Exercise (Instructions): Exploring Ionic Part 1

Exercise Resources

```
services.js
controllers.js
```

Objectives and Outcomes

In this exercise, we will continue to port the AngularJS app that we developed in the previous AngularJS course to an Ionic app. Along the way we will learn more about Ionic framework. At the end of this exercise, you will be able to:

- Set up the controllers, services and communicating with the server to enable the construction of an Ionic app
- Design templates for the Ionic app using Ionic's Angular directives

Note: Make sure that your json-server is up and running

Updating the conFusion App

- Download the *services.js* file provided above to your *www/js* folder of your Ionic project.

Updating services.js

- Open *services.js* and update the angular.module as follows:

```
angular.module('conFusion.services', ['ngResource'])
```

- Save the changes.

Updating index.html

- Open index.html file to import ngResource by adding the following after the statement that imports the Ionic bundle:

```
    <script src="lib/ionic/js/angular/angular-resource.min.js"></script>
```

- Then, import services.js after you import the controllers.js file as follows:

```
    <script src="js/services.js"></script>
```

- Save the changes.

Updating app.js

- Open app.js and then inject the conFusion.services into the Angular module as follows:

```
angular.module('conFusion', ['ionic', 'conFusion.controllers','conFusion.services'])
```

- Save the changes.

Updating controllers.js

- Before you proceed forward, download the *controllers.js* file that we provide above. This is the controller code from the AngularJS application that you developed in the previous course. From this file, copy only the code corresponding to the controllers. You will paste this code into the *controllers.js* file for our Ionic app.
- Open the controllers.js file of our Ionic app to start updating the controllers. In this file, replace the PlayListsCtrl and PlayListCtrl controller code with the code you copied in the step above. Now we have moved all the controller code from the previous course into our Ionic app.

Updating home.html

- Start Ionic serve by typing the following at the prompt in your command window/terminal while in the application folder:

```
    ionic serve --lab
```

- Open *home.html*, and update it as follows and save:

```
<ion-view view-title="Ristorante con Fusion">
  <ion-content>
    <div class="card">
        <div class="item item-divider">
            Our Lipsmacking Culinary Creations
            <span class="badge badge-assertive">{{dish.label}}</span>
        </div>
        <div class="item item-thumbnail-left item-text-wrap">
            <img ng-src="{{baseURL+dish.image}}" alt="Uthappizza">
```

```
            <h2>{{dish.name}}
             <span class="badge">{{dish.price | currency}}</span></h2>
            <p>{{dish.description}}</p>
        </div>
    </div>
  </ion-content>
</ion-view>
```

- When we see the updates in the browser, it shows a card structure, but the content is blank. You need to fix the controller and app configuration.
- Open *app.js* and update the app.home state to introduce the IndexController to the state as follows and save the changes:

```
  .state('app.home', {
    url: '/home',
    views: {
      'mainContent': {
        templateUrl: 'templates/home.html',
          controller: 'IndexController'
      }
    }
  })
```

- Then, open *controllers.js* file and update the IndexController as follows and save the changes:

```
        .controller('IndexController', ['$scope', 'menuFactory', 'corporateFactory', 'bas
eURL', function($scope, menuFactory, corporateFactory, baseURL) {

                        $scope.baseURL = baseURL;
                        $scope.leader = corporateFactory.get({id:3});
                        $scope.showDish = false;
                        $scope.message="Loading ...";
                        $scope.dish = menuFactory.getDishes().get({id:0})
                        .$promise.then(
                            function(response){
                                $scope.dish = response;
                                $scope.showDish = true;
                            },
```

```
                            function(response) {
                                $scope.message = "Error: "+response.status + " " + respon
se.statusText;
                            }
                        );
                        $scope.promotion = menuFactory.getPromotion().get({id:0});
      }])
```

- The app should now correctly display the information about the featured dish.
- Next, update the *home.html* page by adding two more cards for the promotions and the executive chef. You can add the following code to the *home.html* page and save the changes:

```
      <div class="card">
          <div class="item item-divider">
            This Month's Promotions
            <span class="badge badge-assertive">{{promotion.label}}</span>
          </div>
          <div class="item item-thumbnail-left item-text-wrap">
               <img ng-src="{{baseURL+promotion.image}}" alt="{{promotion.name}}">
                <h2>{{promotion.name}}
                 <span class="badge">{{promotion.price | currency}}</span></h2>
                <p>{{promotion.description}}</p>
          </div>
      </div>
      <div class="card">
          <div class="item item-divider">
            Meet our Culinary Specialists
          </div>
          <div class="item item-thumbnail-left item-text-wrap">
               <img ng-src="{{baseURL+leader.image}}" alt="{{leader.name}}">
                <h2>{{leader.name}}</h2>
                <h4>{{leader.designation}}</h4>
                <p>{{leader.description}}</p>
          </div>
      </div>
```

- See the result in the browser to see how the home page now displays the three cards.

Conclusions

In this exercise, you explored some more features of the Ionic platform. You were able to port much of the controller and services code from the Angular application with minor changes. Then you designed the home page with three cards using Ionic's CSS classes and Angular directives.