Exercise (Instructions): Angular ui-router and Resolve

Objectives and Outcomes

In this exercise you will use resolve in ui-router to simplify the code within your controllers. At the end of this exercise, you will be able to:

- Use the resolve object to resolve the dependencies in the state
- Inject the data after resolution into the controller

Updating services.js

- Open *services.js* and update the menuFactory and also introduce a promotionFactory as follows:

```
1   .factory('menuFactory', ['$resource', 'baseURL', function ($resource, baseURL) {
2
3               return $resource(baseURL + "dishes/:id", null, {
4                   'update': {
5                       method: 'PUT'
6                   }
7               });
8
9   }])
10
11  .factory('promotionFactory', ['$resource', 'baseURL', function ($resource,
      baseURL) {
12              return $resource(baseURL + "promotions/:id");
13
14  }])
```

Updating controllers.js

- In *controllers.js* replace all occurrences of menuFactory.getDishes() with just menuFactory.
- In addition, update the IndexController as follows:

```
 1    .controller('IndexController', ['$scope', 'menuFactory', 'promotionFactory',
        'corporateFactory', 'baseURL', function ($scope, menuFactory, promotionFactory
        , corporateFactory, baseURL) {
 2
 3        $scope.baseURL = baseURL;
 4        $scope.leader = corporateFactory.get({
 5            id: 3
 6        });
 7
 8        $scope.showDish = false;
 9        $scope.message = "Loading ...";
10
11        $scope.dish = menuFactory.get({
12                id: 0
13            })
14            .$promise.then(
15                function (response) {
16                    $scope.dish = response;
17                    $scope.showDish = true;
18                },
19                function (response) {
20                    $scope.message = "Error: " + response.status + " " + response
                        .statusText;
21                }
22            );
23
24        $scope.promotion = promotionFactory.get({
25            id: 0
26        });
27
28    }])
```

Updating app.js to use resolve

- Open *app.js* and update the app.favorites and app.dishdetail states as follows:

```
 1      .state('app.favorites', {
 2        url: '/favorites',
 3        views: {
 4          'mainContent': {
 5            templateUrl: 'templates/favorites.html',
 6              controller:'FavoritesController',
 7            resolve: {
 8                dishes:  ['menuFactory', function(menuFactory){
 9                  return menuFactory.query();
10                }],
11                          favorites: ['favoriteFactory', function
                            (favoriteFactory) {
12                  return favoriteFactory.getFavorites();
13              }]
14          }
15        }
16      }
17    })
18
19    .state('app.dishdetails', {
20      url: '/menu/:id',
21      views: {
22        'mainContent': {
23          templateUrl: 'templates/dishdetail.html',
24          controller: 'DishDetailController',
25          resolve: {
26              dish: ['$stateParams','menuFactory', function($stateParams,
                    menuFactory){
27                  return menuFactory.get({id:parseInt($stateParams.id, 10)});
28              }]
29          }
30        }
31      }
```

```
32      });
```

Update controllers.js

- Open *controllers.js* and update FavoritesController as follows:

```
1    .controller('FavoritesController', ['$scope', 'dishes', 'favorites',
        'favoriteFactory', 'baseURL', '$ionicListDelegate', '$ionicPopup',
        '$ionicLoading', '$timeout', function ($scope, dishes, favorites,
        favoriteFactory, baseURL, $ionicListDelegate, $ionicPopup, $ionicLoading,
        $timeout) {
2
3        $scope.baseURL = baseURL;
4        $scope.shouldShowDelete = false;
5
6        $scope.favorites = favorites;
7
8        $scope.dishes = dishes;
9
10       . . .
11
12   }])
13
14
```

- Similarly, update DishDetailController as follows:

```
1    .controller('DishDetailController', ['$scope', '$stateParams', 'dish',
        'menuFactory', 'favoriteFactory', 'baseURL', '$ionicPopover', '$ionicModal',
        function ($scope, $stateParams, dish, menuFactory, favoriteFactory, baseURL,
        $ionicPopover, $ionicModal) {
2
3        $scope.baseURL = baseURL;
4
5        $scope.dish = dish;
6
7        . . .
8
9    }])
```

- Save all changes and examine the resulting app.

Conclusions

In this exercise, you updated the app to use the resolve in ui-router to simplify the controllers.

✓ Complete

👍   👎   🏳