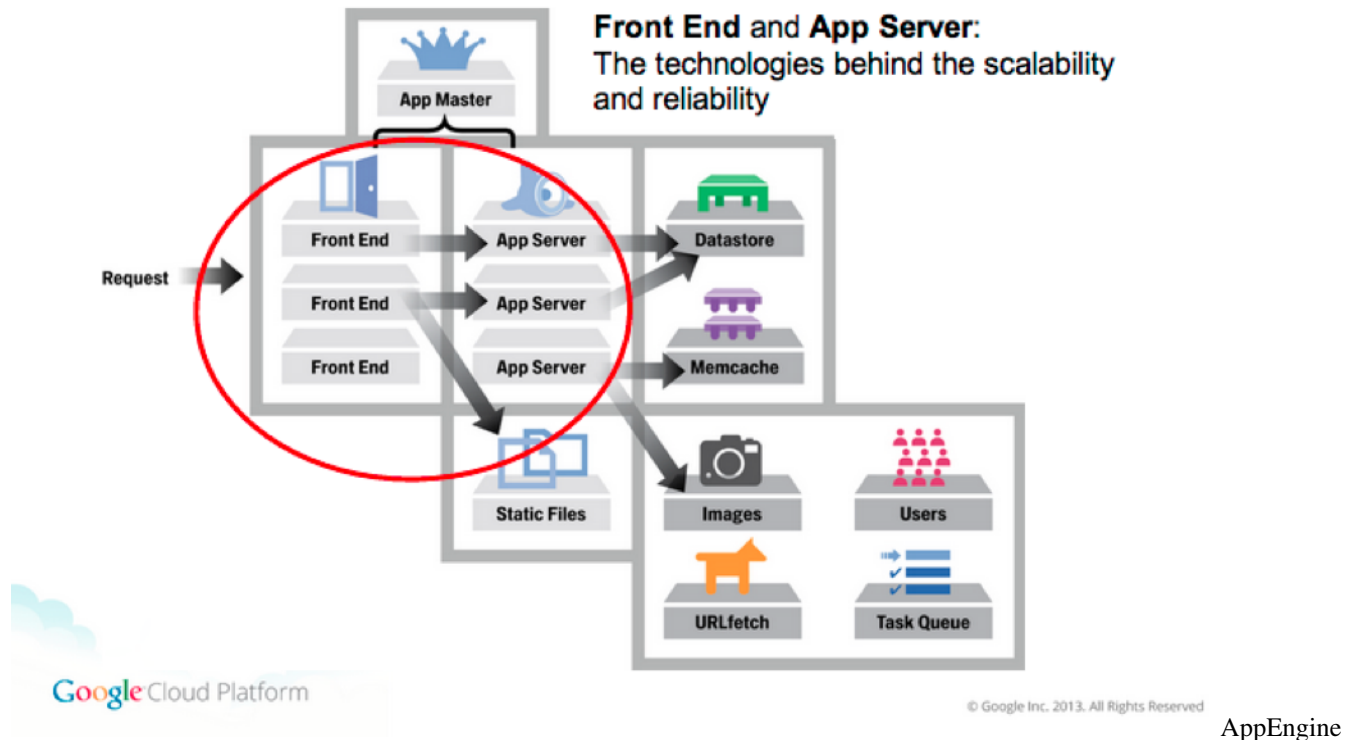
The Translation Proxy is based over Google's AppEngine infrastructure, split into frontend and backend modules. Each module encompasses variable numbers of instances, scaling automatically in response to demand. Modules are versioned and deployed separately, and can be switched independently, if needed.

Frontend instances serve requests from visitors to translated pages (in addition to serving the Dashboard and providing user-facing functionality). Requests are routed to the Proxy Application via the CNAME records created during the publishing process.

Backend modules are responsible for billing, statistics aggregation, and handling potentially long-running tasks, like XML import-export. Backend instances are not directly addressable, and provide no user-facing interaction.

8.1.2 Underlying technologies

Immediately underlying the Proxy Application is the AppEngine infrastructure, responsible for rapidly scaling the deployed application. AppEngine also handles communication with the Google Cloud Datastore, a high-replication NoSQL database acting as the main persistent storage; as well as the Google Cloud Storage system, an also-distributed long-term storage. Logging is provided by Google Cloud Logging, while BigQuery provides rapid search ability in the saved logs on request.



Architecture

Encompassing the entire application is the Google EdgeCache network, proactively caching content in various data centers located regionally to the request originator. Any content bearing the appropriate headers (`Cache-control:public; max-age=/\d+/` and `Pragma:public` - both are required) is cached by the EdgeCache for as long as needed, for requests originating in the same geographic area.

The current instance of the Proxy Application is hosted in the US central region, as a multi-tenant application (serving multiple users, but dedicated to proxying). However, single-tenant deployments (dedicated to a single user), special deployments to other regions (EU or Asia), or internal systems fulfilling the AppScale system requirements can be discussed on a per-request basis.

8.2 Request Handling

In the Translation Proxy, frontend instances are responsible for serving translated pages. Thanks to AppEngine's quick-reaction scaling, the number of frontend instances aggressively follows (and somewhat predicts) demand, keeping latency low. The general life cycle of a proxy request can be described as follows.

- The incoming requests, based on the domain name, reach the Google Cloud (rerouted via DNS record CNAME, pointing to `ghs.domainverify.net`).
- Based on the domain name and the deployed Proxy application, AppEngine decides that this specific request should be routed to the Proxy AppEngine deployment.
- The request reaches the Proxy Application internally; the application does a lookup against the domain for the associated project. There are special domain names, and the final serving domain, for which caching is activated.
- Based on the URL, the Proxy application determines the matching Page in the Proxy database. The database has a list of segments, pointing to our internal Translation Memory (TM). We retrieve all these existing Database entries, including the translations for the given target language.

- The Proxy application processes the incoming URL request, and transforms it to point back to the original site's domain. Then, the source content of the translation is sourced, according to cache settings in effect on the project.
 - If source caching is *disabled*, the application issues a request, and retrieves the result from the original web server, which is hosting the original website language.
 - If source caching is *enabled*, a local copy (a previously stored version of the source HTML) is used, instead of issuing a request to the original web server.
- Depending on the `Content-type` of the response, the appropriate `Translator` is selected, and the response is passed to an instance of the `Translator` as a document. The behavior of the `Translator` can be affected by cache settings as well.
 - If binary caching is *disabled*, the application then builds the Document Object Model (DOM) tree of the result, finally iterates through all the block level elements, and matches them against the segments loaded from the database. If there's a match, we replace the text with the translation. If not, we 'report' it as a missing translation.
 - If binary caching is *enabled* and the hash of the source HTML *matches* the one stored in the cache, a previously prepared and stored translated HTML is served.
 - If binary caching and keep cache are *both enabled*, and the hash of the source HTML *doesn't match* the one stored in the cache, the proxy translates the page using the TM. If the number of translated segments is higher than the previously prepared and stored translated HTML, the new version is served; otherwise the old one. (Keep cache can be thought of as a "poor man's staging server").
- Hyperlinks in the translated content are altered by the `LinkMapper` to point to the proxied domain instead of the original. This affects all `href` or `src` attributes in the document equally, unless the element is given the `__ptNoRemap` class. At this point, resources may be replaced by their localized counterparts on a string-replacement basis.
- The application serializes the translated DOM tree, and writes it to the response object.
- Before final transmission takes place, the Proxy may rewrite or add any additional headers, such as `Cache-control` or `Pragma`.
- Finally, the Proxy serves the document as an HTML5 stream, as a response to the original request. AppEngine *must* close the connection once the response is transmitted, so proxying streaming services is not possible in this fashion!

8.3 Classification of Content

The Translation Proxy distinguishes two main types of content: text content and resources. The key difference is that text content may be translated, while resource content is treated opaquely, and only references can be replaced as resource localization. It is possible to reclassify entities from Resource to Text, but not the other way around.

During proxying, resources are enumerated, and any already-localized references are replaced, while text content is passed to an applicable `Translator` implementation for segmented translation.

8.3.1 Text content

By default, the Proxy only handles responses with `Content-Type:text/html` as translatable. To process HTML content, the source response's content is parsed into a `Document`, then text content is extracted from the DOM-nodes. Additionally, various attribute values are processed (without additional configuration, `title` and `alt`).

The content is then transformed into `SourceEntry` entities server-side. Each block element comprises one source entry, with a globally unique key. If segmentation is enabled on the project, the appropriate rules are loaded (either using the default segmentation or by loading the custom SRX file attached to the project), and the content is segmented accordingly, with the resulting token bounds stored in the `SourceEntry`. Along with the `SourceEntry` entities, the corresponding `TargetEntry` and `SourceEntryTarget` entities are created. `TargetEntry` entities, as the name suggests, hold the translations; `SourceEntryTargets` act as the bridge between the two, and hold the segment status indicators for both.

The content of source entries is analyzed in the context of the project, and statistics are computed. These statistics include the amount of repeated content at different confidence levels based on the similarity of the segment - The Translation Proxy differentiates five levels of similarity:

1. 102%: Strong contextual matches: every segment in the block level element (~paragraph) is a 101% match, where all the tags are identical. These matches do not result in the creation of new `SourceEntry` entities, thus changes in one place are propagated instantly to all occurrences.
2. 101%: Contextual matches: both tags in the segment, and contexts (segments immediately before and after) match.
3. 100%: Regular matches: the segment is repeated exactly, including all tags.
4. 99%: Strong fuzzy matches: tags from the ends are stripped out, words lowercased, numbers ignored.
5. 98%: Weak fuzzy matches: all tags are stripped out (may have to be adjusted manually afterwards!), words lowercased, numbers ignored. If the Proxy cannot match the tags between the translation and the source due to excessive differences, all tags are placed at the end of the segment, requiring manual review!

These classifications are reused during memory-powered pre-translation in order to select the best applicable translation or to propagate existing translations.

8.3.2 Resource content

By default, any content with content types other than `text/html` are treated as a resource, and is not a candidate for translation, only replacement en bloc. This mainly includes `application/javascript`, `text/xml`, and various `image/*` content types. Every resource can be given different replacements per target language, and if required, certain resources (`application/javascript` and `text/xml`) can be made translatable after pre-configuration is done. In this case, instead of references being replaced, the appropriate `Translator` will be instantiated and the content passed to it. This can enable partial or complete translation of dynamic content transmitted as JSON or XML.

8.4 Translation Memories

The Translation Proxy can be configured to maintain and leverage internal translation memories. These memories can contain more than one target locale allowing leveraging them for any pair of locales contained within.

As opposed to project dictionaries, translation memories are keyed to the *user* creating them, and can be assigned to any project the user has Backup Owner privileges or higher. Any project can contain an arbitrary number of memories, but one must always be designated the default: only this memory will be utilized when segments are being written; while pre-translation and suggestions are fed from *all* memories assigned to the project with applicable locale configurations.

