Frosmo "Happy Valley" 2019-11-20

The Happy Valley release introduces two major improvements to the Frosmo Control Panel: a completely redesigned modification UI and an improved content template workflow.

- Major changes
    - New modification UI
    - Improved template workflow
    - Availability on release
- Bug fixes

For the Frosmo Platform component versions in this release, see the changelog.

Major changes

New modification UI

The modification UI, which you use to create and edit modifications in the Control Panel, has been completely redesigned for improved usability and a smoother workflow. The redesign also introduces a more powerful integration with templates and comes with a few conceptual changes.

Here are the highlights:

- Create by use case, not type
- Smooth, centralized workflow
- Content preloading
- Content types and settings
- Variation preview
- New look-and-feel
- New in templates: Schemas
- New in modifications: Content options UI

For more information about modifications, see the modifications user guide.

⚠ The new modification UI will not be available to all customers on Happy Valley release. For more information, see Availability on release.

Create by use case, not type

Modifications are now categorized by use case, or simply "case", rather than by type. Types, namely basic and cached, no longer exist as such. When you create a modification in the new UI, you start by selecting the case for the modification. The Frosmo Platform currently supports the following cases:

- **A/B test.** Create a modification with two or more variations to test which variation performs the best. You manually define the variation distribution, that is, the percentage of eligible visitors to whom each variation is shown.
- **Multi-armed bandit.** Create a modification with two or more variations, and select a multi-armed bandit algorithm for automatically optimizing the variation distribution over time. The better a variation performs relative to the other variations, the bigger the portion of visitors to whom the platform shows the variation.
- **Personalization.** Create a modification with a single variation to provide personalized content to your visitors.
- **Selective profiling.** Create a modification with a single variation (and no comparison group) to provide non-personalized content that is safe to display to visitors who have refused profiling. This case is only available on sites that have the selective profiling module enabled. For more information, see Profiling solutions on the Frosmo Platform.

Figure: Creating a modification in the new UI (left) and in the old UI (right) (click to enlarge)

⚠ Existing type-based modifications created in the old modification UI are converted to conform with the new UI as follows:

- If an existing basic or cached modification uses the multi-armed bandit feature, the modification is converted to a multi-armed bandit modification.
- If an existing basic or cached modification does not use the multi-armed bandit feature, the modification is converted to an A/B test modification. This applies even to modifications that only have one variation.
- Cached modifications are converted to the appropriate case (as described see above) with content preloading enabled.
- Legacy recommendation modifications, which are incompatible with the new UI, are not converted. Instead, opening a legacy recommendation modification in the Control Panel opens the old modification UI.

Frosmo handles the conversions on Happy Valley release, when we enable the new UI for sites.

Smooth, centralized workflow

In the new UI, configuring a modification is built around the following workflow:

1. Define the visitor segments that you want to target with the modification. If you want to target all visitors, do not define any segments.
2. Define the placement for the modification.
3. Define the content for the modification.
4. Optionally, customize the schedule and define the advanced settings for the modification.

