Translating Digioh Campaigns (Creating Internationalized Campaigns)

Last Modified on 08/14/2026 11:55 am EDT

Digioh supports dynamic translation of campaign content via our Internationalization (i18n) extension. This extension allows you to build Campaigns in your native (default) language, and configure translation overrides for the various elements (text, buttons, images, etc) in a hosted CSV file. When visitors trigger the Campaign, the app will detect the language and dynamically replace the campaign elements with their translated equivalents from CSV data.

Install the Extension

Install the extension once from your Digioh account.

1. Click your account dropdown (top right) and select **Extensions**.
2. Click the expand button to see all available extensions:



3. Find the **Internationalization (i18n)** extension and install it.

Language Detection

The i18n App will automatically detect the language in priority, as follows:

1. URL parameter lang, e.g. site.com/?lang=**fr**
2. Custom Page JS function api.getI18nLanguageCustom if defined (see advanced configuration below)
3. Custom Regular Expression (see advanced configuration below)
4. Language in URL path, e.g. site.com/ **fr**/pricing/
5. Browser Language Settings

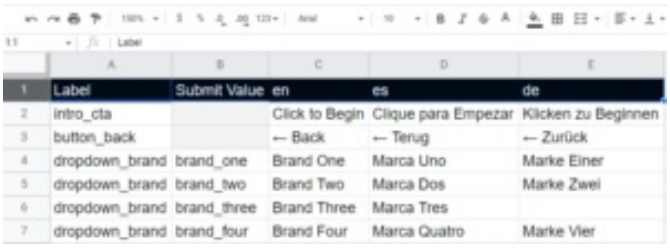
Configure a hosted CSV or Google Sheet:

Your i18n CSV must be structured with columns:

1. Label - the "key" for a specific element needing text translation
2. Submit Value - for configuring translation of dropdown items

3. Columns for each language, e.g. "en", "fr", "es", etc

And each row represents a specific element in your Campaign.

