

# Dashboard Creation

- [Dashboard creation basics](#)
- [Dashboard templates](#)
- [Dual modes](#)
- [Developer-centric dashboard editing](#)

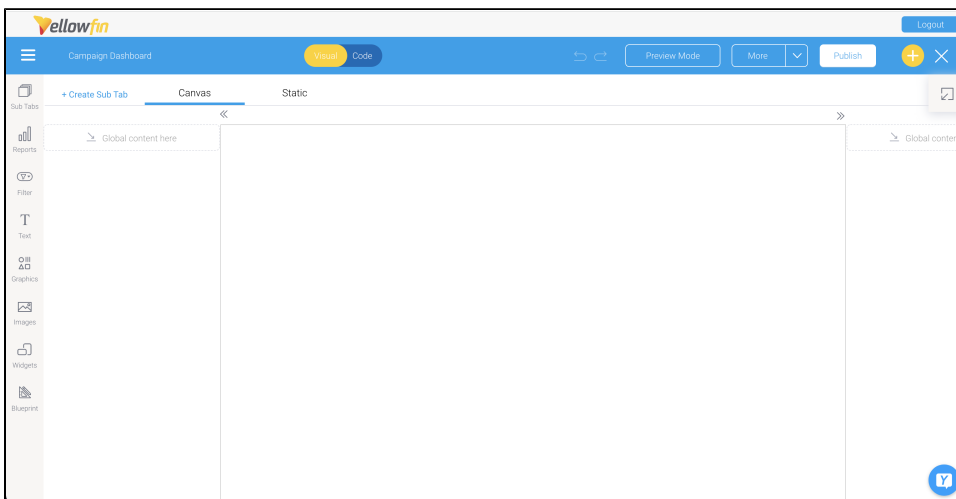
## Dashboard creation basics

In this section, we will cover some of the basics around creating a Dashboard.

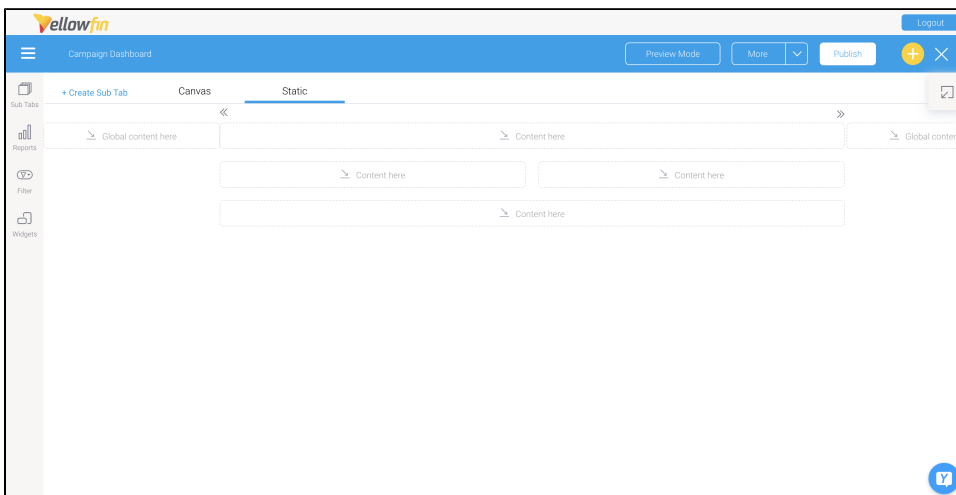
## Dashboard templates

When creating a new dashboard tab, you can choose a pre-built layout or a canvas. Here's the difference:

- **Dashboard Canvas:** A flexible template that allows you to create a pixel-perfect design. There are no restrictions on object positioning so they can be positioned anywhere. Support for all dashboard content objects is provided on the canvas, along with functionality that allows authorized developers to create a highly customizable user experience.



- **Dashboard static layouts:** A range of pre-designed dashboard templates with established layouts, where you can drop in objects. These layouts have less flexibility in positioning objects compared to canvas, and are simpler to build. Objects such as reports and filters can be added to these layouts.



**Note:** Your dashboard can contain multiple tabs, and each tab can use a different template. That means that canvas tabs can be built alongside layout tabs.

## Dual modes

Within the dashboard editor, there are two types of modes used to design and create a dashboard and its content. These are:

- **Visual Mode:** Shows a visual representation of the dashboard. Allows users to drag, drop and configure objects while this mode is enabled.
- **Code Mode:** Swap between visual and code modes with the bilateral editor and edit dashboard code with HTML, JavaScript and CSS. Authorized developers can extend dashboard functionality within Yellowfin and create rich application experiences. Learn more [here](#). Note that Code Mode is supported only on the dashboard canvas.

Both editor modes work seamlessly together, so changes made in one mode will immediately be reflected in the other one. This allows users to use them interchangeably and build a dashboard suitable for any purpose. These two types of modes help users of varying capabilities to come up with their ultimate dashboard design.

## Developer-centric dashboard editing

Developers can create rich application experiences by extending functionality with Code Mode, code widgets, and actions buttons. Only authorized developers will be able to edit dashboard code using HTML, JavaScript, and CSS.

Note that this coding functionality is only available on dashboard canvases; it is not supported on dashboard layouts.

Refer to any of our wiki sections for more information on these:

- [Code Mode](#)
- [Code widgets](#)
- [Action buttons](#)