## Twitter Bootstrap

Next we’ll be looking at Twitter Bootstrap in more depth. Bootstrap is free library consisting mostly of CSS and some JavaScript. Bootstrap was developed by Twitter but is now used on numerous websites. It is one of the most popular libraries in use today for creating user interfaces including menus, forms and tables. After completing this section on Bootstrap you’ll start to recognize the menus and other distinctive Bootstrap UI components on quite a few websites.

For this section we’ll be using examples in the 02\_bootstrap project folder. To follow along you’ll need to set up the project folder and install the Node.js dependencies. If you haven’t already done so, download the 02\_bootstrap example project. Next, open a console window and navigate to the folder and execute the following command.

…02\_bootstrap > npm install

As previously described, this command will use the package.json file to install any dependencies and create a node\_modules directory. You can test that everything has been installed correctly by running the command:

…02\_boostrap > node 02\_logger.js

You should then see a message stating that the server is listening on port 3000.

## Node.js Server Logging

Before continuing, let’s take a brief look at logging on the Node.js server. To add a simple logger to the server we can add the following line to our server:

app.use(express.logger())

As previously mentioned, Express is an extension of the Cy.onnect package from Sencha Labs. The logger is actually part of Connect. You can find out more details on what options are available at <http://www.senchalabs.org/connect/logger.html>

Try running the example server script “02\_logger.js” in the 02\_bootstrap example project folder:

> node 02\_logger.js

The server contains a web page named “logger.html”. What fields are displayed in the console when you run the html from your browser at <http://localhost:3000/02_logger.html>

The logger also has some standard predefined formats including: ‘short’, ‘tiny’, and ‘dev’. Try each one using the format shown below. Remember to stop and start the server after making changes.

app.use(express.logger(’short’))

What happens?

## Twitter Bootstrap Menu

In the course of this book we’ll be developing a sample application for a website called “Track My Things”. The Track My Things website allows users to subscribe to one or more “Collections” where they can track various items that they own. For example, they could have a “Books” collection which would allow a user to track the books that they own, and maybe who the book was loaned to. They could also have a “Video” collection that would allow them to maintain a list of Videos that they own and perhaps a link to where the video is on their hard drive. There could even be a model train collection to allow them to track the model trains that they own and where they are stored.

**\*\*\* TODO: either discuss data model and data modeling for MongoDB here, or wait until later and put a reference to that later discussion here????**

We’ll be describing the application in more detail as we go, but for now, we want to illustrate how we would implement the website’s main menu. The first menu allows a user to choose which collection they want to access. Choosing a collection then changes the name of the menu to the name of the selected collection as shown in Figure 2.1 where the user has chosen to view the “Trains” collection.

Every collection also has a number of different views. A view defines which fields to show from a collection, which rows to show from the collection, and the sort order of the results. Whenever a user selects a collection, the list of Views will also change. The menu then allows the user to choose which view to use for displaying the collection. The views menu name also changes to show the name of the currently selected view as illustrated in Figure 2.1, where the user will be viewing the “Complete Database Listing” view for the “Trains” collection.

One final menu shows the name of the person currently signed on and allows them to choose to view collections, view their profile, or sign off. Initially we’ll be building a version of the application where the menus don’t yet perform an action. Don’t worry about that for now, we’ll be adding the actions later after we’ve added the AngularJS code.

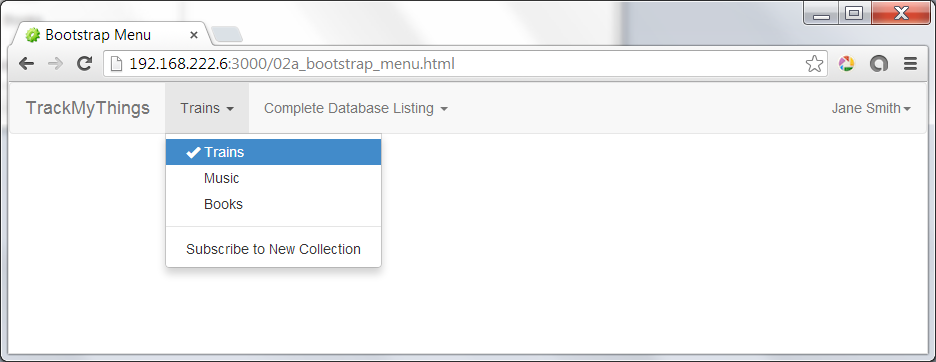
c

Figure 2.1 – Track My Things Main Menu

We’ve already seen how Bootstrap can enable you to easily develop web pages which run on a variety of devices. Now let’s take a look at some of the capabilities that make it easy to develop user interfaces in Bootstrap. We’ll start by creating a menu or “navbar” as bootstrap refers to it. More details on creating Bootstrap navbars can be found at <http://getbootstrap.com/components/#navbar>.

