

# Initiative - TwitterBootstrap Templating Kit

## Background

Since the new corporate website will run use Twitter Bootstrap, Tomas has done some investigation on how to effectively use Bootstrap with Magnolia templating. His first exploration was porting the existing to Demo Project and the existing STK components to Bootstrap. He made progress, but found it quite clunky and many styles had to be overridden. He found that the Bootstrap "responsive design, grid" approach is quite antithetical to some of the core principles of the STK. The next exploration was leaving the STK html behind - and taking a popular existing Twitter Bootstrap theme and creating a set of templates for it. However this still heavily leverages the model & dialogs of existing STK templates - but with totally new html and styles.

This has been "published" at:

Git: <https://git.magnolia-cms.com/gitweb/?p=user/tgregovsky/bootstrap-stk-2.git;a=summary>

## Meeting 2014.10.7

He presented his progress to Jan Shulte, Lars, Philip Mundt and Christopher Zimmermann.

He called the meeting to share what he had been working on, and to solicit feedback and suggestions for what would be the most useful direction to take it now. Attendees were impressed with the progress and excited about this new approach to templating.

## Suggestions

### Clean Modules

Break the work into cleanly scoped modules. One for the Twitter Bootstrap templates. Another Magnolia theme module for each of the Twitter Themes (Including the Modern one already implemented).

### Additional Themes

Create a few additional theme modules (3) - using very different existing themes. Just to verify and demo how easy it is to drop in new themes.

### Business Use-Cases

Create a samples module with page and component templates for a few common business use cases, such as:  
Landing page. Shop. Intranet. Control panel.

### Bootstrap Components

Create STK templates for a bootstrap component. For example: Tab, Dropdown, Carousel  
<http://getbootstrap.com/javascript/#tabs>

### Dynamic Theme

Leverage SASS or LESS (Magnolia already includes a SASS processor) to create a dynamic theme - one that can easily have its key colors, sizes, & typography customized by changing a minimum of values. Start with one file that contains these key theme values. (True enough values can be hand edited in CSS files - but SASS/LESS enable that you would change a value once - and it would be changed in many places in the CSS file. "DRY".)

### Dynamic Theme UI

Next (optional) step would be to create a simple UI for an editor to configure them - such as an App.

### Semantic Classes

A common complaint against responsive frameworks like Bootstrap or Foundation is that the html becomes "non-semantic" class names are just specifying how the layout should 'flex' instead of what the items are. There is a common solution to this which is using the SASS/LESS processing to put the responsive declarations in the css files instead of directly in the html markup. We should explore this.

<https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=bootstrap%20semantic%20markup>

<https://www.ostraining.com/blog/coding/bootstrap-right-way/>

<http://www.wearecube.ch/limitations-of-twitter-bootstrap/>

## Etcetera

Product teams plans for "Minimal Templating Kit" for 5.4 Release: <http://wiki.magnolia-cms.com/display/DEV/Concept+-+Minimal+Template+Kit>

Using SASS with Magnolia: <http://forums.magnolia-cms.com/forum/thread.html?threadId=b9771e9d-0f59-4228-b2fe-684d5e8d6bae>

"Bootstrap UI builders":

<https://jetstrap.com/>

<http://www.layoutit.com/>

<http://brix.io/>

<http://mashable.com/2013/10/20/bootstrap-editors/>