

Angular 2.0 For JEE Lab Book

Document Revision History

Date	Revision No.	Author	Summary of Changes
March 2018	1	Rahul Vikash	Created new lab book as per course contents

Table of Contents

<i>Getting Started.....</i>	<i>4</i>
<i>Overview.....</i>	<i>4</i>
<i>Setup Checklist for AngularJS.....</i>	<i>4</i>
<i>Instructions</i>	<i>4</i>
<i>Learning More</i>	<i>4</i>
<i>Lab 1. TypeScript Angular Component & Data Binding.....</i>	<i>Error! Bookmark not defined.</i>
<i>Lab 2. Angular Directives & Data Binding.....</i>	<i>7</i>
<i>Lab 3. Angular with Form.....</i>	<i>7</i>
<i>Lab 4. Angular Services & Pipes</i>	<i>7</i>
<i>Lab 5: Angular CRUD with Java middle ware application (Spring REST) with Router.....</i>	<i>13</i>
<i>Appendices.....</i>	<i>11</i>
<i>Appendix A: Table of Figures</i>	<i>11</i>

Getting Started

Overview

This lab book is a guided tour for learning Angular 2.0 For JEE version and above. It comprises 'To Do' assignments. Follow the steps provided to work out the 'To Do' assignments given.

Setup Checklist for Angular 2.0

Here's what is expected on your machine for the lab in order to work.

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows XP, Windows 7 or Windows 8
- Memory: 2GB of RAM (1GB or more recommended)
- Google Chrome 36.0 or Mozilla Firefox 31.0 or Internet Explorer 10 or above

Please ensure that the following is done:

- A text editor like Notepad or Notepad++ or Eclipse Luna is installed.
- Node.js install with node module

Instructions

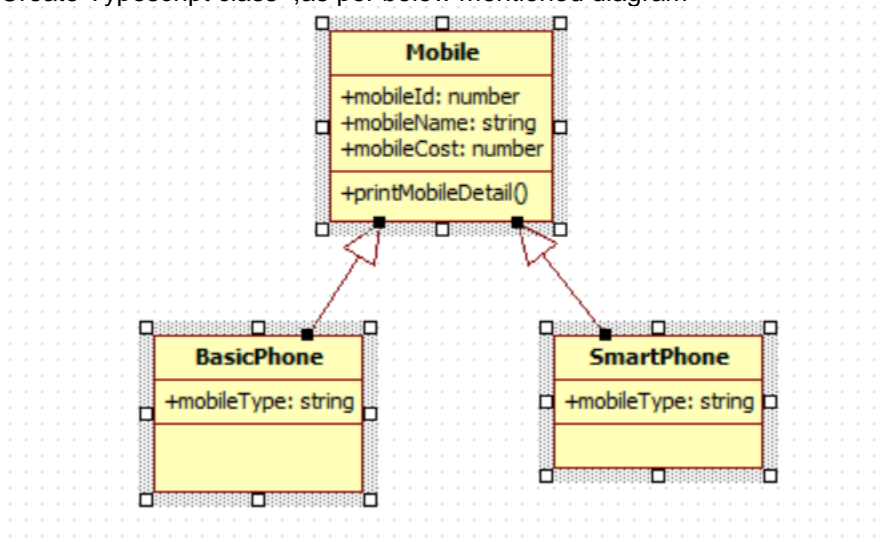
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory angular_assgn. For each lab exercise create a directory as lab <lab number>.
- You may also look up the on-line help provided in <https://v2.angular.io/docs/ts/latest/guide/setup.html>

Learning More

Lab 1. Typescript Angular Components & Data Binding

Goals	<ul style="list-style-type: none"> Understand the typescript & Angular data binding components
Time	60 minutes

1.1 Create Typescript class ,as per below mentioned diagram



All 3 classes are in separate file. We have to add mobile details such as id, name, cost & type from different typescript file & print data in console

a. Get Data from array

1.2: Create a Angular Application & display the result shown below**Angular 2 Operation**

ID	<input type="text"/>
Name	<input type="text"/>
Salary	<input type="text"/>
Department	<input type="text"/>

When we put some data in ID, name, salary, department it will come automatically come below the submit button shown below

