

Exercise (Instructions): Angular ui-router and Resolve I

Coursera

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:

```
.factory('menuFactory', ['$resource', 'baseUrl', function ($resource,
baseUrl) {

    return $resource(baseUrl + "dishes/:id", null, {
        'update': {
            method: 'PUT'
        }
    });

}])

.factory('promotionFactory', ['$resource', 'baseUrl', function ($resource,
baseUrl) {

    return $resource(baseUrl + "promotions/:id");

}])
```

Updating controllers.js

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

```
.controller('IndexController', ['$scope', 'menuFactory', 'promotionFactory',
'corporateFactory', 'baseUrl', function ($scope, menuFactory,
promotionFactory, corporateFactory, baseUrl) {
```

```

$scope.baseUrl = baseUrl;
$scope.leader = corporateFactory.get({
    id: 3
});

$scope.showDish = false;
$scope.message = "Loading ...";

$scope.dish = menuFactory.get({
    id: 0
})
.$promise.then(
    function (response) {
        $scope.dish = response;
        $scope.showDish = true;
    },
    function (response) {
        $scope.message = "Error: " + response.status + " " +
response.statusText;
    }
);

$scope.promotion = promotionFactory.get({
    id: 0
});

}])

```

Updating app.js to use resolve

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

```

.state('app.favorites', {
    url: '/favorites',
    views: {
        'mainContent': {
            templateUrl: 'templates/favorites.html',
            controller: 'FavoritesController',
            resolve: {
                dishes: ['menuFactory', function(menuFactory){

```

```

        return menuFactory.query();
    }],

    favorites: ['favoriteFactory',
function(favoriteFactory) {
    return favoriteFactory.getFavorites();
}]]
    }
}
}
}))

.state('app.dishdetails', {
  url: '/menu/:id',
  views: {
    'mainContent': {
      templateUrl: 'templates/dishdetail.html',
      controller: 'DishDetailController',
      resolve: {
        dish: ['$stateParams', 'menuFactory', function($stateParams,
menuFactory){
          return menuFactory.get({id:parseInt($stateParams.id, 10)});
        }]
      }
    }
  }
});

```

Update controllers.js

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

```

.controller('FavoritesController', ['$scope', 'dishes', 'favorites',
'favoriteFactory', 'baseURL', '$ionicListDelegate', '$ionicPopup',
'$ionicLoading', '$timeout', function ($scope, dishes, favorites,
favoriteFactory, baseURL, $ionicListDelegate, $ionicPopup, $ionicLoading,
$timeout) {

    $scope.baseURL = baseURL;
    $scope.shouldShowDelete = false;

    $scope.favorites = favorites;

```

```
    $scope.dishes = dishes;

    . . .

  })
```

- Similarly, update DishDetailController as follows:

```
.controller('DishDetailController', ['$scope', '$stateParams', 'dish',
'menuFactory', 'favoriteFactory', 'baseUrl', '$ionicPopover', '$ionicModal',
function ($scope, $stateParams, dish, menuFactory, favoriteFactory, baseUrl,
$ionicPopover, $ionicModal) {

    $scope.baseUrl = baseUrl;

    $scope.dish = dish;

    . . .

  })
```

- 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.