8.5 Using TMs

Translation memories are initialized empty, and must be first configured with locales. After the target locales are defined, the memory can be populated. There are three ways a segment can be injected into the memory:

- **TMX-import:** The Proxy can digest a standard TMX (Translation Memory eXchange) file and populate a designated memory based on its contents. The memory must be configured with at least one of the target locales of the TMX file. Duplicate segments are either merged (if for different locales) or discarded during import.
- **Project population:** The Proxy can populate the memory from the project it is currently assigned to. The memory must be configured with at least one of the *project's* target locales for this to work. If there are several locales assigned to the memory, the UI will treat them as a set, and offer the *intersection* of the memory and the project's locales as the default. This set can be further restricted by removing locales from the population task before committing it. This action is logged in the project's Audit Log.
- **Individual Injection:** If a memory is assigned to the project with at least one locale present on both, it will be available on the Workbench for use. Confirming one or more segments will trigger the `saveToMemory` action, injecting the segment in its current form into the memory.

Memories are used for two tasks on the UI:

- Pre-translation tasks can leverage any memories assigned to the project, provided the memory is configured with the correct locale. This applies to user-triggered Pre-translation, as well as Automatic Pre-translation triggered by new content. Only content with confidence levels above the user-configured threshold will be used, matches with lower percentages are discarded.
- The Workbench automatically leverages any memories with the appropriate locales on segment selection. Matches are displayed in the Suggestions tab of the sidebar, along with their match percentages. Additionally, all memories on the project with the applicable target segments can be queried at will by entering a search term.

8.5.1 Confidence levels

The Proxy differentiates five levels of similarity between individual segments/entries ([see here](#)). Memory application yields the best results between 101% and 99% - 98% matches disregard tagging, and may need manual adjustment. However, searching below 98% is also possible, using the Google Search API, but these matches should be used with caution, as there is no guarantee regarding their accuracy due to the Search API's word stemming.

8.6 Page modifiers

Due to the way the Proxy Application operates, it becomes fairly easy to modify the pages as they are being served. Because the datastream must pass through the proxy to have the translation embedded, the Proxy Application can insert JavaScript modifiers, modify style sheets, and even embed entire pages that do not exist on the original.

- **CSS Editor:** the Proxy Application can be used to insert locale-specific CSS rules into the site being served. The rules are inserted as the last element of the `head` on every page served through the proxy. The most common use of this feature is to alter the writing direction for non-Latin scripts, such as Arabic.
- **JavaScript Editor:** the JavaScript edited here is inserted into the `head` element of every page being served through the Proxy Application. As the last element of the `head`, it has access to any global variables added by scripts before it.
- **Content Override:** the Proxy Application can create a “virtual” page in the site or override an existing one with custom code. For any requests to an overridden page, the corresponding remote server request is not sent, and the override contents are used as the basis of the translation. The source is not required to be HTML, custom content-types can be entered, along with customized cache headers, and status codes (HTTP status codes are restricted to those permitted by the [Java Servlet class](#)!) - note that the 300-family of status codes requires the `Location` header to be defined as well.

Both the CSS and JavaScript injectors can use already-existing files for injection instead of copied content. The injected files must be handled by the project in some way (either by being in the project domain, or in the domain of

a linked project), or be created by a content override. The order of definition for these entries matters, as they will be inserted into the document in the order they are displayed on the UI, which may cause dependency or concurrency issues!

8.7 Client-Side Translator

8.7.1 Operations

Overview

The *Client-Side Translator*, codenamed *CREST*, is an alternative publishing mode. Instead of operating in proxy mode, the system generates a Javascript stub that needs to be referenced in the site, and it will translate the page in real time using a dictionary downloaded from the cloud service. Language choice is persisted in the browser's *Local Storage*, enabling automatic translation of any page in the site instantly on landing.

Setup

Content is collected and translated the same way as normal. Once publishing is needed, content is exported by selecting the *Client-side translation* file format, then publishing the latest export (or the one selected for production) from the *Previous Exports* screen and clicking the context menu.

The translation loader script can be inserted with a one-liner script element, which is available from the *Global Settings* screen of the *Publish* section in the sidebar. The website owner needs to insert this script element into pages requiring translation.



JS Translation (CREST)

CREST is the JavaScript-based client-side translation. Not to be confused with CRESBL, the Chrome extension.

Embed code

```
<script type="text/javascript" src="https://app.localhost.com/client/1234567890/stub.js"></script>
```

COPY TO CLIPBOARD

Copy from the Dashboard and paste into the head

Once complete, the translations can be requested by adding a query parameter to the URL, with the name `__ptLanguage` and the chosen locale as the value (for example `https://example.com/path/to/page?__ptLanguage=ja-JP`).

8.7.2 Integrators' Guide

Elements

CREST is controlled by the loader script, inserted into every page requiring translation. The script element should be inserted as high in the `head` as possible in order to begin translation at the earliest possible point. The loader script has a number of query parameters that may be used to manipulate its operation. Any number of these can be combined to customize the loader's behavior from the default settings (existence of said defaults also means *none* of these parameters are mandatory to supply).

- `languageParameter`: This parameter can be used to change the language selector key from its default of `__ptLanguage`.
- `storageParameter`: This parameter can be used to change the `LocalStorage` key used to store a previous selection from its default of `ptLanguage`.

- `noXDefault`: if set to `true`, suppress placing an `x-default` link element in the head if a translated language is loaded. This may have SEO implications!
- `rewriteUrl`: if set to `true`, use `history.replaceState` to rewrite the URL shown to the user so that it always displays the selected language.
- `scriptUrlIsBase`: if set to `true`, the loader will search for the translator script based on its own URL. **CAUTION: This is not supported under Internet Explorer!**
- `disableSelector`: if set to `true`, the stub will not inject its own language selector in the sidebar. In this case, it is up to the website to provide links to the various language versions.

Language selection is possible via the sidebar inserted on the right by default, or custom `a` elements that manipulate the value of the `__ptLanguage` query parameter. Note that once a language is selected, the choice is persisted into the browser *Local Storage*, so further links need not be annotated with the query to maintain translation.

On selecting a language, the loader script will insert a new script element referencing the exported dictionary. It downloads the translations necessary for display and the translator algorithm that processes the available DOM to replace content with the available translations. The translator will also attach a `MutationObserver` to the document being displayed that allows it to react to DOM manipulation or newly-appearing elements in real time.

Interop

In order to provide a seamless user experience, *CREST* exposes a number of events at key points in the process that allow the containing page to react to the translation process and take action to enhance the experience. The following events are dispatched at various points:

- `crestDictionaryLoadingStart`: Dispatched when a language is selected and download of the corresponding dictionary begins. As the dictionary can be sizeable, this event can be used to display a notification to the visitor advising them that the language is about to change.
- `crestDictionaryLoadingEnd`: Dispatched on completion of the dictionary download. Firing this event means translations are available, and they will be applied to the DOM momentarily. If a notification was displayed on download start, it should be removed on this event.
- `crestDocumentTranslationStart`: Dispatched when the initial translation of the document begins. Firing this event means translations are currently being applied to the entire page, and the displayed language is about to change. In case translation takes significant time, the user may benefit from an overlay or other message notifying them of the process and that the displayed language will change soon.
- `crestDocumentTranslationEnd`: Dispatched when the initial translation of the DOM is complete, and all available content has been replaced. At this point, the page is translated to the best of *CREST*'s abilities, and if an overlay or notification was displayed, it should be removed.
- `crestMutationTranslationStart`: Dispatched when a mutation of the DOM is detected by the attached observer and translation of the new/changed elements begins. This event is unique in that it includes a payload, an array of the `MutationRecords` that are being processed. These include information about the element name, DOM path, and other data that may be used by the page to react to changes.
- `crestMutationTranslationEnd`: Dispatched when the mutation observer completes its run and designates all mutated elements translated. If a notification was displayed on the preceding event, it should be removed now.

Example

```
<script type="application/javascript">
  document.addEventListener("crestDocumentTranslationStart", () => console.log(
↪ "Document translation started"));
  document.addEventListener("crestDocumentTranslationEnd", () => console.log(
↪ "Document translation ended"));
  // e.detail.targets contains the array of MutationRecord objects that are being_
↪ processed in the current run. For more information, see https://developer.mozilla.org/en-US/docs/Web/API/MutationRecord
  document.addEventListener("crestMutationTranslationStart", (e) => console.
↪ log(`Mutation translation started. Mutated records: ${e.detail.targets}`));
  document.addEventListener("crestMutationTranslationEnd", () => console.log(
↪ "Mutation translation ended"));
  document.addEventListener("crestDictionaryLoadingStart", () => console.log(
↪ "Dictionary download started"));
  document.addEventListener("crestDictionaryLoadingEnd", () => console.log(
↪ "Dictionary download ended"));

  console.log("Event listeners ready...");
</script>
```

9.1 Contacting Support

Website translation can be a complex task, even with the help of a piece of software like the proxy. Finding the root of the problem can be equally complex for those of us on support duty. Therefore, we would like to give you a few pointers on what information to supply if you decide to contact us.

This first thing we need is the **project code**. This eight-character unique string uniquely identifies the project. As you can see in the screenshot below, the project code is located in the Dashboard 2.0 → Project overview.

The screenshot displays the 'Project overview' page. In the left sidebar, the 'Project overview' link is highlighted with a red box. The main content area is titled 'Project info' and contains the following details:

- Domain: www.example.com
- Project code: 6i7qzgxa (highlighted with a red box)
- Source language: English (United States) [en-US]
- Project created: September 11, 2018. 16:02:24
- Alias: Project displayname

Below the 'Project info' section is a 'Crawl info' section. The sidebar also lists 'CRAWL' (Wizard, List, Recurring) and 'CONTENT' (Work package list, Work package wizard, Delete entries) sections.

code

We also need a **thorough description of the problem**. Screenshots are tremendously helpful, especially if you have layout issues on the translated site. If you're running into translation issues, please give an example segment along with the page link it can be found on.

If you have issues with importing XLIFF or TMX files, please attach them so that we can take a look. If you have questions about statistics, reports, or crawl logs, attaching or linking them in your query will considerably speed things up for us.

The information you provide will help us uncover the root cause of the problem. This often requires a bit of "detective work" on the original site, so we ask for of your patience while we figure things out. Someone from the support team will respond shortly with a solution, a request for more information, or simply an update.

9.2 Issues

9.2.1 XLIFF import error

When you import your translated XLIFF file back to the proxy, you will receive an e-mail notification when the process is ready. This mail contains the URL of the import log, and an overview of the log entries:

- Error:
- Warning:
- Info:

If you see other than '**Error: 0**' in your notification mail, the XLIFF file needs fixing. Usually these are tag placement errors that can be easily fixed in a text editor like Notepad++ or Sublime (the ones that have syntax highlighting, to make this fixing job easier), yet they do need attention, as the corresponding translation will not show up on the website.

- Open the XLIFF in your text editor
- Open the log file and check the error message(s)
- Do the necessary corrections in your XLIFF (see 'Troubleshooting')
- Save & upload the corrected XLIFF

In very serious cases the import might fail completely, but this is very rare. These cases include: attempt to upload an XLIFF related to another project, XLIFF with target language that doesn't exist in the project, and fully invalid XML in the file. In most cases the file is imported, only the faulty entries are omitted.