**Angular 2 Operation**

ID	<input type="text" value="1001"/>
Name	<input type="text" value="Rahul"/>
Salary	<input type="text" value="5000"/>
Department	<input type="text" value="Java"/>

1001 Rahul 5000 Java

When we click on Add employee alert box will open & display all data inserted as shown in given image

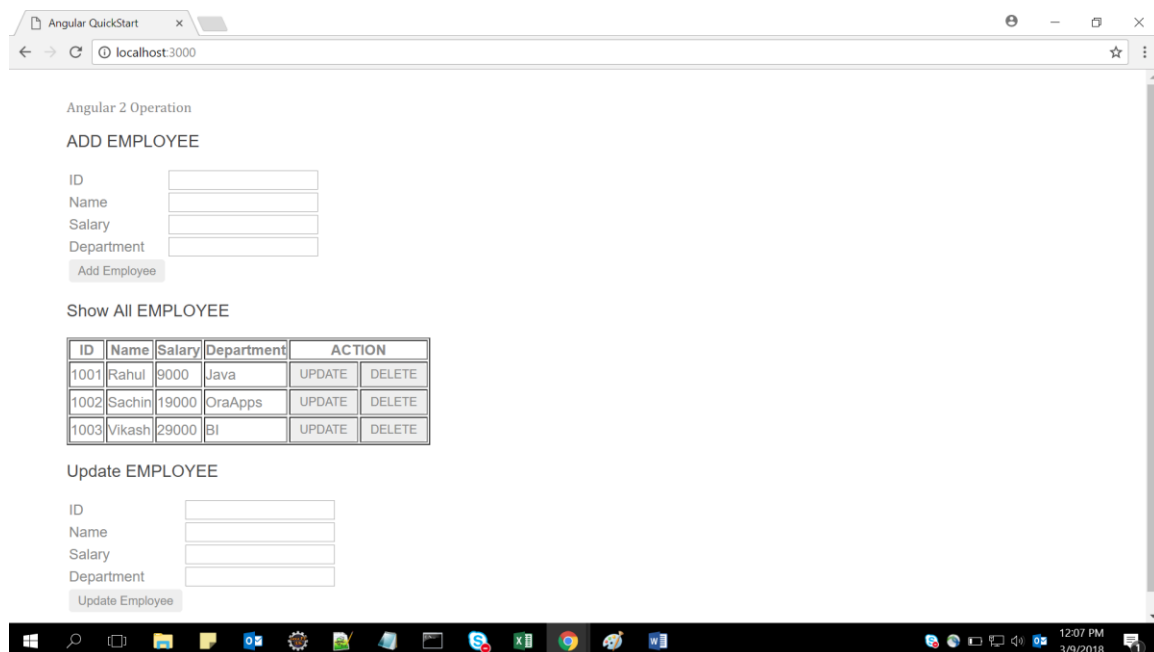


Lab 2. Angular Directives & Data Binding

Goals	<ul style="list-style-type: none"> Understand the Directives & Data Binding
Time	180 minutes

