Master Theme LESS Guide

ion has created a Theme template using LESS for developer use. LESS is a CSS pre-processor that extends the features of CSS and allows many additional functions rather than just coding in CSS. The most important feature that LESS has are variables. These variables allow you to utilize the same piece of code, say a color, throughout the entire Theme without having to update every instance of it.

If you would like more information on LESS and using a CSS pre-processor, here are some external links:

http://lesscss.org/

http://www.developerdrive.com/2012/04/learning-less-an-introduction/

https://htmlmag.com/article/an-introduction-to-css-preprocessors-sass-less-stylus

To get started with LESS, you will need two types of programs.

First, you will need a code editor. Some commonly used programs are:

- Brackets
- Visual Code
- Dreamweaver
- Sublime Text
- Atom

Second, you will need a code compiler. CSS pre-processors require that the code be compiled into a single CSS file to be read by the browser. There are many options like:

- CodeKit
- Koala
- SimpLESS
- Prepos

There are a variety of code editors and compilers out there, both paid and free. Your personal preference will determine the best option for you.

Master Theme LESS Structure

The Master Theme LESS file is a template to create CSS stylesheets for your creatives built in the ion platform. Each component of the Theme is broken out into individual LESS files— from text colors to tab style. All of the individual, component specific LESS files are set up to import into Theme.less, brand-form.less and ThemeForm.less to make up your Theme. Using a compiler, you can then convert those files to css before uploading to your ion console.

Below you will find a description of all LESS files in the Master Theme in alphabetical order:

accordion.less: This file contains the styling for the Accordion component. This file allows you to style the different states of the accordion. There is a default style and also a required secondary style.

backgrounds.less: This file contains the styles of background colors, gradients, and images that can be applied to container and container-like elements when publishing within the ion platform. The first three Micro-Themes are required and should always be White, Transparent White, and Transparent Black.

buttons.less: This file contains the styles for buttons, which can be applied to text and link elements.

colors.less: This file contains the different colors you can use to style text and link elements.

custom.less: This file allows you to add in any additional custom CSS to your Theme. For example, @font-face declarations or additional styles.

directional-buttons.less: This file contains the styles that allow text and link elements to transform into directional arrows.

flow.less: This file contains the styling for the Flow component. It includes styles for Flow Dots, Flow Numbers, and Flow Progress Bar.

form-brand.less: This file contains the styling for the Freestyle Form component. It allows your form fields, dropdowns, radio buttons, checkboxes and more to be stylized to match your brand.

generic.less: This file is the first file pulled into the the Theme.less and contains your general body and html settings.

images.less: This file contains Micro-Themes for image alignment.

ixp.less: This file contains mix-ins used across the other files to pull in the ixp-scopes used by the ion platform. The scopes in this file should not be removed as they will cause your code to not compile. You may add additional ixp-scope mix-ins to this file as necessary.

links.less: This file contains special styles for links.

lists.less: This file contains the styles for unordered lists and ordered lists (bulleted lists).

logo.less: This file contains the definitions for your logo Micro-Themes.

mixins.less: This file contains mix-ins used throughout the LESS files. These should not be edited or removed. If you need additional mix-ins for your Theme, they can be added here.

nav.less: This file contains the styling for the Navigation component. There is a default style and also a required secondary style.

pods.less: This file contains styles used on containers. Pods include styling for padding, background color, text color, headline text color, link color, link hover color, and generic button overrides. Additional CSS could be added for pods to have additional styling such as box-shadows.

regions.less: This file contains styles used on containers that are specific to regions on the page — such as header and footer. ion defines 8 different region types:

- pre-header
- header
- inner-content-wrapper
- pre-content
- content
- post-content
- footer
- post-footer

These styles are defined here. Similar to pods, they style padding, background color, text color, headline text color, link color, link hover color, and generic button overrides.

settings.less: This file is the core of the Master Theme. Here you will define all of the variables that are used throughout the different files. For example, when you change your color palette in the settings file, it will change across all files. For more details on the settings.less file, see below.

styling.less: This file contains styles for containers and images, such as rounded corners and shadows.

tabs.less: This file contains the styling for the Tabs component. Here you can style the different states of your tabs and how they stack responsively. There is a default style and also a required secondary style.

Theme.less: This file imports all of the necessary individual, component specific LESS files and compiles them into the Theme.css file. This also contains your brand name and creator initials to help with versioning of your Theme.

ThemeForm.less: This file determines the styles for your library forms in your ion console.

typography.less: This file styles all headlines and contains the Micro-Themes for fonts.

webfonts.txt: This file allows you to call additional CSS stylesheets or JS files into your Theme. This is typically used to incorporate web fonts into your Theme. When utilizing CSS stylesheets, be sure to remove the beginning ‘http’ or ‘https’ from the link to ensure that any browser security settings will not be triggered. When using the less files, ensure that the reference to Font Awesome stylesheet stays as many icons utilized in the Theme are found in this file.

Editing The LESS Files

The bulk of your work will be done in the settings.less file. In fact, you could style a Theme and only touch the settings.less file!

There are many settings and styles to adjust in the LESS files, just as if you were to code a Theme in CSS from scratch. The LESS files provide a great starting point that can help you code Themes and understand the ion Theme structure with less effort.

Page Defaults

The first section in the LESS files defines some general page defaults — such as your base font size, default font, and your brand name.

Fonts

The Fonts section defines your font name and font-sizes for headlines. There are additional headline sizes for XS screens that are based on best practices determined by ion. We suggest you do not adjust these, or if you do, that you keep your font sizes similar to the guidelines.

Swatches

This is where you will determine your color palette. The Master Theme accounts for up to 9 colors in addition to black and white, though you can add more by adding additional color settings. The notes in this file help to instruct what colors this refers to in the Quick Starts 2 Theme. By ensuring your colors are organized similarly to the Quick Starts 2 Theme, you will find that your Theme applies more easily to our Quick Start Cloud templates. For instance, if your brand has a main brand color, it's best to put that in the color-a slot as color-a is used as the primary color in the Quick Start Cloud templates.

In this section, you will also define the error, active, and placeholder color for your forms.

Pods

The pods sections allows you to define the background color, text color, link color, and link-color hover of all of your pod elements. All pods will use the same padding defined at the top of this section unless you choose to go into the pods.less file and override them.

Styling

This is where you will set the definitions for the styles available to containers and pods — such as Rounded Corners, Drop Shadows, and Borders. Your horizontal rule color and height are also defined here.

Link - Default

This section allows you to define how link elements will look as a default when dragged onto your page. You can set hover states and define if your links are underlined or not.

CTA Links

CTA links are additional styles that can be applied to your links to allow them to have icons at the beginning and end, as well as certain colors. By default, we provide 4 different icon types of Arrow, Check, Link Out, and Caret. These styles utilize Font Awesome to define the icon. You can find additional CSS content values of Font Awesome icons here: http://astronautweb.co/snippet/font-awesome/

Button Styles

Here you will define your button styles. While only `.button-a` is a required style, we encourage you to add as many colors and styles as buttons as you like. As a suggestion, we provide 6 buttons - 2 main colors in solid and outline, white in solid and outline, and black in solid and outline.

Button Sizes

The Quick Start Cloud templates utilize 4 sizes of buttons — small, medium, large, and wide. All 4 sizes are required. We recommend offering the same variety of sizes to your designers. If your brand only allows one or two sizes, we suggest that you then put the same font size for two of the sizes. This way all Quick Start cloud templates will still pull correctly. If completely necessary, you can remove the ixp information in the buttons.less file so the styles are not selectable in the dropdown.

Button Icons

This allows you to put icons at the beginning or end of your button using Font Awesome. As a default, we provide a checked, arrow, back, and down button.

