

Shopify Frontend

The Create and Edit product pages have a similar structure, however there are a few big differences between them. I will describe the common points first, then each specific page after that.

Basic Structure:

Pages are split into 2 sections (3 for the edit page).

1. *Header, name and inputs* - this contains the beginning of each page, with inputs (or dropdowns) for name, brand, and other small details about the product. Also here you can find the Save product button
2. *Main product tags* - The list of tags we can select from, separated into their headline titles. Also contains the Add Subpart button (as a tab header)
2' - when pressing the Add Subpart button, we are shown the *New Subpart* form. After adding the info here, we will have a new tab with a similar view as the *main product tags*, but with headlines customized to fit the subpart.

1. Generate texts - contains the form and result container for the generating of texts. Only appears on the Edit Product view.

Although the structure is mostly similar, the page building is done in different ways for the two pages.

Create Product Page:

The Create Product template is only populated on the backend with the category data (only the category dropdown is known).

The other parts of the page are loaded dynamically depending on which category is chosen by the user. For this, we use the functions **initKinds** to populate and show the kind, name and brand inputs, and **initTags** to show the main product tags. (**renderTags** is used inside this to actually create the HTML elements and add them to the page)

In order to retrieve the page data from the backend, AJAX calls are made inside the **tagStore**, **subpartStore** and **kindStore**. They use a simple caching system to not do the same call multiple times.

Adding a subpart is handled by the **addSubpart** function, while toggling a tag as selected/deselected is handled by **toggleTag**.

Saving the product is done using the **saveProduct** function.

Edit Product Page :

The Edit Product page, in contrast to the Create Product page, is populated as much as possible from the backend. This means that most inputs, and the main product tags come pre-filled directly in the template. Selecting the tags and adding the subparts is then done in the **renderTags** function on the frontend, similarly to what we find on the Create Product page.

Adding a subpart is again handled by the **addSubpart** function, while toggling a tag as selected/deselected is handled by **toggleTag**.

Saving the product is done using the **updateProduct** function.

Generating the texts is then handled in the **generate** function.

SubpartView:

Each subpart tab, on both the Create product and Edit product pages, is handled using a Subpart view. The functionality is similar to what we find with the main product tags on both pages, however it is kept separate for each tab, and under its own scope.

We populate each view using the **render** function, and we once more use a function named **toggleTag** for selecting/deselecting a tag.

Also, in order to retrieve the selected tags on the main pages, we provide the **getTags** and **getRemovedTags** functions.

TagService:

For handling the selection and deselection of tags, and for providing correct JSON when doing the saving calls, we use a Tag Service. It does not interact directly with the DOM and it is only used to keep track of changes on the page and make sure we send them correctly to the backend.

Relevant CSS Classes:

For representing the tags we use Materialize CSS Chips component.

We use several CSS classes to differentiate between the different types of tags.

- The default **chip** class is used to show a simple unselected tag. (current color: green)
- **customChip** - is used for tags that do not appear in a vocabulary grammar (current color: orange)
- **tagSaved** - is used in the Edit product page to show tags that were previously saved. (it shows a highlight border around the tag, with a color similar to the tag itself)
- **chipselected** - is used on both pages to show tags that have been selected but have not yet been saved (it shows a highlight border around the tag, with a color similar to the tag itself)