Please note that you need XLIFF files. Ideally, the export format of the CAT-tool should be the same as the import format, and as you import an XLIFF file for translation, the output should also be a standard XLIFF file. However, some versions of Studio tend to create an **SDLXLIFF** file upon exporting the translation. In this case, simply use the "**Finalize**" batch task or open the document in the **Editor**, press **SHIFT+F12** and select the target file location. This will create the XLIFF file for you (instead of SDLXLIFF).

You might also need to disable segment info storage in Studio (*Options -> File Types -> XLIFF -> Settings -> 'Do not store segmentation information in the translated file'* should be checked). This may require creating a new project.

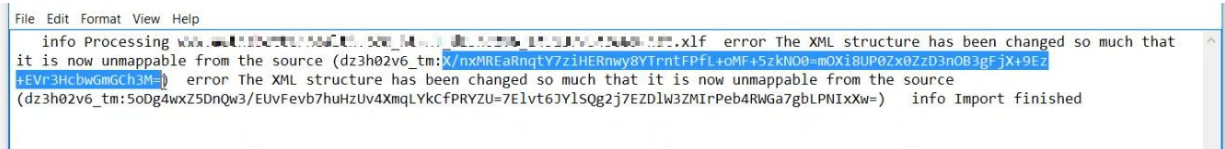
Troubleshooting

Error: *The xml structure has been changed so much that it is now unmappable from the source*

Fix:

1. Open both the **XLIFF file** and the **error log** in a text editor.

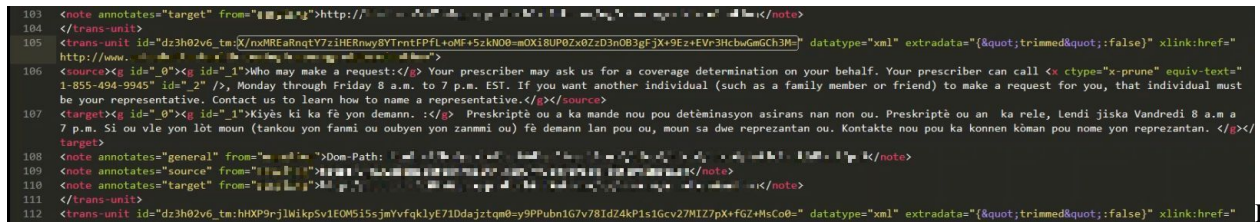
2. Select & copy the **TM-key** of the faulty entry, the part after ‘(trans-unit id=’xxxxx)_tm:’ in parenthesis right after the error message



Select

TM-key

1. Search for this key in the **XLIFF file** by pasting it into the ‘Find’ field. Only 1 translation unit will match.



Find

TM-key

1. Compare the tags in the source and target languages, and fix the mismatch by editing the **target text**. (You can also use an online text comparison tool for this task: copy-paste `<source> ... </source>` in one pane and `<target> ... </target>` in the other one.)
2. Save the corrected XLIFF file and upload again. It should give no error message now.

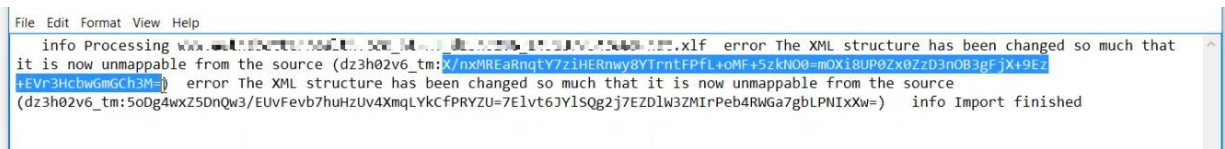
OR, alternatively,

1. Go back to your CAT tool, where you did the translation and open the faulty file for editing
2. Run QA. It will list you all the tag mismatches
3. Navigate to the faulty segment(s) and fix the tags
4. Export the corrected file and upload it. It should give no error message now.

Error: *Content found outside of outermost element***Fix:**

Practically this means that there is an extra space before the starting `<target><g*>` or after the closing `</g></target>`.

1. Open both the **XLIFF file** and the **error log** in a text editor
2. Select & copy the **TM-key** of the faulty entry, the part after ‘(trans-unit id=’xxxxx)_tm:’ in parenthesis right after the error message



Select

TM-key

1. Search for this key in the **XLIFF file** by pasting it into the ‘Find’ field. Only 1 translation unit will match.

```

103 <note annotates="target" from="http://www.1-855-494-9945" id="2" />
104 </trans-unit>
105 <trans-unit id="dz3h02v6_tm:K/nxMREaRngtY7ziHERmwy8YTrntFPfL+oMF+5zkN00-mOX18UP0Zx0ZzD3nOB3gFjX+9Ez+EVr3HcbwGmGCh3M" datatype="xml" extradata="{&quot;trimmed&quot;:false}" xlink:href="
106 http://www.1-855-494-9945" id="2" />
107 <source><g id="0"><g id="1">Who may make a request:</g> Your prescriber may ask us for a coverage determination on your behalf. Your prescriber can call <x ctype="x-prune" equiv-text="
108 1-855-494-9945" id="2" />, Monday through Friday 8 a.m. to 7 p.m. EST. If you want another individual (such as a family member or friend) to make a request for you, that individual must
109 be your representative. Contact us to learn how to name a representative.</g></source>
110 <target><g id="0"><g id="1">Kiyès ki ka fè yon demann. :</g> Preskriptè ou a ka mande nou pou detèminasyon asirans nan non ou. Preskriptè ou an ka rele, Lendi jiska Vandredi 8 a.m a
111 7 p.m. Si ou vle yon lòt moun (tankou yon fanmi ou oubyen yon zanmi ou) fè demann lan pou ou, moun sa dwè reprezantan ou. Kontakte nou pou ka konnen koman pou nome yon reprezantan. </g></
112 target>
113 <note annotates="general" from="Doe-Path: " id=" " />
114 </trans-unit>
115 <trans-unit id="dz3h02v6_tm:hXPP9r-jlWkp5v1EOM5i5sJmYvfklyE71Ddajztqm0-y9PPubn1G7v781dZ4Kp1s1Gcv27MIZ7pX+fgZ+McCo0=" datatype="xml" extradata="{&quot;trimmed&quot;:false}" xlink:href="