For our first example, if the web server from the previous example is not still running, start it using the console command:

> node 02\_logger.js

After the server is started, run the example html at <http://192.168.222.6:3000/02a_bootstrap_menu.html>. You should see the menu shown in Figure 2.1. Let’s look at “02a\_bootstrap\_menu.html” in a bit more detail. Lines 8 -10 define a style is referenced later in the html file. We’ll eventually move this into a separate application CSS file, but for now we are putting the style inline.

**\*\*\*\* TODO: add style for padding of body AND fixing navbar \*\*\*\*\*\*\*\***

**\*\*\*\*\* this will shift down the line numbers for following discussions \*\*\*\*\***

<style>

.font-white {color : white; }

</style>

Line 15 contains the tag:

<nav class="navbar navbar-default" role="navigation">

The use of a special class is something that you’ll see throughout Bootstrap. This is the primary way that bootstrap controls the rendering of the items on the page. You may not be familiar with the <nav …> tag. This is something new in HTML5 and is supported by most browsers with the exception of Internet Explorer versions before version 9. The <nav…> tag can help with accessibility, as does the role=”navigation” attribute.

Lines 17 – 19 then have the name of the website and (eventually) a link to the home page.

<div class="navbar-header">

<a class="navbar-brand" href="#">TrackMyThings</a>

</div>

Lines 23 – 34 contain the first drop down list, the list of collections shown in the drop down menu in Figure 2.1.

<!-- List of Collections -->

<ul class="nav navbar-nav">

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">Trains <b class="caret"></b></a>

<ul class="dropdown-menu">

<li><a href="#"><span class="glyphicon glyphicon-ok"></span> Trains</a></li>

<li><a href="#"><span class="glyphicon glyphicon-ok font-white" ></span> Music</a></li>

<li><a href="#"><span class="glyphicon glyphicon-ok font-white" ></span> Books</a></li>

<li class="divider"></li>

<li><a href="#">Subscribe to New Collection</a></li>

</ul>

</li>

</ul>

Bootstrap uses the <ul…>, HTML unordered list tag, to define a drop down menu. Each item in the menu is a list item enclosed within the HTML <li…> tag. A line item with a class of “divider” generates a line dividing the menu items.

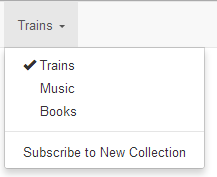


Figure 2.2 – Dropdown List of Collections

Line 27 demonstrates the use of glyphicon via the class="glyphicon glyphicon-ok">. Glyphicons define a font which allows displaying symbols in an efficient manner.

<li><a href="#"><span class="glyphicon glyphicon-ok"></span> Trains</a></li>

<li><a href="#"><span class="glyphicon glyphicon-ok font-white" ></span> Music</a></li>

In our example we use a “glyphicon-ok” to generate a check mark. Menu items which are not selected then use the previously mentioned “font-white” CSS tag to make the check mark white, effectively making them invisible while maintaining proper spacing before unselected menu items. For more information on glyphicons see <http://getbootstrap.com/components/#glyphicons>.

On line 59, the menu showing the signed on user, you’ll see a class of “navbar-right”. This tells bootstrap to right align the last menu so that it appears to the far right of the menu bar.

<ul class="nav navbar-nav navbar-right">

You’ll see a problem with this current menu if you resize the window to make it very narrow or use the web page on a smaller device such as a cell phone. The menu items begin to stack up, as shown in Figure 2.3. On a small device such as a smart phone this could use up most of our screen real estate –something we’d like to avoid. Let’s see how to make our web page appear better on smaller devices, making it more responsive, by collapsing the menu when the width of the web page reaches a certain minimum.

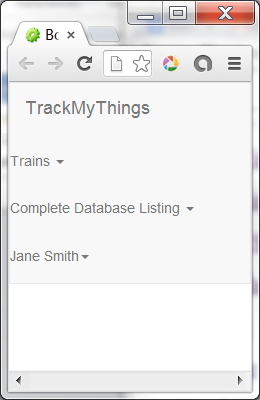


Figure 2.3 – Stacked Menu List

## Bootstrap Collapsed Menu

If you run the example web page “02b\_menu\_collapse.html” you’ll see that when the window becomes narrower, the menus collapse and are represented by a single icon with three bars.