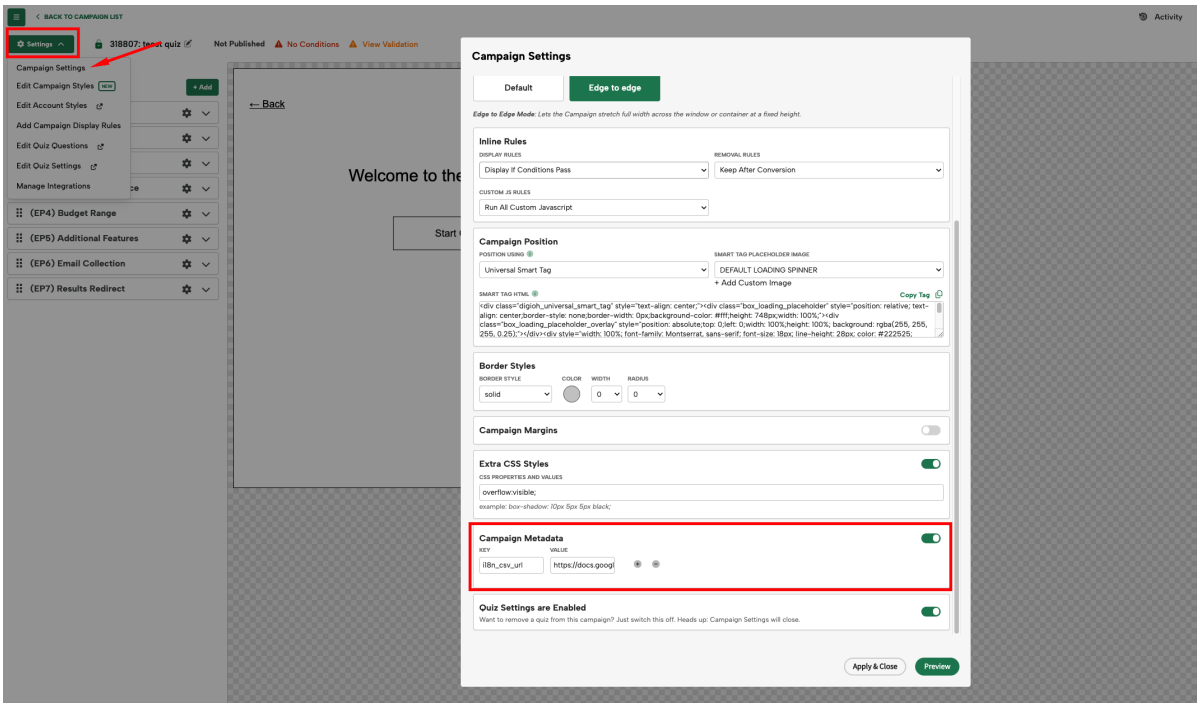


	A	B	C	D	E
1	Label	Submit Value	en	es	de
2	intro_cta		Click to Begin	Clique para Empezar	Klicken zu Beginnen
3	button_back		← Back	← Terug	← Zurück
4	dropdown_brand	brand_one	Brand One	Marca Uno	Marke Einer
5	dropdown_brand	brand_two	Brand Two	Marca Dos	Marke Zwei
6	dropdown_brand	brand_three	Brand Three	Marca Tres	
7	dropdown_brand	brand_four	Brand Four	Marca Quatro	Marke Vier

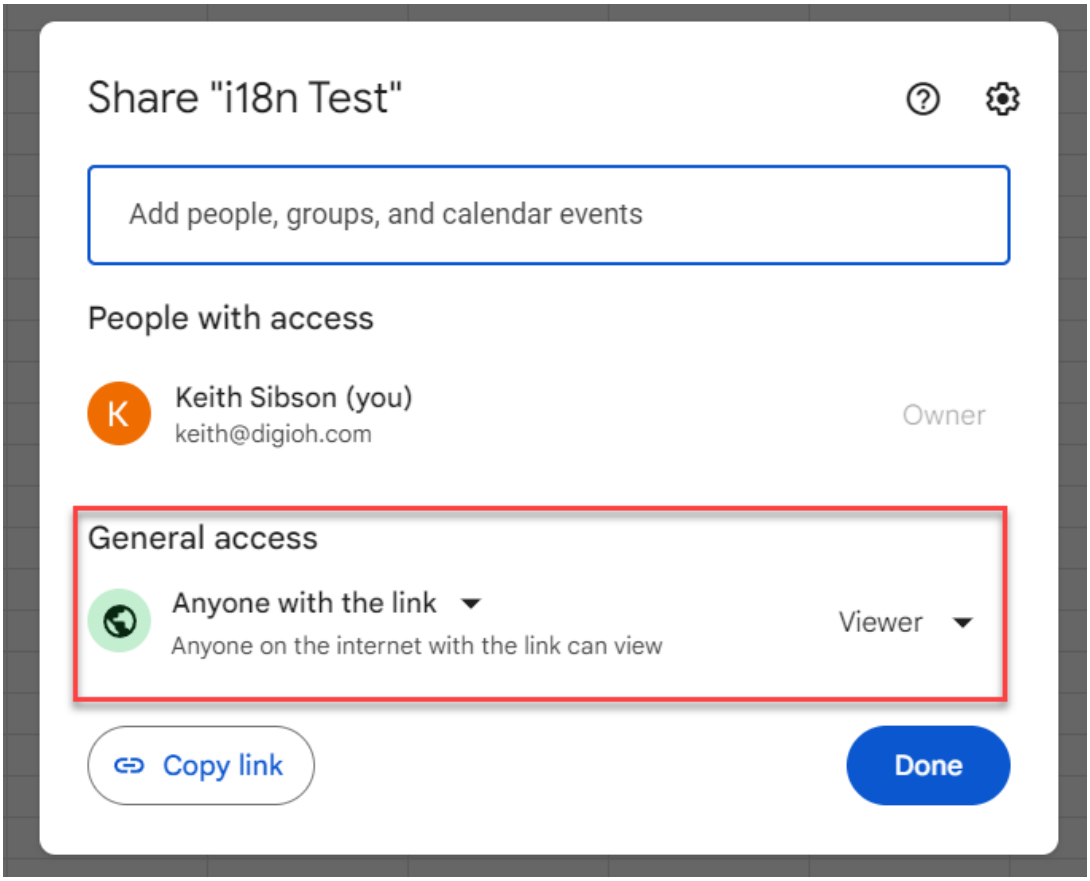
Here is an example reference Google Sheet.

