

Exercise (Instructions): Angular Modules and Controllers

Objectives and Outcomes

In this exercise you will learn to use Angular modules and controllers to organize your Angular application and separate the data from the layout of the page. At the end of this exercise, you will be able to:

- Define an Angular module, and initialize the Angular app with the module
- Define a controller within your Angular app and use the controller within your application

Creating Angular Module

- First, update the ng-app with a name for the module that will define the application as follows:

```
<html lang="en" ng-app="confusionApp">
```

This means that we now need to define an Angular module named *confusionApp*. We will do this next.

- We will include the Angular code within the menu.html page for this exercise. It's better to separate out the code into separate code files. We will adopt this in the exercises later.
- First, go to the bottom of the page and add the following code just before the body closing tag, to create the Angular module:

```
<script>

    var app = angular.module('confusionApp', []);

</script>
```

- Next we will add a controller to this app as follows:

```
<script>
    var app = angular.module('confusionApp', []);
    app.controller('menuController', function() {
```

```
});  
  
</script>
```

- Then, we will shift the dishes JavaScript object array from the ng-init to the controller. Cut out the ng-init from the div completely, and add the following code to the Controller:

```
var dishes=[  
    {  
        name:'Uthapizza',  
        image: 'images/uthapizza.png',  
        category: 'mains',  
        label: 'Uthap'
```

```

        label: 'Hot',
        price: '4.99',
        description: 'A unique combination of Indian Uthap
pam (pancake) and Italian pizza, topped with Cerignola olives, ripe vine che
rry tomatoes, Vidalia onion, Guntur chillies and Buffalo Paneer.',
        comment: ''
    },
    {
        name: 'Zucchipakoda',
        image: 'images/zucchipakoda.png',
        category: 'appetizer',
        label: '',
        price: '1.99',
        description: 'Deep fried Zucchini coated with mild
ly spiced Chickpea flour batter accompanied with a sweet-tangy tamarind sauc
e',
        comment: ''
    },
    {
        name: 'Vadonut',
        image: 'images/vadonut.png',
        category: 'appetizer',
        label: 'New',
        price: '1.99',
        description: 'A quintessential ConFusion experienc
e, is it a vada or is it a donut?',
        comment: ''
    },
    {
        name: 'ElaiCheese Cake',
        image: 'images/elaicheesecake.png',
        category: 'dessert',
        label: '',
        price: '2.99',
        description: 'A delectable, semi-sweet New York St
yle Cheese Cake, with Graham cracker crust and spiced with Indian cardamoms'
    },
    {
        name: 'Mango Lassi',
        image: 'images/mangolassi.png',
        category: 'beverage',
        label: 'New',
        price: '1.99',
        description: 'A refreshing blend of ripe mangoes and
yogurt, garnished with fresh mint leaves and a dusting of
cardamom powder.',
        comment: ''
    }
];

this.dishes = dishes;

```

Attaching the Controller

- Modify the row class as follows to add the controller to the div:

```

<div class="row row-content" ng-controller="menuController as menuCt
rl">

```

This adds the menuController as the controller for this div, and also assigns an alias to the controller as menuCtrl.

- Next update, the list tag as follows:

```
<li class="media" ng-repeat="dish in menuCtrl.dishes">
```

- The web page itself will show no change, except that the code is now factored out into a controller.

Conclusions

In this exercise, you learnt about Angular module and controller and how we assign a controller to a div and make the data within the controller accessible within the div.