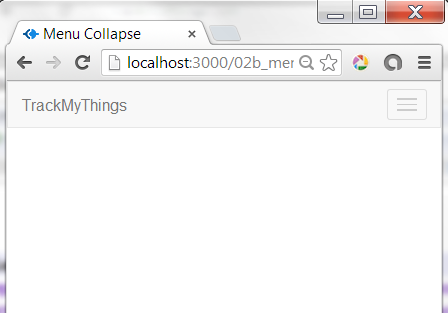


Figure 2.4 – Collapsed Menu

If you then click on the collapsed menu icon you’ll see the stacked list of menus. If you look at the source for the menu collapse html, “02b\_menu\_collapse.html”, you’ll see that we’ve added the following lines starting at line 19:

<button type="button" class="navbar-toggle" data-toggle="collapse"

data-target=".**collapsible-menu-list**">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

The class defines the button as a toggling navbar that should collapse when toggled. The data-target= attribute defines class of the menu list to collapse. We’ve added the “collapsible-menu-list” class to the group of menus that we want collapsed on line 29.

<div class="collapse navbar-collapse **collapsible-menu-list**">

A note on accessibility - on line 20 we have the line:

<span class="sr-only">Toggle navigation</span>

This class hides the text from views, but allows a screen reader used for accessibility, to read a label for the button.

Now that we have a menu bar, let’s add the bootstrap table.

## Bootstrap Table

A table in Bootstrap is based on the standard HTML table with a number of special classes to control the styling. You can see what the example table by starting the Node.js server and using your browser to open the web page <http://localhost:3000/02c_bootstrap_table.html>.

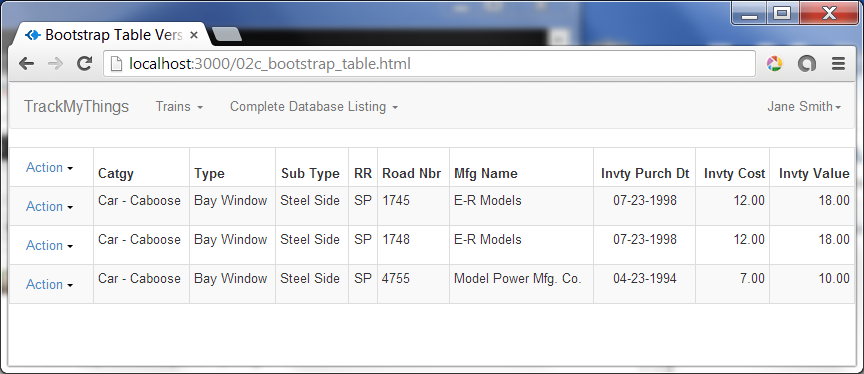


Figure 2.5 – Example Bootstrap Table

~~If you look at the example HTML in client/02c\_bootstrap\_table.html you’ll find the definition for the table starting at line 84. The first line you’ll see is:~~

~~<div class="table-responsive">~~

**\*\*\*NOTE\*\*\* remove table-responsive. This interferes with drop down menus within the table if the table is scrolling horizontally. This will affect line numbers of code.**

~~The class “table-responsive” wrapping the HTML table will cause a scroll bar to appear on smaller devices or if the window width falls below a certain threshold.~~

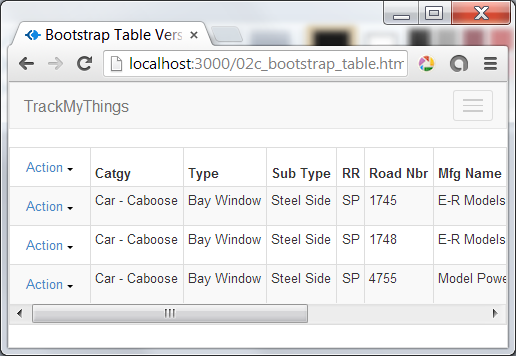


Figure 2.6 – Effects of “table-responsive” Tag

Next you’ll see an HTML “<table …” tag with a number of special classes.

<table class="table table-striped table-bordered table-hover table-condensed">

**“table-striped”** will cause the odd numbered rows to be shaded. **“table-bordered”** places lines around the table as well as the vertical lines between columns in the table. **“table-hover”** causes a row to be highlighted when the cursor is moved over the row. **“table-condensed”** makes a table with less space between cells. To see better what effect each class has on the table appearance, try removing one or more of them and see how it changes the display.

Starting on line 88 you’ll see a table cell which contains a drop down list similar to those in the main menu navbar. Our table uses these in both the headers as well as the data rows.