You perform each step from a single, shared settings view. Moreover, in defining the segments and the placement, you can either select existing resources, or create and select new ones, without ever leaving the workflow and the modification UI.
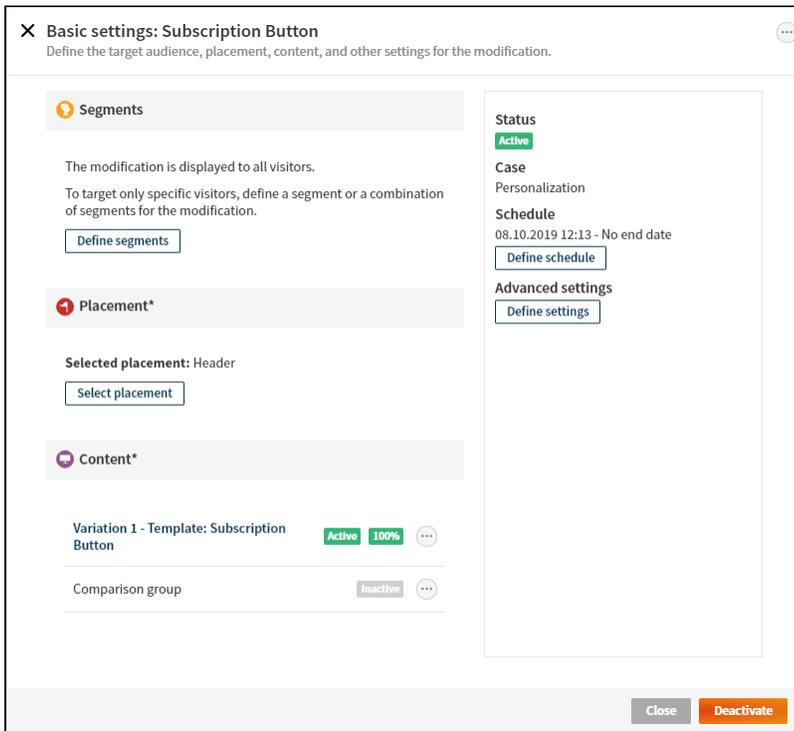
Figure: Easy access to all resources related to building a modification (click to enlarge)

Content preloading

In the new UI, the **cached modification** type is replaced by the **content preloading** advanced setting.

When you enable content preloading for a modification, the modification content gets loaded in the browser as a part of the custom script, decreasing the delay in rendering the modification. In short, the modification now functions like the cached modifications of old. Unlike with cached modifications, though, you can always disable content preloading for the modification, after which the modification content is again retrieved from the Frosmo back end.
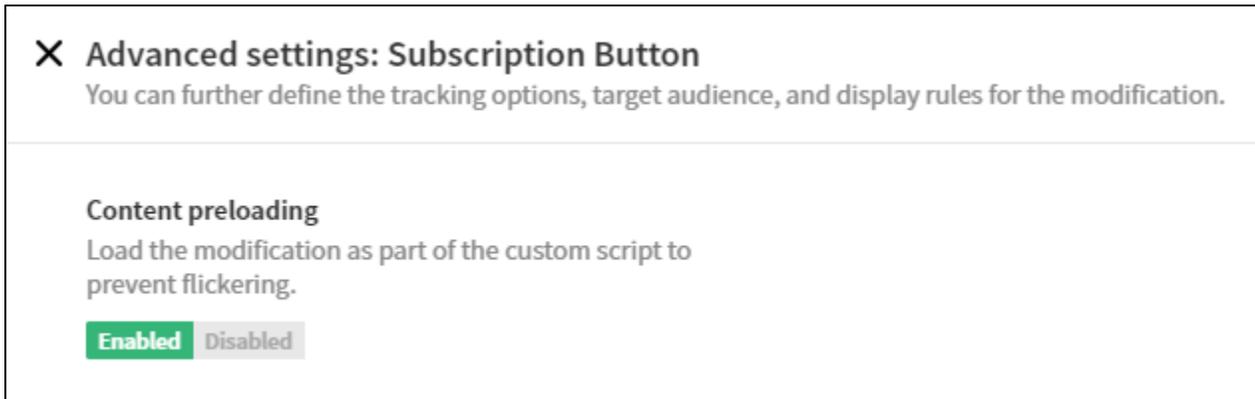

Figure: Enabling content preloading in the advanced modification settings

Content types and settings

Modification variation content involves the following new key concepts and settings:

- **Content type** determines where the content is defined. You can select from the following types:
  - **Custom content**: The content is defined directly in the variation and is therefore only available for that variation.
  - **Content template**: The content is defined outside the modification in a template . A variation links to the template for its content. The template, and therefore the content, is also available to other modifications and variations.
  - **Original content**: No content is defined for the modification. The variation is used for tracking the performance of the original page content. This allows you to compare how the other variations perform against the original content.
- **Content settings** determine the actual content that gets rendered on the page. The available settings depend on the selected content type:
  - For custom content: Single text field where you manually define the HTML, CSS, and JavaScript code that together make up the content. This is basically the same field as the variation content field in the old UI.

