

# Theme Settings

The primary files that determine theme settings are located in the theme root directory: `theme.json` and `theme-ui.json`. Theme settings determine options for your site such as metadata, default fonts, product image sizes, content types, and more. You can choose to expose individual settings for configuration in Admin or leave their configuration up to theme developers.

## theme.json

This file contains the majority of theme settings. Many of the settings are provided by default in the Kibo Core Theme, but you can modify any setting or add additional settings as you choose. By default, the settings in `theme.json` are organized into the following top-level arrays:

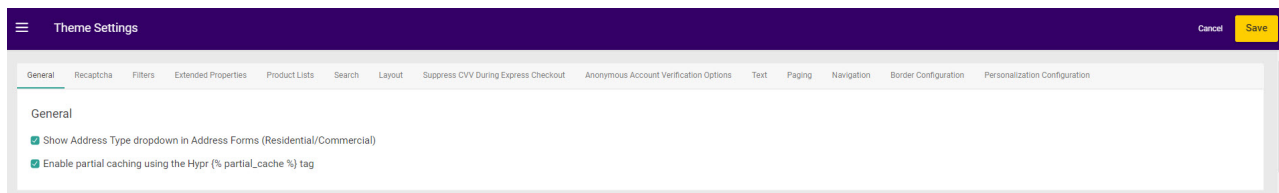
Array	Description
<code>about</code>	Metadata about the theme such as the author, theme name, default language, and device support.
<code>editors</code>	A collection of editors available for the theme, which expose configuration menus in Admin.
<code>emailTemplates</code>	A collection of email templates available for the theme.
<code>backOfficeTemplates</code>	A collection of back-office templates available for the theme, such as order details or packaging slips.
<code>pageTypes</code>	A collection of page types available for the theme, such as "About Us", "Landing Page", "Category", "Cart", etc.
<code>settings</code>	A variety of settings for the theme, such as default sort options, font size, image settings, page size options, product details options, cart behavior, search preferences, and more.
<code>widgets</code>	A collection of content widgets available for the theme, with their relevant configuration data, such as the display name, widget icon, and view data (either a custom editor or an edit view field) for configuring widgets.
<code>layoutWidgets</code>	A collection of layout widgets, with their relevant configuration data, designed to create variable-width columns for page content.

You can view the contents of the latest `theme.json` file on [Github](#).

## theme-ui.json

This file contains a collection of menus that allow merchants to configure theme settings through Admin. In `theme-ui.json`, you set up `xtypes` such as a `panel` or `mz-input-checkbox` to build the menus that Admin users can later configure by going to **Main > Content > Themes**, clicking the dots on a theme, and selecting **Settings**. The following code is an example of options that show up under the "General" panel for theme settings:

```
{
  "xtype": "panel",
  "title": "General",
  "collapsed": false,
  "items": [
    {
      "xtype": "mz-input-checkbox",
      "name": "showAddressTypeDropdown",
      "fieldLabel": "Show Address Type dropdown in Address Forms (Residential/Commercial)"
    },
    {
      "xtype": "mz-input-checkbox",
      "name": "enablePartialCaching",
      "fieldLabel": "Enable partial caching using the Hypr {% partial_cache %} tag"
    }
  ]
}
```



To make the `theme-ui.json` settings accessible to the rest of your theme, add the same settings to the `settings` object in `theme.json`, making sure to provide default values for the setting for cases when merchants do not provide values of their own. For example, if a couple of inputs you add to `theme-ui.json` are named `settingA` and `settingB`, you would add these settings to `theme.json` as follows:

```
"settings": {
  "settingA": "default",
  "settingB": false,
  ...
}
```

When Admin users configure settings using the menus you create in `theme-ui.json`, Kibo saves these settings in a CMS document that merges with the settings in `theme.json`. If a setting is set through both `theme.json` and a menu provided by `theme-ui.json`, Kibo uses the value of the latter.

## Access Theme Settings in Hypr Templates

Use the `themeSettings` variable to access theme settings from `theme.json` in a Hypr template. For example:

```
templates/modules/example.hypr
```

```
{% for size in themeSettings.pageSizeOptions %}
...
{% endfor %}
```

## Access Theme Settings in Stylesheets

Theme settings are available in stylesheets through the `@theme-settings-settingName` LESS variable. You can access any theme setting from your `theme.json` file using this syntax, as shown in the following example, where the font size set in `theme.json` is set to the value of another variable for organizational purposes in your stylesheets:

```
stylesheets/base/variables.less
```

```
@fontSize: @theme-settings-fontSize;
```

## Access Theme Settings in JavaScript Files

To access theme settings in a script file, require `hyprlive` in the `define` function and then access the theme setting using the `Hypr.getThemeSetting('settingName')` syntax within a function. The following example shows the beginning of the `scroll-nav.js` script, which accesses the `gutterWidth` theme setting:

```
define(['modules/jquery-mozu', 'hyprlive', 'underscore', 'modules/api', 'shim!vendor/bootstrap/js/affix[jquery=jQuery]', 'shim!vendor/bootstrap/js/scrollspy[jquery=jQuery]'], function ($, Hypr, _, api) {
  {
    if (!Modernizr.mq('(max-width: 800px)')) {
      var gutterWidth = parseInt(Hypr.getThemeSetting('gutterWidth'), 10);
      $(document).ready(function () {
        ...
      });
    }
  }
});
```