<th>

<div class="btn-group">

<a class="btn dropdown-toggle" data-toggle="dropdown" href="#">

Action

<span class="caret"></span>

</a>

<ul class="dropdown-menu">

<li><a href="#">New Item</a></li>

</ul>

</div>

</th>

On lines 105 and 106 you’ll see the use of classes “text-center” and “text-right”. These classes can be used in paragraphs and other text blocks as well as in HTML “th” and “td” tags to change the alignment of text. The default for table cells, both headers and data, is align left.

<th class="text-center">Invty Purch Dt</th>

<th class="text-right">Invty Cost</th>

The rest of the HTML should look familiar if you’ve worked with HTML tables before. You can see a few more options available for bootstrap tables in the documentation at <http://getbootstrap.com/css/#tables>.

Unless you are showing a small table, you’ll usually have more rows than you want to display on a single page. Next we’ll look at how bootstrap supports pagination, allowing a user to browse through a large number of pages without having to download the entire table to the browser.

## Bootstrap Table Pagination

Although we really don’t need pagination yet, we’ll add a standard pagination tool bar for use later on. Bootstrap uses the <UL> and <LI> tags, the same tags used for drop down lists, to implement a horizontal list of page numbers as shown in Figure 2.7. As before, classes are used to control the layout. Try navigating to the web page <http://localhost:3000/02d_table_pagination.html> to see the example.

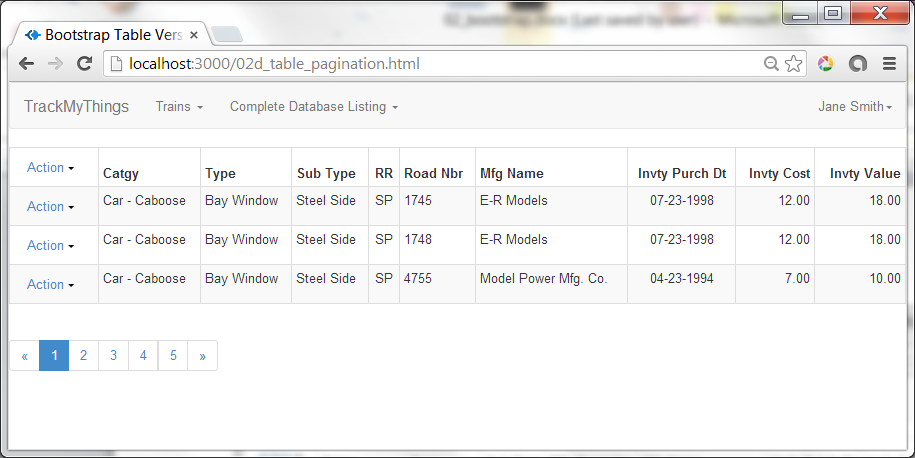


Figure 2.7 – Boot Page with Pagination

If you look at the example web page, “02d\_table\_pagination.html”, you’ll see the pagination HTML starting at line 207. This consists of the following lines of code:

<ul class="pagination">

<li><a href="#" title="Previous">&laquo;</a></li>

<li class="active"><a href="#">1</a></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

<li><a href="#">4</a></li>

<li><a href="#">5</a></li>

<li><a href="#" title="Next">&raquo;</a></li>

</ul>

You can also specify smaller or larger “blocks” in the pagination area by adding a second class after “pagination”, including “pagination-sm” and “pagination-lg”. For examples of this see the bootstrap documentation at <http://getbootstrap.com/components/#pagination>. We can also specify the current page by using the class=”active” attribute on one of the <li …> tags, as we’ve done for page 1. If a page is not available, say we’ve reached the end of our list, you can use class=”inactive”.

## Bootstrap Themes

One of the advantages of using a popular package is a community that continues to help the product stay up to date and also builds upon that package. For Bootstrap there is the Content Distribution Network (CDN) as previously discussed. There are also a number of free and paid “themes” one can use to provide different appearances for your application. One set that is free and easy to experiment are those provided by Bootswatch at <http://www.bootstrapcdn.com/#bootswatch_tab>. Using any one of the themes shown on this requires only that we modify the file reference we are using for bootstrap’s CSS files. For example, currently we’ve been using the following tag in our HTML:

PUT CURRENT TAG HERE

To change our appearance to the screen shown below, we just need to change that single line to be

PUT NEW TAG HERE

\*\*\* paste screen capture here \*\*\*\*\*

Using the tag below produces yet a different look.

PUT NEW TAG HERE

\*\*\*\*\* past screen capture here \*\*\*\*

We now have a bootstrap page with a menu, table, and paging controls. We can even change the theme easily, perhaps even allowing a user to choose what theme they want to use. Now we’re ready to start adding AngularJS to make it easier to show larger tables.