

Log in

Find

Updating a Tiki theme from Bootstrap 3 to 4

This document is a work in progress, as ~~Bootstrap 4 is still in beta at the time of this writing~~. But theme-related details are probably pretty stable so the method described here should continue to be useful.

The Bootstrap developers have taken pains to facilitate custom themes compared to what was offered for release 3, and this helps us as Tiki theme creators/maintainers. For an overview of (generic) Bootstrap theming, see <http://getbootstrap.com/docs/4.0/getting-started/theming/>.

Less to SCSS (Sass)

Bootstrap 4 uses SCSS precompiling rather than Less, so the workflow needs to be switched accordingly. As before, compiling can be done with PHP in the terminal in PhpStorm. Or the .scss files can be compiled by another method. In my work on this so far, I found that PHP compiling isn't very verbose when there are errors, and when updating or making a theme there tend to be a lot of errors in my experience.

However you can give it a try (it should work flawlessly most of the time):

```
$ php console.php scss:compile MyTheme
```

Assuming you have a `themes/MyTheme/scss/MyTheme.scss` file it should produce compiled CSS theme in `themes/MyTheme/css/MyTheme.css`.

Scout

There is an application called Scout (<http://scout-app.io/>), however, that can work nicely as part of the workflow to compile the theme files and get better error notices when things need to be fixed. It's free and available for MacOS, Linux, and Windows. After installing Scout, just add the theme as a project and Scout will watch for file changes as you edit the .scss files, and will compile in the background and specify the problematic file and line and nature of the problem when there's an error. (Then, if you're working on the Tiki project files so need to use PhpStorm for inter-developer consistency, when the compiling goes smoothly you can compile again in PhpStorm and commit the files.)

Rename MyTheme.less and update it

MyTheme.less will now be MyTheme.scss, and of course all the files it imports will have .scss extensions instead of .less. Tiki's base CSS files have already been through this process, so the import paths will be correct.

I've found that SCSS is not as forgiving as Less in regard to the order that files are imported and variables defined. To avoid compiling errors, this is the order of partials imports that has been successful for me:

```
@import "variables"; // Needs to come first, to override defaults.
@import "../../vendor_bundled/vendor/twbs/bootstrap/scss/bootstrap.scss";
@import "../../base_files/scss/_tiki-variables.scss"; // Values/definitions for Tiki variables
(outside of Bootstrap variables) such as _tiki-selectors.scss.
@import "../../base_files/scss/_tiki-selectors.scss";
@import "../../base_files/scss/_bs3-bs4-transition.scss"; // Temporary transition file
@import "_tiki-selectors.scss";
@import "../../base_files/scss/_external-scripts.scss";
```

Note: The file `_bs3-bs4-transition.scss` is a temporary file the content of which will eventually be relocated or redefined - it was just a kludge to get to a clean compile of the base files.

Prepare the variables file

The way I've been doing it, I rename the theme's bootstrap-variables.less file to `_bootstrap-variables.scss` (using here the SCSS convention of starting the name of the partial file with an underbar, to prevent it from being compiled into bootstrap-variables.css), and then delete everything below the file identification/comments lines at the top (I remove all the variables information). Then I paste in an outline of commented-out lines to provide some structure for the variables that I'm going to be adding back in. The outline looks like this:

[+] Initial outline in `_bootstrap-variables.less`

Find and input the variables to be used

If the Mytheme's bootstrap-variables.less file (or variables.less file) is the equivalent of the Bootstrap default variables.less file, but with values specific to the theme, then it's pretty easy to see which variables need to be used in the theme's variables.scss file for Bootstrap 4/Tiki 19. I generally use a Windows application called WinMerge to get the side-by-side diff view, but PhpStorm or another editor can be used, of course.

I go down the files, copying the variables that are changed from the Bootstrap default, and pasting them into the new `_bootstrap-variables.scss` file. Of course in SCSS variable names start with "\$" rather than "@", so this needs to be changed for each variable. Many of the Bootstrap 3 variable names are carried over to Bootstrap 4, but not all, so it's good to have a resource on hand that lists the new variable names. I check Bootstrap 4's `_variables.scss` file, for one thing.

So the variables to be used are pasted or input in the theme's `_bootstrap-variables.scss` file. Other changes need to be made for things like image paths, lighten/darken functions, including mixins, etc. I'll add a table of the ones I encounter most often in Tiki themes; the SCSS docs should also be checked for other cases.

Less function	SCSS equivalent
coming soon	coming soon

The tiki-selectors.scss file

The Less version of this file can be used as a basis for the SCSS version. That is, it doesn't have to be/shouldn't be wiped clean. Again, the Less variable names need to have their initial "@" replaced by "\$". This can be done with a global find and replace, but then any "\$import" will need to be changed back to "@import".

Corresponding with the CSS class name changes from Bootstrap 3 to 4, you will need to replace instances of "panel" with "card", "navbar-default" with "navbar-light", and so on. I'll add a table here of the changes I've had to make when updating Tiki themes, but in the meantime

https://www.quackit.com/bootstrap/bootstrap_4/differences_between_bootstrap_3_and_bootstrap_4.cfm is a good reference.

Links

- Easy to use and informative SCSS compiler - <http://scout-app.io/>
- Online Bootstrap 4 theme creator (useful for finding variable names) - <https://pikock.github.io/bootstrap-magic/>
- Differences Between Bootstrap 3 & 4 (including class names) - https://www.quackit.com/bootstrap/bootstrap_4/differences_between_bootstrap_3_and_bootstrap_4.cfm
- Outline of what has changed - <http://www.dotnetcurry.com/javascript/1334/migrating-bootstrap-3-to-bootstrap-4>
- Visual guide to what's new in Bootstrap 4 - <https://medium.com/wdstack/bootstrap-4-whats-new-visual-guide-c84dd81d8387>
- CSS Flexible Box Layout - https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout