 OBMP

UI Design/Development

# OBMP app - UI overview

## Built with ANGULAR, BOOTSTRAP, CSS 2/3, SCSS, jQuery, JavaScript, D3.js, nvd3.js (for charts), Leaflet maps and Mapbox

## The OBMP charts are developed with nvd3 and d3.js – we will cover the file structure, layout, elements, styling, best practices and testing.

## OBMP app is not responsive yet.

## Many interns and engineers have contributed to the app. You will notice inconsistencies. Feel free to make fixes and suggestions.

## Atom is a good editor to use as well as Sublime. Atom comes with many great plug ins. Sublime requires you to use a package controller to add plugins.

# file STRUCTURE

## Section files from left-side main navigation bar

#### HTML and CSS/Sass, JS files are located in the [app] directory

### Each section or area of the OBMP app has its own directory under the [views] directory. Each directory should contain at least 1 html, Sass, and JS file

#### For example, the Dashboard > Global View section files are located in [views/globalView] files: globalView.html, globalView, globalView.js

## Main content in body (right side of navigation bar)

### The scaffold/layout of the main section/body is located in the [view/container] directory. Files are container.html, container.scss, and container.js

### The container.html file also contains the top navigation bar where the breadcrumbs are located to the left and the admin navigation panel is located to the right

## Fonts are located in the [fonts] directory. Currently using Cisco fonts.

## Icons are located and imported from the bootstrap library. We will be adding Font Awesome soon in order to have a larger icon library. Here is the Font Awesome website: <http://fontawesome.io/icons/>

## Images directory contains only a few files. The logo is located in app/images. Note: All charts and maps are dynamic.

## Leaflet maps: <http://leafletjs.com/> - located under bower components

## Mapbox: <https://www.mapbox.com/> - located under bower components

## D3 and nvd3 dependencies are located under bower components for charts are located under bower components. <https://d3js.org/> and <http://nvd3.org/>

# Sass and css files

All scss files contain sass syntax. Once the .scss file is saved it s exported to the main.scss file located at in the [views] directory root level. Then the main.scss get compiled into the main.css file located at .tmp /styles/main.css

When you save a sass file it will automatically run a build. It takes about 30 seconds to see the change on your local server. If you find you have made an error the browser will indicate what line the syntax error is locate. Once the fix has been made you can save the file, run a build (automatically) and the browser automatically will refresh to render the style changes

* Sass site: <http://sass-lang.com/>
* Here is a good Sass tutorial: <https://www.codeschool.com/courses/assembling-sass>
* HTML5 and CSS dev standards guide: <http://codeguide.co/>
* Other resources for dev standards other CTAO teams follow these standards. Our stack is different, but this is a good reference: <https://cto-github.cisco.com/CTAO-BMP/dev-standards>
* When creating new scss files you will need to add a path to import it to the main.scss [views/main.scss]

# bootstrap grid system – layout and other elements

OBMP uses bootstrap grids for the layout of columns

* More info on how Bootstrap grids are implemented: <http://getbootstrap.com/css/#grid>
* We can use other elements from the bootstrap framework for
* The buttons on OBMP are not consistent, but does use bootstrap buttons: [http://getbootstrap.com/css/ - buttons](http://getbootstrap.com/css/#buttons). We are currently cleaning up the app. Currently there is no button factory
* OBMP does not have forms. Only search fields. When we receive a requirement to use a form we can use bootstrap forms: [http://getbootstrap.com/css/ - forms](http://getbootstrap.com/css/#forms)

# Maps

Maps are located in [view/components]

* Global View and Peer View maps are split into two sections
* Map view and table view
* We have been redesigning the diagrams that are associated with the data from each map/table
* Both maps have donut charts for Routers and Peers and are implemented with nvd3.js

# Best practices, Naming conventions for CSS selectors, Semantic HTML5

OBMP has been designed by many developers. You will come across several inconsistencies as you implement designs. Feel free to clean up any code that needs refactoring. However, use caution as many selectors are reused throughout the site. It would be best to become familiar with the app before refactoring occurs.

There are currently no naming conventions outside of bootstrap html markup and css, so use best judgement when naming selectors. Avoid basic selector names like “box” or using names that are typically used as an <html> element. Also, avoid using html elements for any wrapping and keep those elements for their intended purpose.

Here is an example of bad usage: There is a “p” tag that is currently being used for a donut chart wrapper. Although the donut chart that is wrapped inside of the “p” tag works, it is not best practice or semantic HTML.

You will also notice the use of “pt” as a unit of measure. The preferred units of measure are “px” and “rem” for fonts, and “px” for CSS box models width and height dimensions.

* Here is an example of a proper selector naming when using buttons from the bootstrap library: label-success, label-danger, label-info
* Avoid adding inline styles to html mark up tags. Inline style example: - Please add style classes to scss files.
* The site uses <div> tags for container content and wrapping
* For inline elements use <span> tags
* Use semantic html when necessary. For a quick review go to : <http://www.w3schools.com/html/html5_semantic_elements.asp>

**Testing:** Once you commit to repo log into dev: <http://bmp-dev.openbmp.org:8001/#/login> and Test your code: username: **openbmp**, password: **CiscoRA**

* Test your changes in the Chrome browser. Use the DOM inspector to see what could be going wrong with your styles if they are not rendering as desired. In Chrome press: ‘Option’ ‘Command’ ‘i’ to access the DOM inspector. Then select the “Elements” tab to inspect html markup/nodes and css selectors.

**Important note!** When developing and testing on your local server you will need to change the following line of code in the [factory/apiFacory.js] file in order to view the data for maps, charts, and tables.