Configure Campaign i18n URL

If the i18n app is installed, it will only activate for your Campaign if it finds a configured CSV URL in Campaign metadata:



The metadata key is **i18n_csv_url**. The metadata value can be any URL that returns raw CSV data. You can host CSVs in Digioh Files, refer to this page for instructions on getting the URL. Using a Google Sheet is also a good option, when in the Google Sheet, simply copy the URL from the address bar. Note that the URL changes depending on which sheet (tab) you are viewing, the link references a specific sheet so you can store all of your translation data for different campaigns in different sheets within the same document. The i18n app will automatically convert the configured GSheet URL to a CSV export mode. When using a GSheet, you **must** share it so that "anyone on the internet can view".



An older version of this app supported metadata keys i18n_gsheet_id and i18n_gsheet_gid, but we recommend using i18n_csv_url regardless of where your i18n data is hosted.

Configure Translation Elements

We recommend designing your campaigns using your default language (e.g. English). Then in the Campaign Editor, you configure metadata for all elements you wish to translate that is their i18n "label". These labels are what the i18n app uses to look up replacement content based on the detected language, and so should exactly match the values in 'Label' column of the Translation GSheet. When Previewing the campaign, or in Campaign QA Mode, the app will show you an error if any label configured on an element is not found in the CSV.

Example:



In this example, a CTA on the first page of the lightbox will be translated ('Click to Begin'). The translation spreadsheet has a row with 'intro_cta' in the label column, and the translations for that button in the CSV columns. For example, the Spanish translation under the 'es' column, and German under 'de', etc.

Image Elements

If you want to swap out images based on language, simply add i18n with an appropriate key name to the image element. The "translation" value in the sheet should be a full image URL starting with "http".

Pro Tip: You don't need to define the URL for the default (usually "en") image, or any image that is the same as the default configured in the Campaign Editor. Just leave the cell blank and the image will show as it is in the Campaign Editor.

Reusing Translations

When there are buttons or text used multiple times throughout the lightbox, simply use the same label for each.

For example, if there are back buttons on each page, each back button in the editor can have its metadata configured identically:

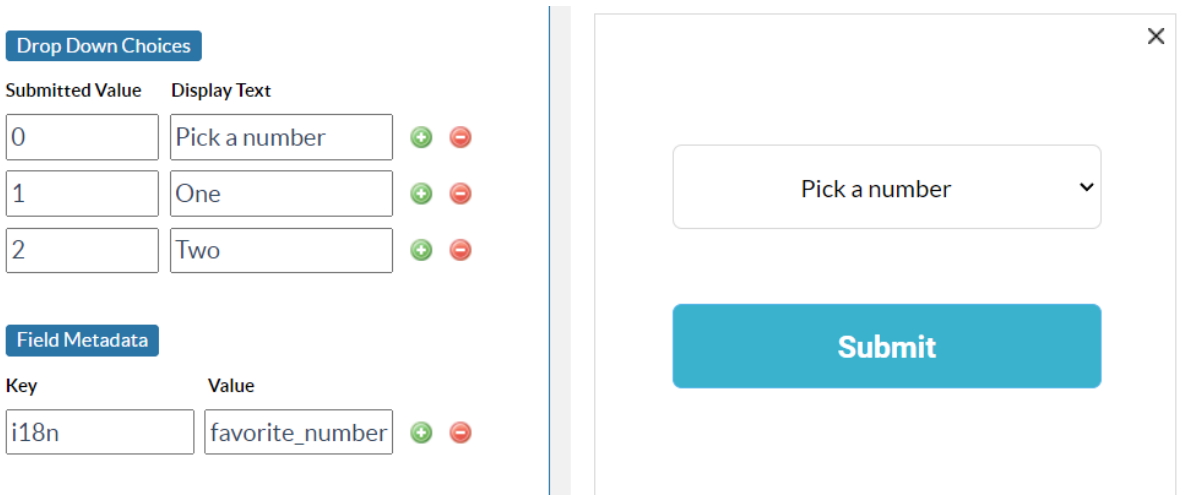
- Key: i18n
- Value: back_button

In the i18n CSV, only one row needs to be set for these buttons with the translation.

Internationalization of Dropdown Option

To translate dropdowns options, you need to set the option values for each language. This is the purpose of the "Submit Value" column in the i18n CSV. This allows you translate some or all of the dropdown options. The Submit Value column also allows you to change the available options for a specific language. For example, if you have a dropdown for "Select your Region", then those regions can be different for each locale. You will need to have a row in the CSV for every unique dropdown option submit value in all languages, but simply leave the translation cell blank for columns that are not applicable in that language.

Example:



Here, we translate the options, while also changing the available options to "Zwei" and "Drei" (two and three) for German specifically.

	A	B	C	D	E
1	**Label**	**Submit Value**	**en**	**es**	**de**
2	favorite_number	1	One	Uno	
3	favorite_number	2	Two	Dos	Zwei
4	favorite_number	3			Drei

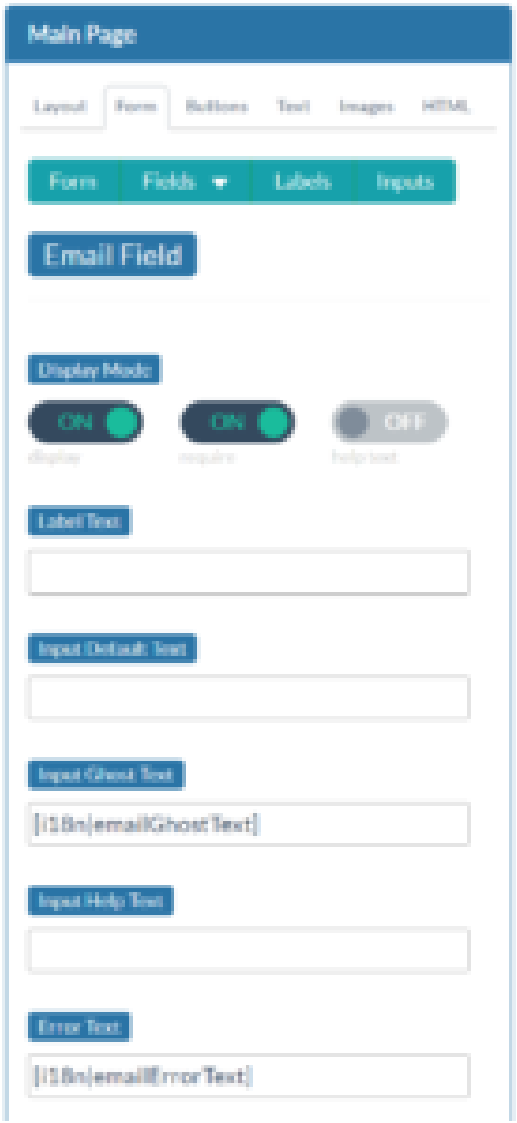
Pro Tip: You can use the i18n app to power dynamic dropdowns without translating them!

[i18n|*label*] Merge Tag

If you need to translate parts of an element or element attributes like ghost text, error messages, or values in hidden fields, you can use the i18n **merge tag**. First, you need to install the **Universal Merge Tags App**.