- For a content template: **Content options UI** generated from settings defined in the selected template. You customize the content for this variation by simply filling out whatever fields the template requires. The values you provide replace whatever placeholder values the template content contains, resulting in the final content that gets rendered on the page.
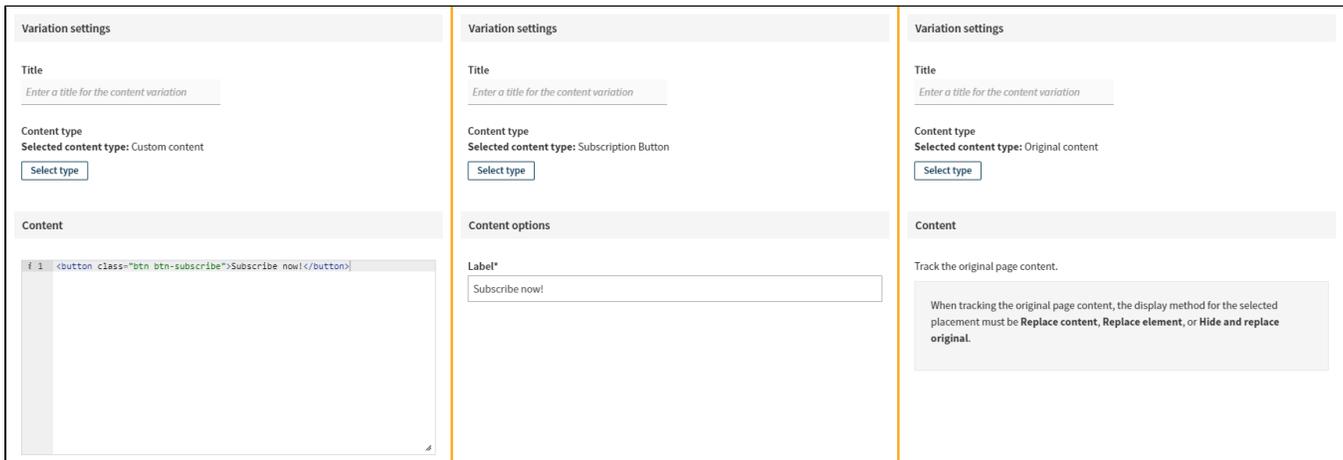- For original content: No content settings, since you define no content for the modification.

Figure: Variation content settings for each content type: custom, template, original (click to enlarge)

Variation preview

You can now easily preview variations directly from the Control Panel. To preview a variation, all you need to do is define a preview URL for the associated placement, and then select **Preview** from the variation quick menu in the modification settings. The preview opens the specified preview URL and displays the variation on that page.

