Exercise (Instructions): Angular Filters

Objectives and Outcomes

This exercise is designed to illustrate to you how Angular filters can be used effectively in an application. In this exercise, you will create a menu that either presents the whole menu, or parts of the menu as selected by the user. At the completion of this exercise, you will be able to:

- Use Angular filters to filter items based on user's selection
- Use Bootstrap navigation tabs in conjunction with AngularJS
- Make use of the ng-class and ng-click Angular directives

Deleting Comments from Menu Items

- Starting with the *menu.html* file from the previous exercise, you will first **delete** the following lines corresponding to the comments in the menu items from the page:

```
                        <p>Comment: {{dish.comment}}</p>
                        <p>Type your comment:
                         <input type="text" ng-model="dish.comment"></p>
```

Setting up Tabbed Navigation for the Menu

- We will now take the help of Bootstrap tabs to set up a tabbed navigation for the menu. We will construct four tabs to display all the menu items, appetizers, mains, and Desserts based on user's selection. Add the following code to the column div, just before the <ul> containing the menu items:

```
                <ul class="nav nav-tabs" role="tablist">
                    <li role="presentation">
                    <a
                     aria-controls="all menu"
                     role="tab">The Menu</a></li>
                    <li role="presentation">
```

```
                <li role="presentation">
                <a
                 aria-controls="appetizers"
                 role="tab">Appetizers</a></li>
                <li role="presentation">
                <a
                 aria-controls="mains"
                 role="tab">Mains</a></li>
                <li role="presentation">
                <a
                 aria-controls="desserts"
                 role="tab">Desserts</a></li>
            </ul>
```

You will see the four tabs set up by the above navigation. However clicking on the tabs does not do anything, since we have not set up the logic to activate the tabs.

- Next, enclose the menu *<ul>* inside a * * with class *tab-content*. Also appy the *tab-pane fade in active* class to the ul as shown below:

```
        <div class="tab-content">
           <ul class="media-list tab-pane fade in active">
               . . .
           </ul>
        </div>
```

Keep the *<li>* item in the *<ul>* as is.

Activating the Navigation Tabs

- We will now activate the tabs so that the user can select any of the four tabs. We need to do this by adding some functions to the menucontroller and then use the functions to activate the tabs when selected. First we add a variable named tab to the menucontroller code as follows:

```
        this.tab = 1;
```

Add this as the first statement in the menu controller. Note that the tabs are given indices 1 .. 4. The variable tab will keep track of the currently active tab. By default this will be assigned as the first tab. Obviously, the remaining tabs will be numbered 2 .. 4.

- Whenever the user clicks on a tab, we need to activate that tab. To do this, we take the help of the *ng-click* Angular directive. This directive will be fired when the user clicks on the tab. To do this, we will add an ng-click directive to every <a> tag within the tabs as follows:

```
            <ul class="nav nav-tabs" role="tablist">
                <li role="presentation">
```

```
            <li role="presentation">
            <a ng-click="menuCtrl.select(1)"
             aria-controls="all menu"
             role="tab">The Menu</a></li>
            <li role="presentation">
            <a ng-click="menuCtrl.select(2)"
             aria-controls="appetizers"
             role="tab">Appetizers</a></li>
            <li role="presentation">
            <a ng-click="menuCtrl.select(3)"
             aria-controls="mains"
             role="tab">Mains</a></li>
            <li role="presentation">
            <a  ng-click="menuCtrl.select(4)"
             aria-controls="desserts"
             role="tab">Desserts</a></li>
        </ul>
```

Note that the complete code is given above, just in case you wish to cut and paste the code. Note how the *ng-click* directives are introduced using *ng-click="menuCtrl.select(i)".*

- Upon clicking of the tab, the ng-click directive will cause the execution of the select() function in the menucontroller. Next, we need to implement the select() function. Add the following code to the menucontroller to implement the select() function:

```
    this.select = function(setTab) {
        this.tab = setTab;

    }
```

This function will set the tab variable to the selected tab index.

- Next we take the help of the *ng-class* directive in order to add the Bootstrap *active* class to the selected tab. To do this modify the *<ul>* for the tabs as follows, where you can clearly see the ng-class directive being used:

```
        <ul class="nav nav-tabs" role="tablist">
            <li role="presentation"
             ng-class="{active:menuCtrl.isSelected(1)}">
            <a ng-click="menuCtrl.select(1)"
             aria-controls="all menu"
             role="tab">The Menu</a></li>
```

```
                role="tab">The Menu</a></li>
               <li role="presentation"
                ng-class="{active:menuCtrl.isSelected(2)}">
               <a ng-click="menuCtrl.select(2)"
                aria-controls="appetizers"
                role="tab">Appetizers</a></li>
               <li role="presentation"
                ng-class="{active:menuCtrl.isSelected(3)}">
               <a ng-click="menuCtrl.select(3)"
                aria-controls="mains"
                role="tab">Mains</a></li>
               <li role="presentation"
                ng-class="{active:menuCtrl.isSelected(4)}">
               <a  ng-click="menuCtrl.select(4)"
                aria-controls="desserts"
                role="tab">Desserts</a></li>
           </ul>
```

Note that for each of the *<li>* elements, we introduced the *ng-class="{active:menuCtrl.isSelected(i)}"* directive. This directive will apply the active class to that element, if the function on the right (that specifies a condition) evaluates to true.

- Next, we need to implement the isSelected() function in the menucontroller. Add the following code to implement this function in the menucontroller:

```
        this.isSelected = function (checkTab) {
            return (this.tab === checkTab);
        }
```

This function will return true if the current tab is the same as the tab specified in the function parameter.

- Now the selection of the tabs, and activating the selected tab should work correctly. However no matter which tab is selected, you will still see the list of all menu items.

Using Angular Filter

- Next we take the help of the Angular filter to select only those items from the menu corresponding to each tab. Note that each dish object already has a property named *category*. We can use this to filter our items.

- We will now set up the filters on the list items in the menu as follows:

```
            <li class="media" ng-repeat="dish in menuCtrl.dishes | filte
r:menuCtrl.filtText">
```

The above modification implies that the filter will use the variable filtText from the menucontroller to filter the items from the dishes array.

- Next we need to introduce the filtText variable in the menucontroller and have a way of setting it to the appropriate value when the various tabs are selected. To do this, add the following code to the menucontroller:

```
            this.filtText = '';
```

We are specifying that initially since the first tab is selected as default, the filtText should not filter out any item from the menu. Hence filtText is set to the empty string.

- Then, whenever a tab is selected, the filtText value should be updated to reflect the selected tab and the corresponding filter value to be applied. To do this, modify the select() function as follows:

```
            this.select = function(setTab) {
                this.tab = setTab;

                if (setTab === 2)
                    this.filtText = "appetizer";
                else if (setTab === 3)
                    this.filtText = "mains";
                else if (setTab === 4)
                    this.filtText = "dessert";
                else
                    this.filtText = "";
            }
```

Note that using the if statements, we are able to set the filtText to the appropriate value.

- Now the application should show the correctly filtered results from the menu items when the tabs are selected.

Conclusions

In this exercise we explored the use of Angular filter to select specific items from a JavaScript array object and use them to construct the list of items in the menu. We also explored the use of the *ng-click* and the *ng-class* directives.