Then, for example, if you want to internationalize input ghost text for a specific field:

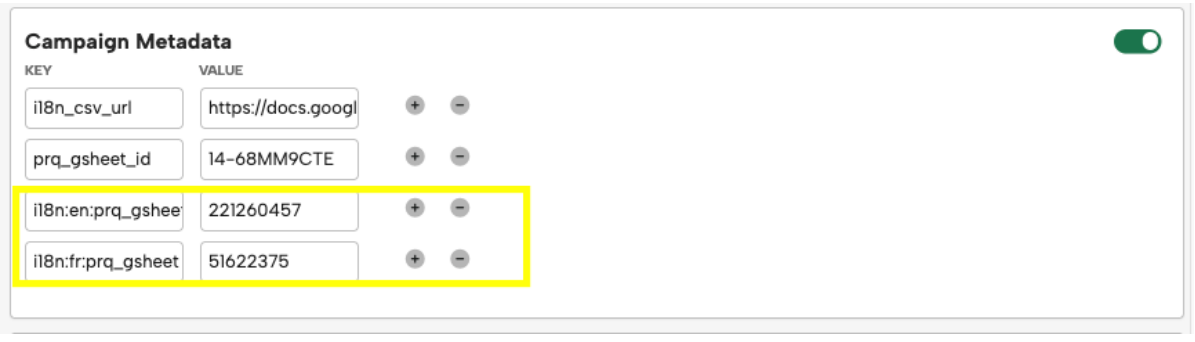
- Configure the ghost text input of the element with [i18n|email_ghost_text]
- Add a row for label email_ghost_text to your CSV
- Add translations in each column



Pro Tip: You can use the [i18n|*label*] merge tag anywhere in your campaign, e.g. to translate individual words in a text element.

i18n Interoperation with Other Apps

You may need to adjust the configuration of other apps based on the dynamically determined language. To support this, i18n recognizes and processes a prefix of **i18n:**key_name: for metadata keys. So for example, if you are using the Product Recommendation Quiz app, you might want to swap out a different product GSheet based on the language:



This works for any metadata command. In general, configure alternate metadata commands for each possible language, with the i18n:*lang*: prefix. So if your metadata command is "app_command", and you have languages en and fr, then use metadata:

i18n:**en**:app_command = something English specific i18n:**fr**:app_command = something French specific

How this works is that i18n first detects the language, then scans all metadata looking for a prefix match. All matching commands have the prefix removed, thereby activating those app commands dynamically. The other app will not recognize commands where the prefix remains, only those that have been activated by i18n.

Customizing Language Detection (Advanced)

There are two ways to customize language detection.

1. Widget-level metadata that defines a regular expression to match against the current URL. This can be any regular expression, but the app will take the **first** match, and use that as the language. So whatever the regular expression returns as the first match must exist as a column header in the CSV data. For example, this is a configuration that would match "fr" in these URLs: https://site.com/fr/welcome/ and https://site.com/fr?utm_source=google

i18n_lang_regex : /\/([a-z][a-z])(\/|\?|$)/

2. Alternatively, you can implement custom language detection by creating a Custom JS function in Page JS: **api.getI18nLanguageCustom**. Here is an example that extracts language from your site's subdomain, e.g. https://**fr**.site.com/

```
api.getI18nLanguageCustom = function( ){
    let url = window.location.href;
    let match = url.match(/^https?:\/\/([a-z][a-z])\./);
    if(match) return match[1];
    else return "en";
}
```

Customizing the Source of CSV Data (Advanced)

If you want to provide CSV translation data directly to the app, instead of (or in addition to) a hosted URL, you can do so by configuring widget-level metadata; one of:

- i18n_csv_function : window.*function_name*
- i18n_csv_function : api.*function_name*
- i18n_csv_function : campaignapi.*function_name*

You must define a function on your site, directly in the window object, or in Digioh Custom JS in Page JS (api.) or Campaign JS After DOM Ready (boxapi.).

The i18n_csv_function will be called (with no arguments) and is expected to return raw CSV data in the Digioh i18n format as a string. If the function doesn't exist at runtime, or returns null,

undefined, or empty string, then i18n will fall back to the configured i18n_csv_url, if any.

Note that the i18n app does not currently support asynchronous functions for i18n_csv_function.

Questions? Comments? Let us know at support@digioh.com