The preview uses the Preview Tool to display the variation, so you can preview both active and inactive variations, and the modification itself can be either active or inactive. The preview always retrieves the modification content from the Frosmo back end, so the content preloading advanced setting has no effect, if enabled. In addition, you don't have to be in the target audience defined for the modification to see the variation.
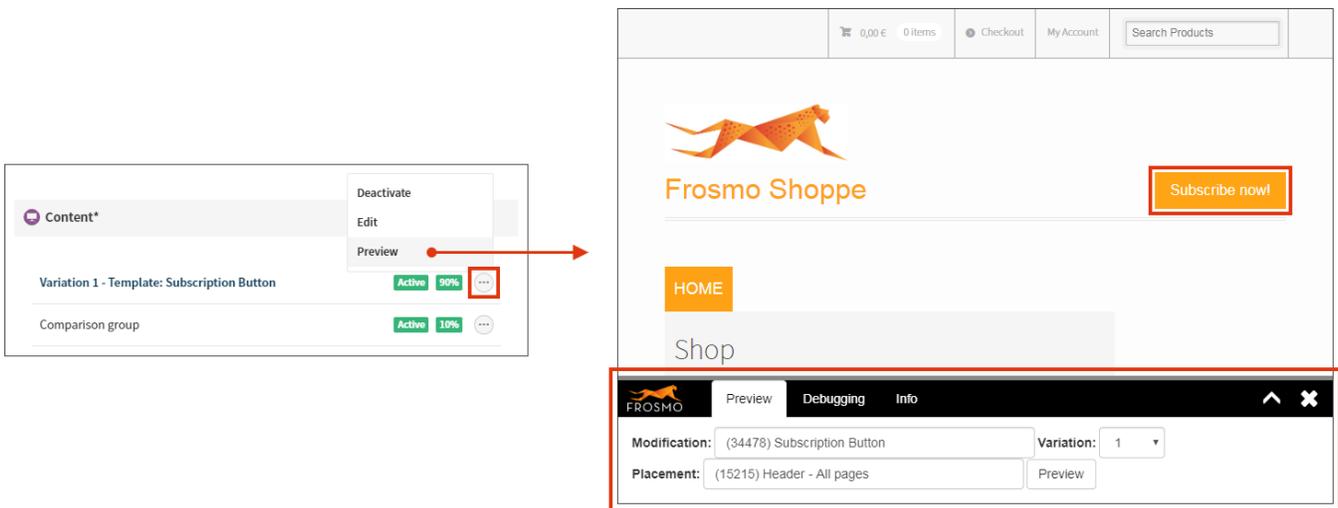

Figure: Previewing a variation from the Control Panel (click to enlarge)

New look-and-feel

The layout and organization of the modification UI has been overhauled to better serve the workflow, and the visual style has been updated to match the rest of the Control Panel. The UI also now uses the sliding modal windows introduced for other key features in the Goodnews Bay release.
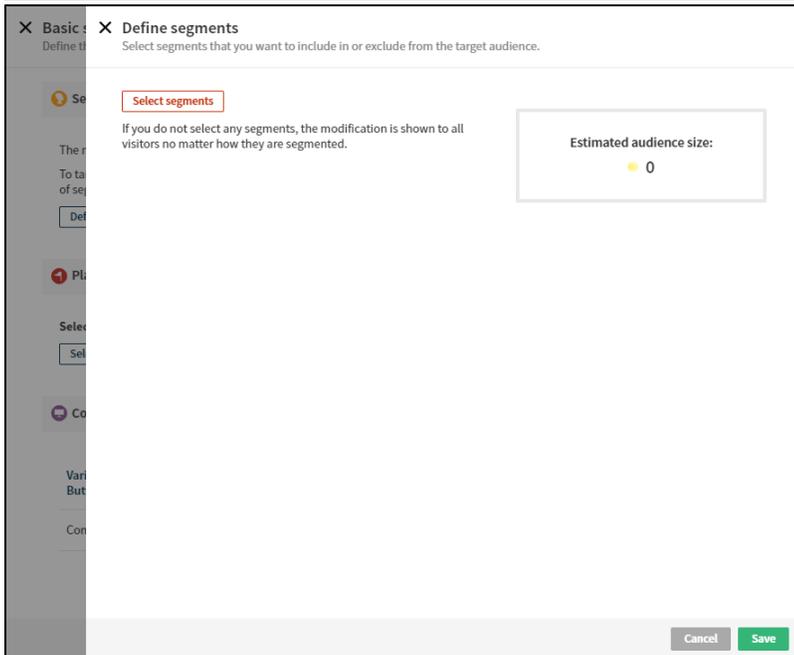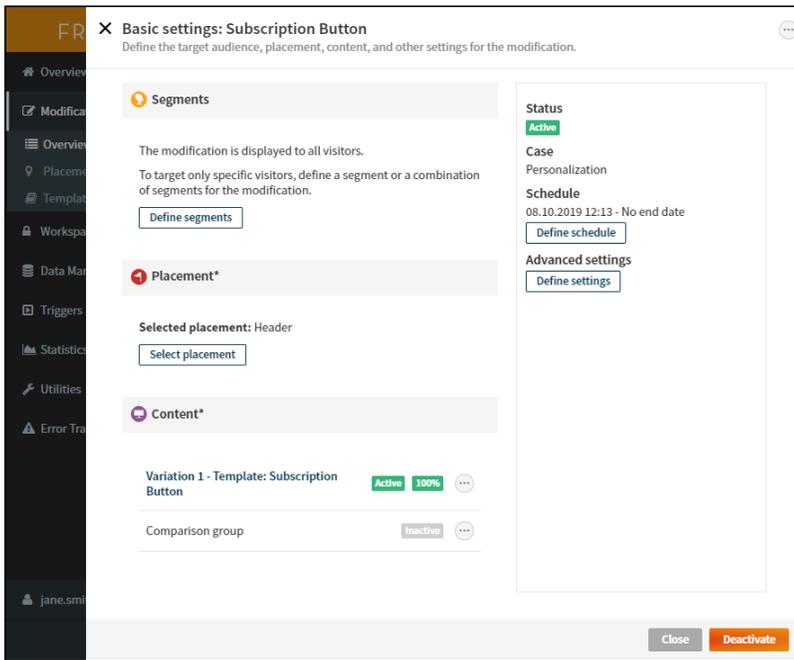
Figure: Modal windows in action (click to enlarge)