```

Find

TM-key

1. Delete the extra space around the tags
2. Save the corrected XLIFF file and upload again. It should give no error message now.

OR, alternatively,

1. Go back to your CAT tool, where you did the translation and open the faulty file for editing
2. Run QA. It will list you all the tag mismatches
3. Navigate to the faulty segment(s) and fix the tags
4. Export the corrected file and upload it. It should give no error message now.

IMPORTANT! Most of these issues can be avoided if the QA parameters of your CAT tool are set up properly, and you run QA before exporting your XLIFF files. Please make sure to check your translation for tag consistency and extra spaces; these are critical errors in website translation that can spoil the code.**

Error: Illegal character

Fix:

The reason for this error is usually a coding mismatch: all our XLIFF files are exported using the world-standard UTF-8 codepage. However, your CAT tool may save the file using another codepage, depending on the language, which may cause certain characters to appear as invalid.

1. Open both the **XLIFF file** and the **error log** in a text editor
2. Select & copy the **TM-key** of the faulty entry, the part after ‘(trans-unit id=’xxxxx)_tm:’ in parenthesis right after the error message

File Edit Format View Help

info Processing ... error The XML structure has been changed so much that it is now unmappable from the source (dz3h02v6_tm:K/nxMREaRngtY7ziHERmwy8YTrntFPfL+oMF+5zkN00-mOX18UP0Zx0ZzD3nOB3gFjX+9Ez+EVr3HcbwGmGCh3M) error The XML structure has been changed so much that it is now unmappable from the source (dz3h02v6_tm:50Bg4wxZ5DnQw3/EUVFevb7huH2Uv4XmqLYKcFPRY2U=7Elvt6JYl5Qg2j7EZDlW3ZMIRPeb4RWGa7gbLPNIxXw=) info Import finished

Select

TM-key

1. Search for this key in the **XLIFF file** by pasting it into the ‘Find’ field. Only 1 translation unit will match.

```

103 <note annotates="target" from="http://www.1-855-494-9945" id="2" />
104 </trans-unit>
105 <trans-unit id="dz3h02v6_tm:K/nxMREaRngtY7ziHERmwy8YTrntFPfL+oMF+5zkN00-mOX18UP0Zx0ZzD3nOB3gFjX+9Ez+EVr3HcbwGmGCh3M" datatype="xml" extradata="{&quot;trimmed&quot;:false}" xlink:href="
106 http://www.1-855-494-9945" id="2" />
107 <source><g id="0"><g id="1">Who may make a request:</g> Your prescriber may ask us for a coverage determination on your behalf. Your prescriber can call <x ctype="x-prune" equiv-text="
108 1-855-494-9945" id="2" />, Monday through Friday 8 a.m. to 7 p.m. EST. If you want another individual (such as a family member or friend) to make a request for you, that individual must
109 be your representative. Contact us to learn how to name a representative.</g></source>
110 <target><g id="0"><g id="1">Kiyès ki ka fè yon demann. :</g> Preskriptè ou a ka mande nou pou detèminasyon asirans nan non ou. Preskriptè ou an ka rele, Lendi jiska Vandredi 8 a.m a
111 7 p.m. Si ou vle yon lòt moun (tankou yon fanmi ou oubyen yon zanmi ou) fè demann lan pou ou, moun sa dwè reprezantan ou. Kontakte nou pou ka konnen koman pou nome yon reprezantan. </g></
112 target>
113 <note annotates="general" from="Doe-Path: " id=" " />
114 </trans-unit>
115 <trans-unit id="dz3h02v6_tm:hXPP9r-jlWkp5v1EOM5i5sJmYvfklyE71Ddajztqm0-y9PPubn1G7v781dZ4Kp1s1Gcv27MIZ7pX+fgZ+McCo0=" datatype="xml" extradata="{&quot;trimmed&quot;:false}" xlink:href="

```

Find

TM-key

1. Check if you see any strange characters, like squares or other meaningless characters.
2. Go back to your CAT tool and change the export options to use UTF-8 encoding. (As UTF-8 is a universal standard, it should be available.)
3. Re-export the XLIFF with UTF-8 encoding and upload it. It should give no error message now.

9.2.2 Publishing issues

Translated page doesn't show up

Issue: *I've just uploaded the translation of some new pages, but they don't show up on the translated site / they still appear in the source language.*

Fix:

You experience this problem because Target cache is enabled, and you need to clear the Cache to update content to be served. The very reason for using a Target cache is to mask unfinished translation and avoid bleed-through. The Target cache shows the last fully translated version of the pages - so if content changes on the original page, it remains hidden on the translated site until a fully translated version is available.

To fix the issue you need to explicitly delete the page from the Cache, so that the updated content could be loaded in upon the very first viewing of the page.

If you have several languages, it might be more convenient to clear the entire Target cache by clicking the Trash icon.

