Integrate your project with translation management systems

March 13, 2023 • Jiri Lojda and Ondrej Chrastina • 12 min read

If you need to provide localized content to audiences in multiple regions, you want to simplify the translation process and make it scale. To help with the translations, use a translation management system (TMS) of your choice and integrate it with Kontent.ai. This helps ensure consistent and accurate translations.

Let's go through the decisions you need to make to ensure the translation process works just the way you need.

How your content flows from Kontent.ai to a TMS and back

The translation process is triggered in Kontent.ai. You react to the trigger by creating a translation unit for the TMS. A translation unit contains one or more content items that need to be translated.

Once the translation unit is processed on the TMS's side, you get a new translation unit back. Now you take the translated content from the new translation unit and import it to your Kontent.ai project.

See the diagram on https://viewer.diagrams.net?lightbox=1&nav=1#Uhttps://raw.githubusercontent.com/Kontent-ai-Learn/kontent-ai-learn-diagrams/master/integrations/tms/tmsIntegrationOverview.drawio

1. Choose a trigger for the translation

Start by choosing the action that triggers your translation process.

A. Move an item to a specific workflow step

The easiest approach for triggering the translation process is changing a content item's workflow step.

Advantages	Disadvantages
— Prevents users from editing the item while it's being translated. — Provides a clear overview of all items being translated, for example, by filtering by workflow step.	The item's content might be in an invalid state due to incomplete elements.

B. Trigger from a custom app or custom element

Use a custom-built app to trigger the translation process. Optionally, you can include the app as a custom element in Kontent.ai.

Advantages	Disadvantages
Offers the most customization options.	— Content creators might forget to trigger the translation process. — Doesn't enforce the translation workflow. — Requires you to insert the custom app somewhere outside the content item. — The item's content might be in an invalid state due to incomplete elements.

C. Publish an item

Trigger the translation process when users publish a content item.

Advantages	Disadvantages
Ensures the item's content is in a valid state without any incomplete elements.	— You need to publish content in the source language before you receive translations from your TMS. This means you cannot publish your content in all languages at once. — If users create a new version of the published item, you cannot retrieve the published version via Management API.

2. Choose how to respond to the trigger

Once you decide how to trigger the translation process, you need to respond to it.

A. Passive waiting for webhook push notifications

The easiest way is to set up your app to wait for webhook notifications.

Advantages	Disadvantages
Easy to configure.	No way to control the number of incoming requests.

B. Active waiting with checking Sync API

If you need control over the time the translation process starts or over the number of requests your app receives at once, consider active waiting. You can use the Sync API to periodically check for the latest changes.

Advantages	Disadvantages
Fine control over when the translation process starts.	Requires more work to set up.

3. Choose your translation strategy

Define your translation units. Do you need to send a single content item for translation or several of them at once? This depends on the way you structure your content. For example, information about products might live in individual content items, whereas web pages might be built from multiple content items.

When you trigger the translation process, what content should be sent for translation? This can differ based on ***content types*** *or other criteria.*

See the diagram on https://viewer.diagrams.net?lightbox=1&nav=1#Uhttps%3A%2F%2Fraw.githubusercontent.com%2FKontent-ai-Learn%2Fkontent-ai-learn-diagrams%2Fmaster%2Fintegrations%2Ftms%2FtreeTranslation.drawio

A. Translation unit as a single content item

In some cases, you might want your translation unit to consist of a single content item. This would be the content item that triggered the translation process.

Advantages	Disadvantages
Simple to implement.	— Content from a single identifiable piece of content will not be collocated in the TMS. This could lead to a worse experience for human translators or possibly worse results for machine translators. — You will need to trigger the process for all content items that are part of one identifiable piece of content if you want to translate it whole.

B. Translation unit as all items in an identifiable piece of content

If you structure your content from multiple content items, you need to trigger the process for the root item. That is the root of your identifiable piece of content like a web page. Then traverse the children of the root content item.

Advantages	Disadvantages
— The whole identifiable piece of content is translated together. — Easier for machine and especially human translators to see the context in which the content is used and create a better translation.	Harder to implement compared to a single-item approach.

Ensure your content is sent for translation just once

Keep in mind that you should protect all items that are part of the same identifiable piece of content. However, this change might trigger your webhook or Sync API listener again for all the child items.

To prevent that, either set up your webhook to only trigger for the root content type such as Page, or, if using Sync API, filter only changes related to the specific content type.