Improved template workflow

Accompanying the new modification UI, creating and using templates in the Control Panel now follows a workflow that more clearly separates the template work from the modification work. In this new workflow, the coding happens when creating and editing a template, while using the template in a modification mostly involves filling out predefined fields in a graphical user interface.

In other words, as a rule, you no longer need to write JSON code for a variation that uses a template. This makes it easier to have developers focus on templates and non-developers on modifications. The only exception to this are templates that do not define a content options schema but that nonetheless require content options to be set. For these templates, you still need to define a JSON object in the variation.

Learn more about how the separation works in the Control Panel:

For more information about templates, see the templates user guide.

⚠ The improved template workflow will not be available to all customers on Happy Valley release. For more information, see Availability on release.

New in templates: Schemas

To support the new workflow, templates now include two new settings:

- **Content options schema** describes any content options that must be set for template content before the content can be rendered on a page.

  If the template content includes Mustache tags as placeholder elements, those tags must be replaced with specific values in every variation that uses the template. Content options are the mechanism for defining the specific values. The Control Panel uses the content options schema to generate the **content options UI** of a variation, where the user can then provide the specific values for that variation.
- **Content options UI schema** allows you to further customize the content options UI generated from the content options schema.

  For example, if one of your content options is a hexadecimal color code, the content options schema only allows you to provide a simple text field for the user to enter the color code. With the UI schema, however, you can turn that text field into a color picker widget, allowing the user to easily select the color they want without any need for hexadecimal codes.

Both schemas use JSON Schema.

The template editor also provides a live, interactive preview of the content options UI as rendered based on the content options schema and content options UI schema.

Content options schema

If the template content must be customized in a modification variation, define the appropriate content options as a JSON schema. The content options UI of the variation is generated based on this schema.

```
 1 ▾ {
 2       "type": "object",
 3 ▾     "properties": {
 4 ▾         "bannerContent": {
 5               "type": "object",
 6               "title": "Banner content",
 7 ▾             "properties": {
 8 ▾                 "heading": {
 9                       "type": "string",
10                       "title": "Heading"
11                   },
12 ▾                 "body": {
13                       "type": "string",
14                       "title": "Body"
15                   }
16               },
17 ▾             "required": [
18                   "body"
```

Content options UI schema

If you want to customize the content options UI, define a UI schema for it.

```
 1 ▾ {
 2 ▾     "bannerContent": {
 3 ▾         "heading": {
 4               "ui:placeholder": "Enter the banner heading text"
 5           },
 6 ▾         "body": {
 7               "ui:placeholder": "Enter the banner body text",
 8               "ui:widget": "textarea",
 9 ▾             "ui:options": {
10                   "rows": 2
11               }
12           }
13       },
14 ▾     "bannerCta": {
15 ▾         "buttonText": {
16               "ui:placeholder": "Enter the button text"
17           },
18 ▾         "buttonUrl": {
```

Content options UI preview

The following is a live, interactive preview of the content options UI as rendered based on the content options schema and content options UI schema. Below the UI, is a preview of the JSON object generated from the values set in the UI.