Directional Button Styles

These buttons allow you to make text and link elements into simple arrow or chevrons. You can change these to be larger, change their icon, or names in this area.

Bullets

These variables control the styles of your unordered and ordered lists, as well as additional styling of unordered lists utilizing Font Awesome so that you can have checkmarks as your bullet icon, for example.

Logo

These variables allow you to set an image (png, jpg, or svg) URL as a logo. This image will need to be uploaded to your Theme folder in the ion framework. The height and width should be the largest size of the logo you want to be shown. The `.logo-a-height-sm` and `.logo-a-width-sm` will be the size the logo reduces to in screen sizes 768px wide and smaller.

Regions

There are 8 common regions defined by ion: pre-header, header, inner-content-wrapper, pre-content, content, post-content, footer, and post-footer. Here you will style these regions by giving them padding, background colors, text colors, link colors, and link hover colors, similar to how you did above with Pods

Navigation

These variables allow you to adjust the default designs of the navigation component. There are many variables to adjust here. If you're looking to redesign the navigation, you will possibly need to dig into the nav.less to change styling further.

Accordion

Here you can size and style your accordions. The accordions utilize Font Awesome to pull in open and closed icons. There are variables for font size, padding, and margin that will apply across all your accordion Micro-Themes. The toggle, active, and hover state variables will apply to a specific accordion Micro-Theme.

Tabs

The tabs variables work similar to the accordion variables above. The first set of variables will apply to all tab Micro-Themes. The toggle, active, and hover variables are specific to each accordion Micro-Theme.

Flow Settings

These variables allow you to change the colors of the flow dots, progress bar, and numbers. By following the comments, the colors will be best optimized for use in the different settings by default. Any further styling adjustments will require editing in the flow.less file.

Freestyle Forms & Forms

These variables apply to both styles of forms and will affect both the ThemeForm.less and brand-form.less files. Here you will define your border color, input background, text color for forms, and more. Within these section there are various other variables you may change as necessary.

Textbox

This applies to textbox inputs in freestyle forms only. It utilizes the styles you have defined in the variables above. If you need to change the default styling that would be done here. For example, if on hover, you did not want the border to be the `@active` color, that could be adjusted here.

Slider

This applies to slider elements in freestyle forms only.

Choice Group, Choice Group - Border Box, Choice Group - Toggle Switch, Choice Group Sticker

This applies to choice groups and the specific styles available for choice groups. Most of the settings will not need to be changed unless you have specific guidelines that would adjust them. The images can be copied from the Quick Starts 2 Theme and are only for use as a fallback for IE8 and below.

Dropdown

Here you can change the default styles for your dropdown elements as well as the search and menu functions that appear when selecting a choice. If you want to change the arrow styling of a dropdown, you will need to do that in the form-brand.less file.

Library Form Drop Downs

As library forms are structured differently than ion's freestyle forms, they require some additional settings. The icon_select will be the arrow that shows when the drop down is being used. Most of these settings will not need to be adjusted.

Compiling and Uploading Your Theme

In order to upload your Theme into ion, you will need to use a code compiler to compile your LESS files into three CSS files — Theme.css, ThemeForm.css, and form-brand.css. These names will need to remain the same in order to be found and pulled by the ion platform.

You can copy the 'Quick Starts 2' Theme in the ion_framework_4.0 and rename it. You can then override the Theme.css, ThemeForm.css, and form-brand.css files by uploading them into the new Theme. Be sure to upload any new imagery and replace the logo.png file as well.

Additional Resources

Pixel to Ems

Helps to translate font-sizes set in pixels to web-friendly em sizing.

http://pxtoem.com/

Font Awesome CSS Content Cheat Sheet

Gives the font awesome icons as their CSS content values

http://astronautweb.co/snippet/font-awesome/

LESS Basic

Beginner's guide to less

http://www.hongkiat.com/blog/less-basic/