**Exercise04\_01\_01 – Step 1**



1. Copy the ***Exercise04\_01\_01.zip*** file from Google Classroom to your Desktop. Unzip the files to create a folder in your working sandbox structure. Open the new project with your IDE. Expand the ***builds/angular*** folder and open up the ***index.html*** file. this project uses ***gulp.js*** to create and run a simple server, which will be required for AngularJS to run its ***HTTP*** and ***Router*** services. All of the ***gulp automation*** has been pre-built, in later projects we will build it ourselves. From a ***Terminal*** in the new project folder, or if your IDE is capable, run ***gulp*** to run the project in a browser.

**Exercise04\_01\_01 – Step 2**



1. Go to the ***builds/angular*** folder and open up the ***index.html*** file. Run the ***gulp*** command and examine the web page. There is an ***<input>*** field displayed that will search for and display a list of artists. We will see that running the application with ***gulp*** provides us a ***Live Preview*** of the web page, it can stay running and will reflect changes as we make them. We can watch progress in the ***server console***.
2. The first step will be to link in the AngularJS ***scripts*** that we will use to manage this application. The scripts for this project have been pre-loaded in the ***builds/angular/lib/angular*** folder. AngularJS scripts are always loaded at the ***top***, so we will place the ***<script>*** elements just above the closing ***</head>*** tag. We will start with the main AngularJS script:  
    ***<script src="lib/angular/angular.min.js"></script>***</head>
3. Now let’s implement our AngularJS ***application***. We will use the ***ng-app*** directive, which may be used to mark a section, or the whole application. We will use the whole application, so we will place it in the opening ***<html>*** tag:  
   ***<html ng-app>***
4. To start to implement our ***data binding***, we will need to use an ***ng-model*** directive. We will place this as an attribute to the ***<input>*** field that will take our data. We will name it ***”query”*** and, in essence, this actually becomes a ***variable*** in our application’s scope:  
    <input class="form-control form-control-lg"   
    id="searchQuery" placeholder="Search for artists"   
    autofocus  
    ***ng-model="query"***>
5. Let’s see how we can ***bind*** data to the ***ng-model*** directive. We can build an AngularJS ***expression*** and place it into the ***content*** of the ***<label>*** element for the ***<input>*** field as follows:  
    <label class="col-form-label text-right"   
    for="searchQuery">  
    ***search {{ query }}***</label>  
   Type some data into the input field, and watch it appear next to the ***search*** label.
6. The AngularJS expression can actually contain any type of JavaScript. So let’s make it a little more user-friendly:  
    ***search {{ 'for: ' + query }}***</label>  
   Let’s give this a test in the browser.

**Exercise04\_01\_01 – Step 3**



1. Go to the ***builds/angular/js*** folder and create a file called ***app.js***. The first step is to build a module with the AngularJS module ***constructor***. It’s parameter will be the name of the app. We also give it an array as a parameter, which would hold our dependencies, of which there are none right now:  
   ***var myApp = angular.module('myApp', []);***
2. Now let’s go to the ***index.html*** file and connect the module with the app by assigning a value to the ***ng-app*** directive in the ***<html>*** tag:  
   ***<html ng-app="myApp">***
3. Below the module declaration, we can define a ***controller*** with it’s constructor. We will name it as the first parameter. As the second parameter, we can define a function which will take ***$scope*** as its parameter:  
   ***myApp.controller('MyController', function MyController($scope) {  
      
   });***
4. Let’s add a property to the ***$scope*** object named ***artist***. :  
   myApp.controller('MyController', function MyController($scope) {  
    ***$scope.artist =***});
5. Open up the ***data.json*** file, and you will see an ***array*** of objects containing information about artists. Copy the entire first object and paste it on the right of the ***$scope.artist*** assignment statement:  
    $scope.artist = ***{  
    "name": "Barot Bellingham",  
    "shortname": "Barot\_Bellingham",  
    "reknown": "Royal Academy of Painting and Sculpture",  
    "bio": "Barot has just finished his final year at The Royal   
    Academy of Painting and Sculpture, where he excelled in glass   
    etching paintings and portraiture. Hailed as one of the most   
    diverse artists of his generation Barot is equally as skilled with   
    watercolors as he is with oils, and is just as well-balanced in   
    different subject areas. Barot's collection entitled \"The Un-  
    Collection\" will adorn the walls of Gilbert Hall, depicting his   
    range of skills and sensibilities - all of them, uniquely Barot, yet   
    undeniably different"  
    }***
6. Let’s go back to the ***index.html*** file to use the ***ng-controller*** directive to define the section that will use this controller. A module can have many controllers, but this one will just use one for the entire app. So let’s define it as an ***attribute*** of the ***<body>*** element:  
   ***<body class="bg-secondary" ng-controller="MyController">***
7. Open up the ***more.html*** file and copy its entire contents. Go to the ***index.html*** file and copy the contents in, below the comment underneath the last closing ***</div>*** element:  
    <!-- col-container -->  
     
    ***<div class="artist-list container">  
    <div class="row justify-content-center">  
    <div class="col-12 col-sm-9 col-md-7 col-lg-5">  
    <ul class=" list-group d-flex">  
    <li class="list-group-item">  
     
    <div class="media d-flex align-items-center">  
    <img class="rounded-circle mr-3"   
    src="images/\_tn.jpg"   
    alt="Photo of ">  
    <div class="media-body">  
    <h5 class="my-0 text-dark"></h5>  
    <div class="text-secondary font-italic"></div>  
    </div>  
    <!-- media-body -->  
    </div>  
    <!-- media -->  
     
    </li>  
    </ul>  
    </div>  
    <!-- col-12 -->  
    </div>  
    <!-- row -->  
    </div>  
    <!-- col-container -->***Let’s give this a test in the browser. As we can see, the new HTML shows up, but there is not yet any dynamic content in it. It also appears that our ***search*** data binding is now broken. This is happening because we have now named our ***ng-app*** to be a module. We have not yet linked that module into the HTML code.
8. Now let’s add some AngularJS ***expressions*** to the HTML. They will use the data we added to the ***$scope*** in our ***controller***:  
   <img class="rounded-circle mr-3"   
    src="images/***{{artist.shortname}}***\_tn.jpg"   
    alt="Photo of ***{{artist.name}}***">  
   <div class="media-body">  
    <h5 class="my-0 text-dark">***{{artist.name}}***</h5>  
    <div class="text-secondary font-italic">***{{artist.reknown}}***</div>  
   </div>  
   Let’s give this a test in the browser. The new AngularJS expressions show up, but there is not yet any dynamic content in them. We have still not yet linked that module into the HTML code.
9. In index.html, directly before the closing </head> tag, let’s link our module into the HTML:  
    ***<script src="js/app.js"></script>***Let’s give this a test in the browser. Now everything is looking good. We have ***dynamic*** ***content***, and our ***data binding*** on the search ***<input>*** field is now working again.
10. Now we have the rudiments of an MVC architecture. The HTML file has now become our ***View***. The $scope global object is being used in conjunction with AngularJS expressions to form our ***Model***. The ***Controller*** that we built forms the glue between them.

**Exercise04\_01\_01 – Step 4**



1. Go to the browser and turn on the ***Developer Tools***. In the Console tab, there is a ***Status 404*** error, indicating a resource was not found. If we look closely, we can see that it was thrown while trying to load one of our images. The browser does not understand the AngularJS notation because we are using an expression in the src attribute.
2. The fix for this is to replace the HTML ***src*** attribute with a special AngularJS directive, ***ng-src***, as follows:  
   <img class="rounded-circle mr-3"   
    ***ng-src="images/{{artist.shortname}}\_tn.jpg"***   
    alt="Photo of {{artist.name}}">