4. Consider protecting content during the translation process

It might be beneficial to protect specific content item variants during the translation process to avoid errors. You can protect a variant by creating a workflow step such as *Translation in progress* where no user, apart from the integration itself, can make any changes. Then you need to move the variant to the workflow step. Once the translation is done, the process moves the variant to the next workflow step so that users can work with it again.

See the diagram on https://viewer.diagrams.net?lightbox=1&nav=1#Uhttps%3A%2F%2Fraw.githubusercontent.com%2FKontent-ai-Learn%2Fkontent-ai-learn-diagrams%2Fmaster%2Fintegrations%2Ftms%2FsourceTargetVariants.drawio

Protect the target variants

Target variants are those that you are translating to. Once the translation is finished, its result is written into those variants. Existing content, if the variants already exist, is overwritten. For this reason, we recommend protecting those variants for the duration of the translation process to avoid any data loss.

Protect the source variant

The source variant is the one you are translating from. There's a chance that someone will edit the source variant during the time after the variant was triggered for translation and before the translation process reads the data from the source variant, which results in unexpected data being sent for translation.

If you protect the source variant:

- The translation process is safe from modifications between the trigger and the process that is reading the data.
- You can easily see all the item variants that are part of a translation process.

5. Transform and translate content

To maintain the original structure of your rich text elements in the translated text, convert the elements into a format that preserves the information about the position of entity references like linked items or images.

Also, make sure to transform the translation unit into a format supported by your TMS, for example, JSON or XLIFF.

A. Insert placeholders in the rich text content

Insert placeholders that are not supposed to be translated into the text and then use the placeholders to insert the references back into the translated text.

See the diagram on https://viewer.diagrams.net?lightbox=1&nav=1#Uhttps%3A%2F%2Fraw.githubusercontent.com%2FKontent-ai-Learn%2Fkontent-ai-learn-diagrams%2Fmaster%2Fintegrations%2Ftms%2FrichTextPlaceholders.drawio

Advantages	Disadvantages
- Easy to implement. - Content from the rich text element remains as one piece of text in the TMS.	Some translation management systems might not support placeholders in the text sent for translation.

B. Split the rich text by references

Split the rich text by references into text chunks and encode the references that should be between them as translation keys or metadata of those chunks.

For example, you can use a slight modification of XPath to encode a place of each text chunk in a JSON and use an empty string as a value for keys pointing to a reference.

See the diagram on https://viewer.diagrams.net?lightbox=1&nav=1#Uhttps%3A%2F%2Fraw.githubusercontent.com%2FKontent-ai-

Learn%2Fkontent-ai-learn-diagrams%2Fmaster%2Fintegrations%2Ftms%2FrichtextSplit.drawio

Advantages	Disadvantages
Doesn't require special placeholders in the text.	— More challenging to implement than placeholders. — The rich text element is split into multiple chunks, which might be more difficult to translate for the TMS, especially for human translators.

6. Choose the right API to read from Kontent.ai

You can read content from Kontent.ai using either Delivery API or Management API. For updating content, you need to use Management API.

A. Read from Management API only

Because Delivery API is designed for displaying data in your application, we recommend using Management API for transforming and translating content. This is because you need to use it to update content items. Also, using Management API only means you won't need to transform the rich text format between the APIs as each API deals with rich text differently.

Advantages	Disadvantages
No need to transform between different rich text formats.	— Tighter rate limitations. — Limited filtering options.

B. Filter with Delivery API and read from Management API

If you need to retrieve specific content items using filters, use Delivery API. Then use the IDs of the content content items to also retrieve the items via Management API one by one.

Advantages	Disadvantages
— No need to transform between different rich text formats. — Use advanced filtering in Delivery API.	— Tighter rate limitations of Management API.. — A combination of two APIs might add extra complexity.

C. Read from Delivery API only

You can also choose to read data just from Delivery API, but the API is primarily designed for displaying data to users.

Advantages	Disadvantages
Use advanced filtering in Delivery API.	The format of responses, rich text elements specifically, is different for Delivery and Management API. Because you can update content only via Management API, you need to convert the content to the Management API format. This adds extra complexity and might break in the future.

7. Import translated content

The final step in the translation process is importing the translated content. It is the reverse process of the text transformation and a language variant upsert request.

After the import is finished, your translation process can move all the source and target variants into the next workflow step. If the variants were protected during the translation, the change of workflow lets users know that the translation is done.

What's next?

- Check out the list of Kontent.ai's translation integrations.
- Set up languages and language fallbacks in your project.
- Set up your regional content to meet your worldwide audiences.
- Check out the Kontent.ai APIs.