

Project
Software
Get / Try Tiki
Help
Get Involved
News

Log in

Home Page
Tiki Customization
Feature Examples

Find

Bootstrap Transition Preps

See: [TikiFestBootstrap](#) and [Planning for Tiki 13](#)

As part of [Theme Revamp](#), this page is to discuss, and plan an eventual transition to using Bootstrap in Tiki. As there is a general consensus we should move in the general Bootstrap direction (the minimum being the ability to use Bootstrap themes), the remaining questions are no longer "if" but more "how" and "when".

An experimental branch is here:

<http://sourceforge.net/p/tikiwiki/code/HEAD/tree//branches/experimental/redesign-proof-of-concept/>

- [Questions](#)
- [Evolution or revolution?](#)
 - [Arguments for clean break](#)
 - [Arguments for legacy support](#)
- [Things to watch out for](#)
 - [Things that Bootstrap doesn't cover](#)
 - [Things that Bootstrap overlaps with existing components](#)
 - [Pre-requisites](#)
 - [In general](#)
 - [CSS file reorganization](#)
 - [Coming in Bootstrap version 3.0](#)
- [Workflow](#)
- [Related links](#)

Questions

- Where will we coordinate work?
 - I think on themes.tiki.org

- I think dev.tiki.org would be more appropriate as it is about development (not a themes how to and tips and tricks stuff)
 - #But then we wouldn't be able to enjoy the retro environment of Planetfall or Faulkner, etc. 😊
 - as far as I see it, coordination already started on themes.t.o and we should stay here as facts have been made - earlier on I would have opted for dev.t.o as well
- ~~Bootstrap 2 and later 3, or directly to Bootstrap 3?~~
 - Bootstrap 3.1 is decided in the meantime and will apply licence wise as it will be duallicensed to MIT
 - Licensing issue prevents us use Bootstrap 2
 - There are several non backward-compatible changes
 - I suggest we shoot for compatibility with Bootstrap 3 to avoid doing a lot of work twice.
 - I agree
- What about LTS handling?
 - Option 1: start revamp after Tiki12 LTS
 - Benefit: We can do pre-requisites such as [Menu Revamp](#)
 - Benefit: So we have less work later, we can start with migration of some aspects like
 - <http://twitter.github.io/bootstrap/customize.html#plugins>
 - Some of the themes (ex.: we should have a FiveAlive for Bootstrap)
 - Benefit: We can start directly with v3
 - Benefit: Community has several months to get accustomed to Bootstrap
 - Benefit: No change on the schedule
 - Drawback: First LTS with Bootstrap will be Tiki15
 - Drawback: We could have started earlier
 - Option 2: start after 11.0 and try to finish for Tiki12
 - If Tiki12 isn't stable enough for a release on schedule, either
 - Option 2a) Retroactively identify Tiki11 as LTS
 - Option 2b) Identify Tiki13 as LTS
 - Option 3: Experimental option in 12 and rocksolid default in 13?
- Do we try to support [Bootstrap](#) and <https://www.openhub.net/p/zurb/Foundation> as well?
 - => in my mind that depends on how we structure the /templates directory
- Will we still support the core/existing customisation approach that Tiki has - even with a Bootstrap theme (I'm probably showing my ignorance here! - Geoff)

I see no reason for not keeping the core/existing customisation approach and my first tests with bootstrap show me, that there is no technical problem with it

... - Geoff) e.g

 - being able to take a standard theme but apply an 'option' (/styles/theme_name/options) that tweaks/adds different css etc., which includes:
 - option specific images
 - the Newsletter specific css
 - etc.
 - how will quirky browser tweaks be handled eg ie7, ie8 files etc
 - having a set of .tpl's that are theme specific (/templates) - with customised mail templates folder
 - be able to apply theme specific modules

[My view is that we should use Bootstrap to provide:

- Uniform styling (classes) for page components such as gadgets, buttons, tabs, and tables, etc. ("content"), and

— (First we'll need to clean up Tiki's tpl files to uniformize the construction of these things.)

- Page layout classes for grids, responsive web design, etc. ("framework")
- Tiki's CSS classes should be a superset of the Bootstrap classes IMO.
- A quick check of more-interesting "made with Bootstrap" sites shows this is common practice.
- Sites using only default Bootstrap classes seem to be pretty limited style-wise.
- We can offer more flexibility to designers and reduce their work by adding classes for styling.
- We can offer more page layout options by having a few more divs available (to support FiveAlive-style header and footer graphic treatment, etc.).]

Evolution or revolution?

Do we go for a clean break / "burn all bridges" approach and force major changes on all legacy themes? or do we make an effort to try to make old themes still work?

Arguments for clean break

- End result will be simpler code
- End result will be closer to what people already know (people that use Bootstrap that is)
- Most theme intense sites have more than enough features in the LTS versions and can use that until they decide to refresh their site design.

Arguments for legacy support

- Makes it easier for heavily customized themes to upgrade
[I'm not sure how important legacy support per se is, but I think it'll be good to have a framework available that enables people to replicate their old designs if they want to go to the trouble.]

Things to watch out for

Things that Bootstrap doesn't cover

- Logo?
 - Presumably, Tiki's HTML classes will be a superset of the Bootstrap classes, so our current method for specifying the logo can still be used, in combination with the benefit of Bootstrap's layout.
- RTL is not in v2
 - Licensing issue prevents us use Bootstrap 2 anyway (see <http://www.dwheeler.com/essays/floss-license-slide.html>)
- Let's use <http://schema.org/> when possible
 - I do not understand how do Schema attributes validate in HTML - luci
 - luci, I think the suggestion was to use them for element/class names, etc.

Things that Bootstrap overlaps with existing components

- <http://twitter.github.io/bootstrap/customize.html#plugins>

- jQueryUI
- jQuery Mobile

Pre-requisites

This is a list of things that must or should be done before or as part of the Bootstrap integration.

- [Menu revamp](#)
- Migrate more things to Smarty plugins
- New space /themes/ directory (but old way still works)
- Move all temp content to one place
- Layout switcher (from experimental branch)
- Change to use Bootstrap [font icons, which can be themed](#)

In general

- This is a lot of work
 - True, but Tiki's templates and CSS really need a major overhaul anyway. If that happens in a way that can take advantage of the Bootstrap bandwagon (as long as we still have design flexibility), so much the better.

CSS file reorganization

- In parallel with the Smarty templates cleanup, CSS selector globalization, and use of [Less-CSS](#), I suggest reorganizing Tiki's selectors into new "modular" files, which would be Less includes, perhaps similar to the [breakdown](#) in Bootstrap (not all will be needed: perhaps others will be).

Coming in Bootstrap version 3.0

- No set release date, but probably sometime in 2013
 - ["As we've previously mentioned, v2.3 is our last planned release before moving onto v3 fulltime"](#) - 2013-02-07
- Many changes in class names
- [Dropping submenu support](#)

Workflow

PHPStorm has a Less plugin (<http://plugins.jetbrains.com/plugin?pr=&pluginId=7059>), and apparently supports an "import" feature so there can be modular .less (pre-compilation) files like "typography.less" that can be imported into the main theme CSS file rather than being compiled separately (similar to how SASS supports "partials"). This should be confirmed.

- Some references:
 - <http://jeffcroft.com/blog/2012/feb/23/many-ways-to-use-css-preprocessors/>
 - <http://coding.smashingmagazine.com/2013/03/12/customizing-bootstrap/>

Related links

- [Bootstrap Themes Transition](#)
- <http://tiki.org/forumthread46596>
- <http://tiki.org/forumthread46597>

Alias

[BS Transition Preps](#)