Translated page is not listed in Google search results

Issue: *The pages served from the Proxy are not listed in Google search results.*

Fix:

Google has never indexed the site in the first place, most likely due to the fact that there are no “hreflang” links on any of the pages, so the Googlebot has no idea there are other pages to look for. More information on the element and how it affects Google rankings may be found at <https://support.google.com/webmasters/answer/189077?hl=en>

Additionally, creating and submitting a sitemap (more information at <https://support.google.com/webmasters/answer/2620865>) to Google in order to force an indexing of the pages can also help. Even so, without the hreflang attributes, it may mean that some penalty is applied to the rankings, due to perceived duplicate content.

9.2.3 Using the hreflang Element

Google's CrawlerBot will eventually find your translated page if there are any links to it. However, if the content there is not marked appropriately, it will not be given the same SEO scores as your main content. In fact, it may even be treated as duplicate content, and a scoring penalty may be applied.

To prevent this from happening, you need to provide the GoogleBot with information on how the translated sites relate to the original. The easiest way to do this is the `<link rel="alternate" hreflang="" href="">` element.

These elements have to be placed in the page head (i.e. before the HTML body), and have two rules that must be satisfied in order for the GoogleBot to consider them:

- hreflang elements must be reciprocal: if a link points to a translated site, the translated site must point back to the original as well.
- hreflang elements must be circular: each language must also refer to itself with a link.

Consider the following HTML snippet from an imaginary site at `http://example.com`:

```
<html>
<head>
<title>Title Here</title>
<link rel="alternate" hreflang="en" href="http://example.com" />
<link rel="alternate" hreflang="jp" href="http://jp.example.com" />
```

(continues on next page)

(continued from previous page)

```
</head>
[...]
```

and its translated counterpart at `http://jp.example.com`:

```
<html lang="ja-JP">
<head>
<title>Title Here</title>
<link rel="alternate" hreflang="en" href="http://example.com" />
<link rel="alternate" hreflang="jp" href="http://jp.example.com" />
</head>
[...]
```

This snippet will provide proper SEO, since it satisfies both criteria: the references the English and the Japanese site are reciprocal (one refers to the other and vice versa) and the references are also circular (both languages also refer to themselves as well as their counterparts). This provides all the information GoogleBot needs to index each site in its rightful place and apply SEO scores across both domains.

For more information on this topic, see this article from Google: <https://support.google.com/webmasters/answer/189077?hl=en>

9.2.4 WPEngine issues

Redirections during crawling

Due to WPEngine’s *caching system* and *Redirect bots* settings you might experience any of the following issues on your **WPEngine hosted sites**:

1. Scan extracts outdated content from WPEngine cache
2. Scan returns **301: Moved permanently** error message for existing pages
3. Translated page is redirected from HTTPS to HTTP - which results in an error due to mixed content

WPEngine caching uses different so-called *buckets* based on request type, and they have one for bots. If the request comes from Google or other listed user-agents, and/or the URL has `?ver=` followed by a number, the redirect bots settings take effect.

The above issues can be resolved on the WPEngine site by turning off the *redirect bots*.

Intermittent HTTP403 on proxied pages

WPEngine automatically blocks traffic from “problem” IP addresses, typically those that generate large amounts of traffic in a short time. Due to the nature of the proxy, requests from several users may appear to have come through one IP, leading to WPEngine blocking that node due to their perception of “increased traffic”.

If that happens, and you note random pages of the proxied site not loading intermittently, call or chat WPEngine Customer Service with the following:

In relation to issue #874002, please enable proxy access on our installation.

According to an agreement with WPEngine, they will enable an alternate method of IP resolution that should no longer prevent access to the translated pages.

9.2.5 Captchas

Captcha doesn't work on the translated site

This issue is most likely results from page caching. Certain captcha solutions, like WP plugins are hardcoding the image URL into the HTML, instead of sending it asynchronously. During the crawl that builds up the Source cache for the project one hash is saved, so it becomes static, while it should be changing on each occasion. As a result, the server rejects the request because of the outdated verification image.

Fix:

- disable caching altogether to make captcha work
- use caching without captcha
- use another CAPTCHA system that uses async requests to retrieve the verification image, like Google's Re-Captcha solution

Another possible cause can be the CORS-header. If the proxied page is not listed as an allowed origin, the browser blocks the page when it tries to load the image.

10.1 General

10.1.1 Where are my translations published?

Instead of “where?”, a better question to ask is “how?”.

Imagine the proxy as standing between the original site and a visitor’s browser. Publishing the Japanese translation of example.com on the jp.example.com subdomain means mapping jp.example.com (presumably owned by the owner of example.com) to point to the proxy.

Visiting jp.example.com/contact.html results in that request being caught by the proxy and relayed to example.com/contact.html - the origin server. The contact.html page is served as a response, which is caught on the way back, translated on-the-fly at blazing speeds and served to the visitor.

This requires that jp.example.com be mapped to the cloud proxy application in the owner’s DNS settings.

10.1.2 Does the proxy host a copy of my site?

No. The proxy does not store any copies of the original site pages, it only stores translations, which it uses to process responses served by the original site to visitor’s queries.

There is one exception to this principle: if a source cache is built and enabled for a proxy mode, that cached version of the page will be used in place of the origin server’s response.

10.1.3 Some parts of a site are on a subdomain. How will the crawler pick them up?

The sites www.company.com and blog.company.com are treated as separate domains by the crawler. From the vantage point of a crawler running on www.company.com, a path on blog.company.com is an external and will be treated as such. The solution is to create two separate projects and link those with each other.

10.1.4 The Discovery went beyond the limit I set. Why?

A crawl will finish the current round and visit the redirects and links on the last page. If it took the limit too literally, that could potentially result in trailing links being thrown out.

10.1.5 Can I get page-specific repetition statistics?

Repetition statistics make the most sense in a site-wide context. The problem with controlling calculations on a per-page basis is that it is not true to life to call a segment on a given page a “canonical instance”. Take a navigation bar or a footer, for example. It will be “repeated” on all pages, but it cannot be said to “belong” to any one of them. the translation proxy stores the first instance it comes across and then propagates its translation to all other instances.

10.1.6 The page I’m trying to translate has prices. What can I do to handle local currencies?

The prices themselves can be made translation-invariable, but real-time price handling for different currencies will have to be implemented by the client on the source site, making it possible for the proxied site to access the locale-specific information. Pricing of products and services also has legal / market implications that are beyond the tasks of LSPs. Of course, once currency-specific information is accessible from the original site, we are happy to help with integrating any backend calls / ajax requests on the proxy.

10.1.7 How do I enable automated scan on my project?

To enable automated content extraction on your site, please go to Content, and choose either of the daily, weekly or monthly options in the drop-down next to the Look for changes option.

10.1.8 Is it possible to set up automated scanning behind secure login?

No, scanning can’t be automated behind secure login. For such processes you need to [extract cookies](#) with you browser’s dev. tools and pass them on to the proxy. Some cookies get invalidated over time, and we don’t store cookies either.

10.1.9 What do the various tags mean next to each page in the page list?

See the Glossary for a detailing of the various tags encountered in the page list.

10.2 Caches

10.2.1 Can I preview newer content on the workbench without causing bleedthrough on the published site?

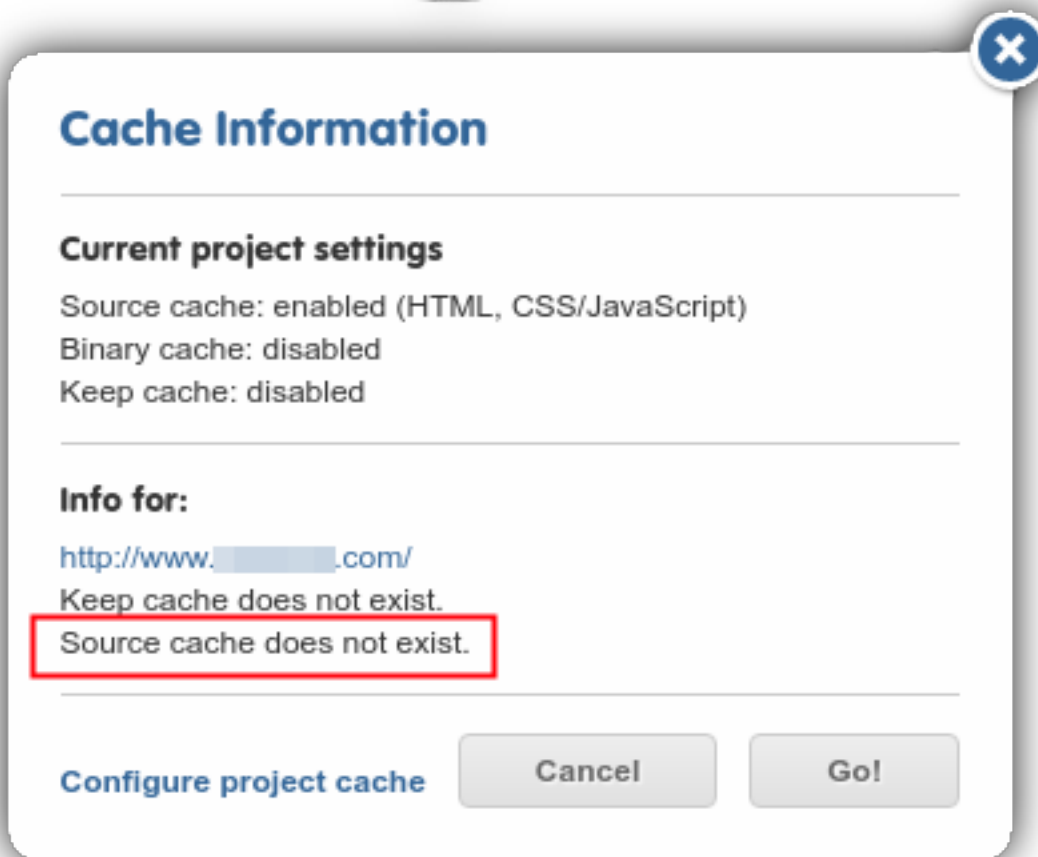
You can customize which Source Cache to use on which proxy mode - go into Page Cache, choose custom settings and select Disabled from the dropdown menu. The preview mode will display all new content. It is recommended that you keep TM Freeze turned on while exploring the new content, otherwise everything will be automatically added to the Workbench.

10.2.2 Does building a Source Cache cost any money?

You can use Content Scan with the appropriate options checked to build your Source Cache. As long as there is no new content to pick up, Scan costs the same as a Discovery.

10.2.3 How can I check if a page uses the Source Cache?

Go into **Pages view** in the Dashboard. If you hover with the mouse over a page in the list, you will see a **Cache** button. Click on it to verify Source Cache information for that page. If there is no Source Cache for that page, you will see the following screen:



Cache

Does Not Exist For Page

10.2.4 Does building a Target Cache cost any money?

Setting aside the inherent cost of the Page Visits you have to accrue to build them, Target Caches are free of charge.

CHAPTER 11

Glossary

We use a set of recurring terminology in this manual and also in our Support Channels - we collect them here for your reference.

11.1 Page List Tags