**ANGULARJS HANDS-ON PRACTICE GUIDELINES**

CSC

3170 Fairview Park Drive  
Falls Church, VA, 22042

|  |  |  |
| --- | --- | --- |
| **PREPARED FOR:**  .NET FRESHER | **DATE:**  06//2015 | **SUBMITTED BY:**  DUY NGUYEN -CSC |
|  | | |
|  |  |  |
|  |  | |

Table of Contents

Table of Contents 2

1 Set up structure 3

1.1 Create separate folders for controllers, directives, views, filters 3

1.2 Prepare layout for main view in index.html 3

1.3 Create fake data for books model in app.js 4

1.4 Run the app 5

2 Display Product Details 5

2.1 Render books list 5

2.2 Run the app 6

3 Display Categories List 6

3.1 Target 6

3.2 Creat custom filter 6

3.3 Render categories list 7

3.4 run the app 7

4 Handle Select Category function 8

4.1 Target 8

4.2 Define booklist Controller 8

4.3 Filter Product by Category 8

4.4 Run the app 9

5 Highlight selected category 9

5.1 target 9

5.2 Define highlight class for selected category 9

5.3 Run the app 10

6 Using service 10

6.1 Target 10

6.2 Create JSON data model 11

6.3 Retrieve data using $http 11

6.4 Handling ajax error 11

6.5 Run the app (in case of http error) 12

7 Create shopping cart 12

7.1 Create Cart Module to handle business logic for cart. 12

7.2 Create cartSummary directive 12

7.3 Create cart summary template 13

7.4 Add Cart Summary directive to header 13

7.5 Run the app 14

8 Adding Add-to-cart function 14

8.1 Add an Add-to-cart button in each Book item in index.html 14

8.2 Changes in bookListController.js 14

8.3 Run application 15

9 Working with Routing 15

9.1 Target 15

9.2 Separate BookList View 16

9.3 Checkout View 16

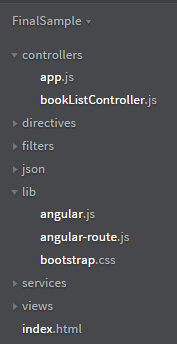
9.4 Routing configuration 16

9.5 Run the app 17

# Set up structure

Create a folder for the web app. (whatever name, eg. **FinalSample**)

## Create separate folders for controllers, directives, views, filters



## Prepare layout for main view in index.html

<!DOCTYPE html>

<html ng-app="myapp">

<head>

<title>Books Store</title>

<script src="lib/angular.js"></script>

<link href="lib/bootstrap.css" rel="stylesheet" />

<script>

var app = angular.module("myapp", []);

</script>

<script src="controllers/app.js"></script>

<script src="lib/angular-route.js"></script>

</head>

<body ng-controller="mainCtrl">

<div class="navbar navbar-inverse">

<a class="navbar-brand" href="#">Books Store</a>

</div>

<div class="panel row">

<div class="col-xs-3 well">

Categories display here.

</div>

<div class="col-xs-8">

<div class="well">

Books display here.

</div>

</div>

</div>

</body>

</html>

## Create fake data for books model

* Create app.js in controllers folder. This is the main controller for the app.

app.controller('mainCtrl', function($scope){

**$scope.data** = {

**books**: [

{"category": "Category #1", "name": "Book #1", "description": "Some description", "price": "100"},

{"category": "Category #1", "name": "Book #2", "description": "Some description", "price": "200"},

{"category": "Category #2", "name": "Book #3", "description": "Some description", "price": "300"},

{"category": "Category #2", "name": "Book #4", "description": "Some description", "price": "400"},

{"category": "Category #2", "name": "Book #5", "description": "Some description", "price": "500"},

{"category": "Category #3", "name": "Book #6", "description": "Some description", "price": "600"},

{"category": "Category #3", "name": "Book #7", "description": "Some description", "price": "700"},

{"category": "Category #3", "name": "Book #8", "description": "Some description", "price": "800"},

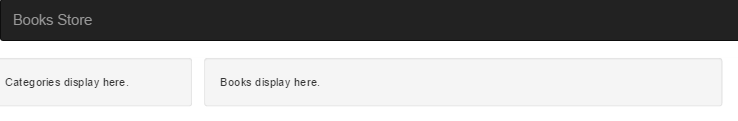
{"category": "Category #3", "name": "Book #9", "description": "Some description", "price": "900"}

]

};

});

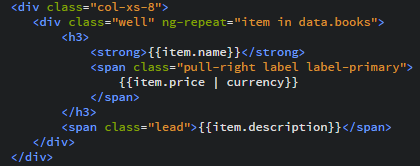
## Run the app



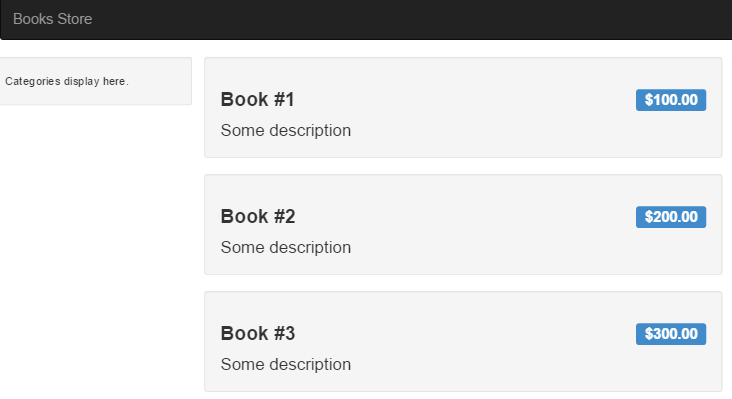
# Display Product Details

## Render books list

* Modify books content section in **index.html** using **ng-repeat** built-in directives along with one-way binding.



## Run the app



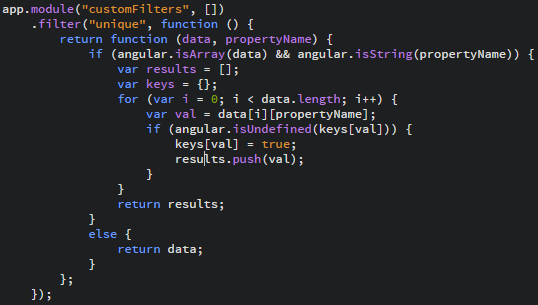
# Display Categories List

## Target

* Dynamically get unique categories from fake data to display the categories list
* To do this, we need a custom filter to filter the data.

## Creat custom filter

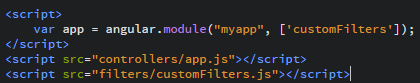
* Add new file named **customFilters.js** in Filters directory.
* Create a module ‘**customFilters’**, which have a new filter named ‘**unique’**



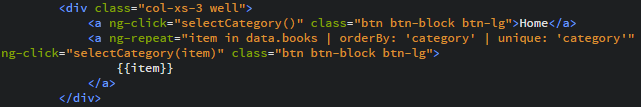
(app -> angular)

## Render categories list

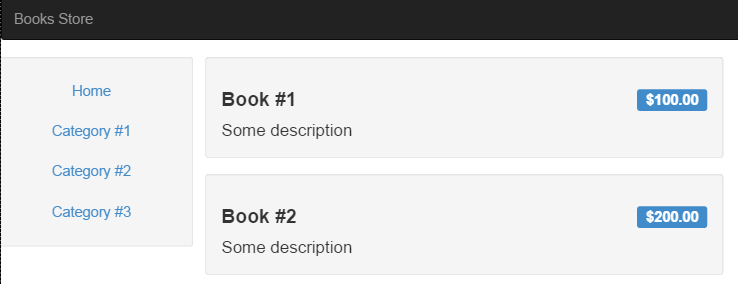
* Include customFilters.js script in index.html
* Inject dependency ‘customFilters’ module to main module.



* Binding data for Category item. Note that we are using unique custom filter to get unique data.



## run the app



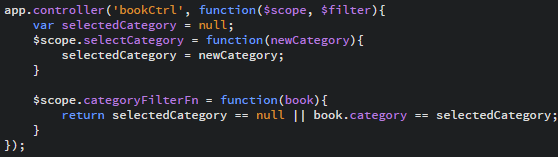
# Handle Select Category function

## Target

* Create new controller in order to respond to the user clicking event on Category button.

## Define booklist Controller

* Add new file , named **bookListController.js**, to **controllers** folder.

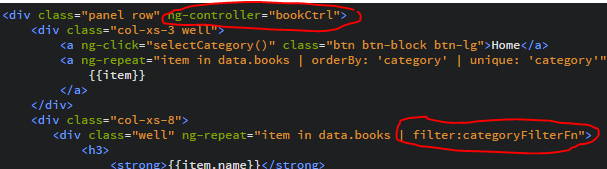


## Filter Product by Category

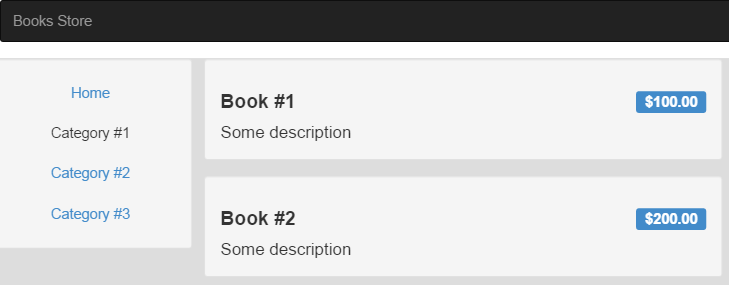
* Include bookListController.js script in index.html



* Make use **bookCtrl** for the content div, and apply filter to filter product by selected Category



## Run the app



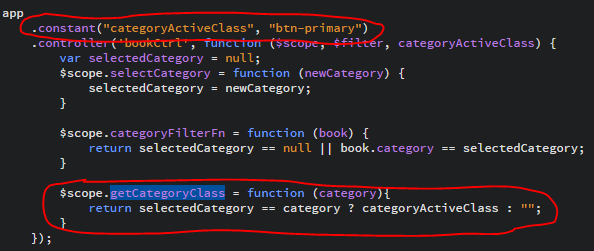
# Highlight selected category

## target

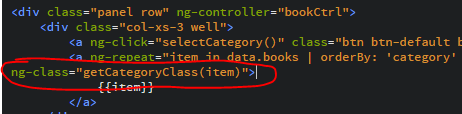
* Make use of **Constant** of angular js
* Make use **of ng-class** directive.

## Define highlight class for selected category

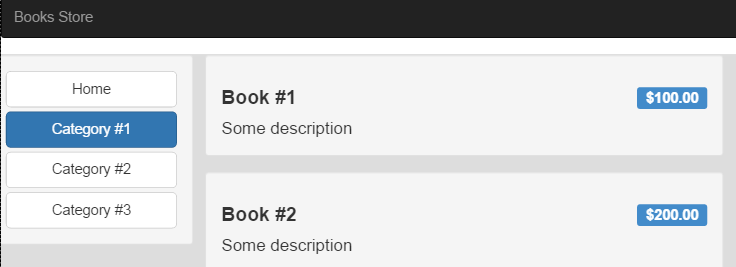
* Modify **bookListController.js**



* Add **ng-class** to Category list section (index.html) to highlight the selected category



## Run the app



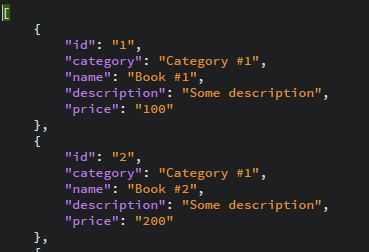
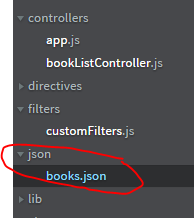
# Using service

## Target

Using Angular Service to make ajax request to retrieve JSON data

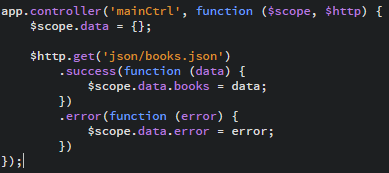
## Create JSON data model

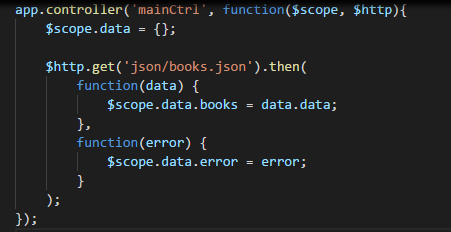
Move sample data from **$scope.books** in **mainCtrl** to separate json file (eg. **Json/books.json**)

## Retrieve data using $http

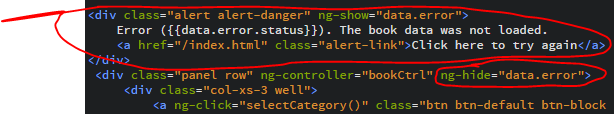
Modify the way to get data using **$http** Service in **mainCtrl** controller



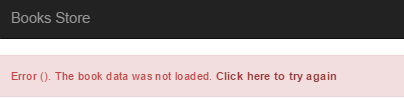


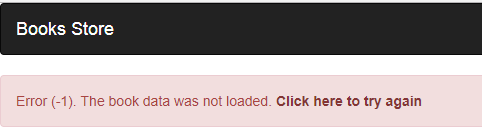
## Handling ajax error

Add div element to show the error warning in case $http service return error.



## Run the app (in case of http error)





# Create shopping cart

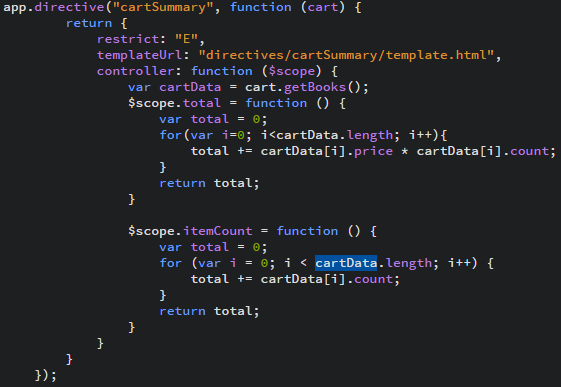
## Create Cart Module to handle business logic for cart.

* Create new file, named **services/cart.js**



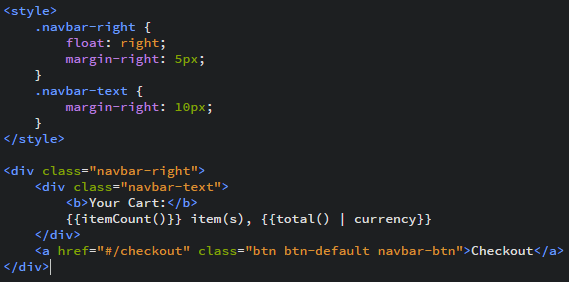
## Create cartSummary directive

* (**directives/cartSummary/directive.js**)



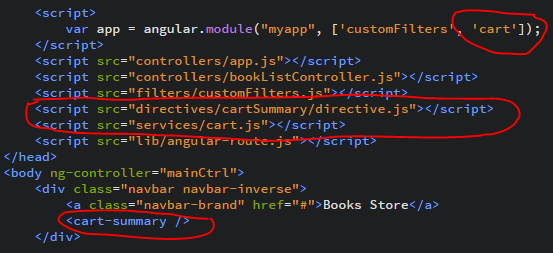
## Create cart summary template

* **(directives/cartSummary/template.html)**

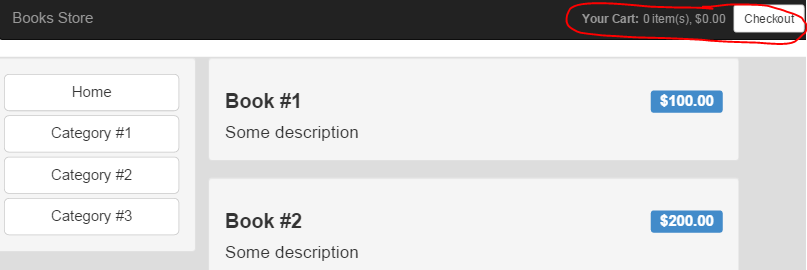


(href="#!checkout")

## Add Cart Summary directive to header

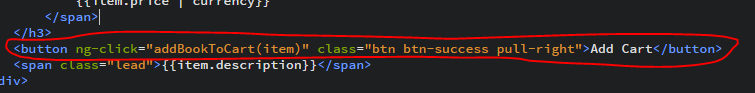


## Run the app

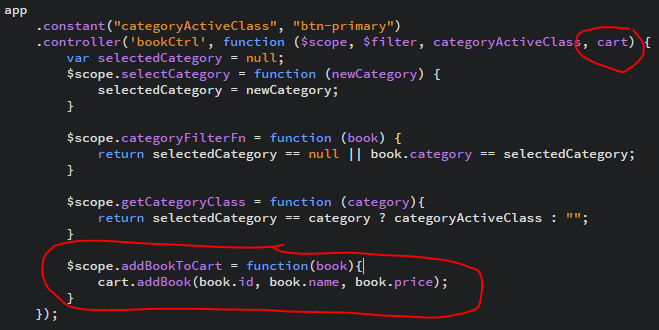


# Adding Add-to-cart function

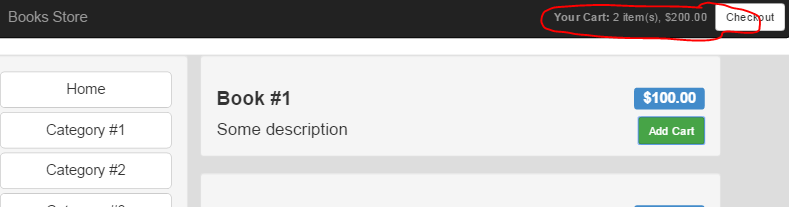
## Add an Add-to-cart button in each Book item in index.html



## Changes in bookListController.js



## Run application

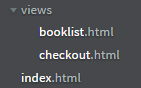


#### 

# Working with Routing

## Target

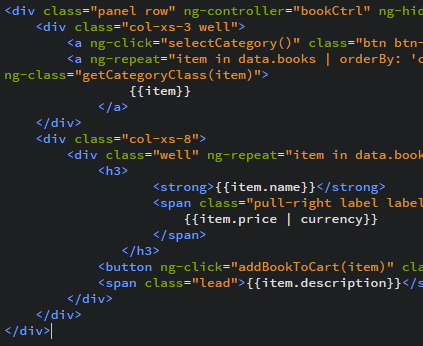
* Separate into multiple views in order to control the source code easily.



* Using **ngRoute** module to control url routing.

## Separate BookList View

* Cut and paste booklist section from index.html file to views/booklist.html.

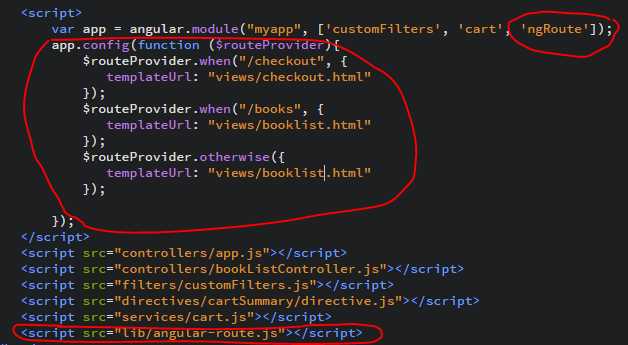


## Checkout View

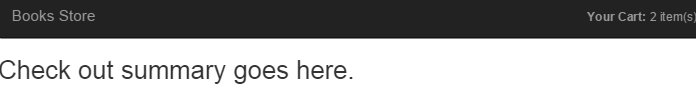


## Routing configuration

* Inject ngRoute module to main module. Then config routing using $routeProvider.



## Run the app



# Points Summary

* Working with Module, Controller, Directive, View
* Know how to use data binding
* Know how to call ajax service, constants
* Know how to create and apply custom filter.
* Know how to use some Angular built-in directives, like ng-class, ng-show, ng-hide, ng-repeat
* Know about routing in Angular using ngRoute

YEAH…. WE ARE NOW THE GURU OF ANGULAR JS.