Banner content

Heading

My Hero Banner

Body*

Enter the banner body text

Banner CTA

Button text*

Enter the button text

Button URL*

Enter the URL that opens when a visitor clicks the button

Banner colors

Background color

Heading and body text color
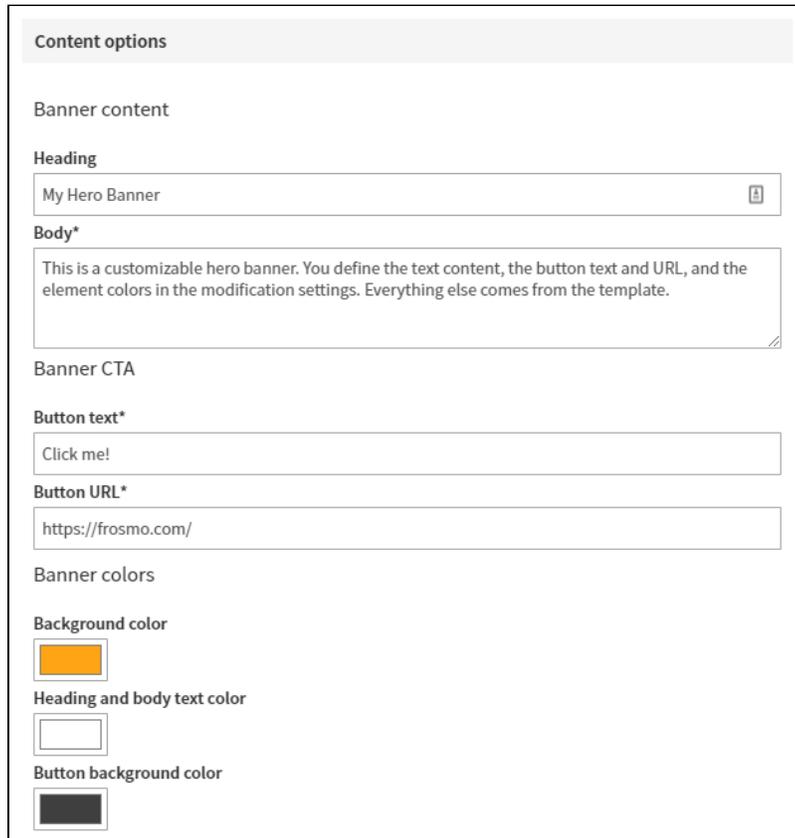
Button background color

Figure: Content options schema and content options UI schema, and content options UI preview (click to enlarge)

New in modifications: Content options UI

In the old modification UI, if a template used Mustache tags, not only did you have to provide the replacement values ("template defaults") in a JSON object in the variation, but you also had to know what the correct JSON properties were. The UI provided no information about the expected values.

In the new modification UI, the variation settings include a graphical content options UI, which allows you to easily define whatever values the template content requires to be defined. Under the hood, the template still uses a JSON object, which the content options UI generates, but the code is completely hidden in the modification UI.

Figure: Content options UI for a variation (click to enlarge)

Availability on release

The new modification UI and, by extension, the improved template workflow will not be available to all customers on Happy Valley release. The new UI is incompatible with a custom template solution used in some modifications on certain sites, which means that the new UI cannot yet be enabled for companies to which those sites belong.

We're working to convert the affected modifications to use the standard template solution. Once we've converted all modifications for a company, we'll enable the new UI for that company. In the meanwhile, the company users have access to the old modification UI.

If you do not see the new UI in the Control Panel on release day, this issue affects your company. If you have any questions, contact your Frosmo representative.

Bug fixes

- Fixed a bug in the old modification UI where, if you created a new basic modification and assigned a template to a variation, the template content was not added to the custom script if the export was set to **Automatic** and the template was assigned to the variation before activating the modification.