2.1. Creating a webpage to work with angular

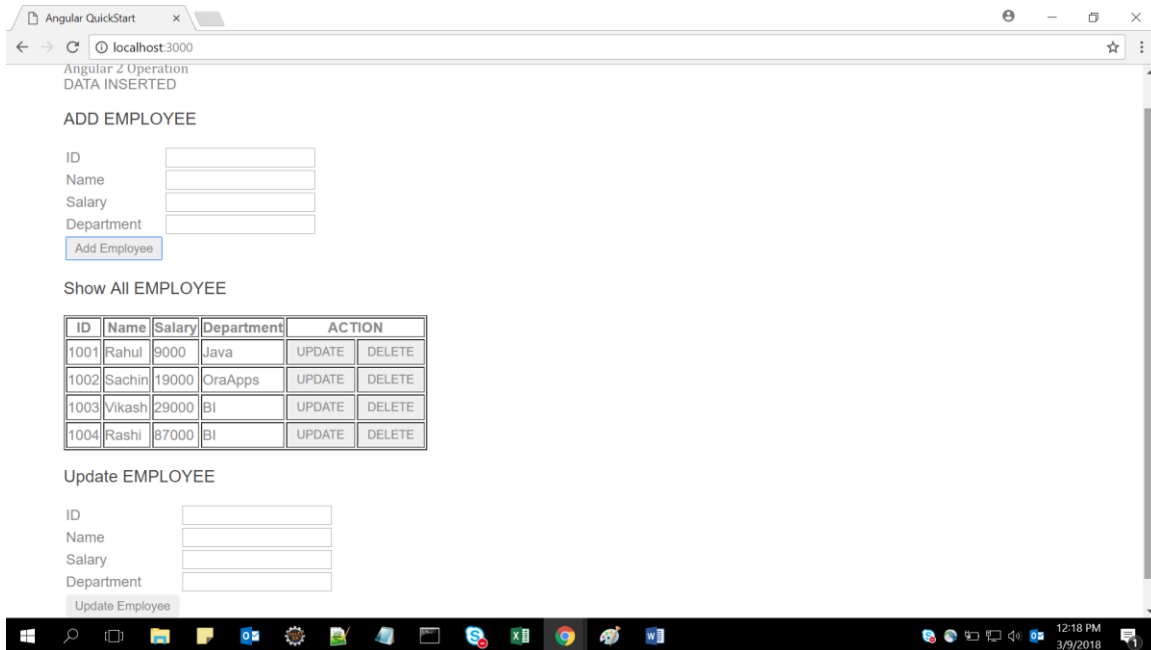
Create a web page in Angular & display the results shown below.



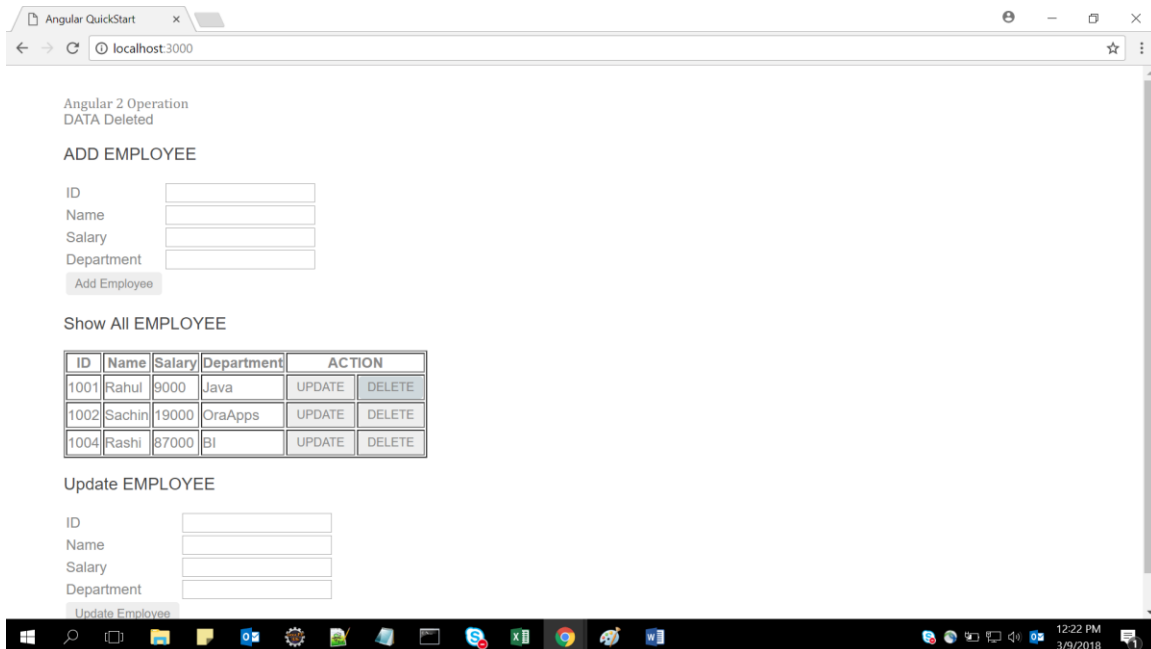
Show all data comes from array as below :

```
[
  {empId:1001,empName:"Rahul",empSal:9000,empDep:"Java"},
  {empId:1002,empName:"Sachin",empSal:19000,empDep:"OraApps"},
  {empId:1003,empName:"Vikash",empSal:29000,empDep:"BI"},
];
```

When the user enter Id, name, salary & department in Add Employee & click button Add Employee data will come in Show All Employee & message will display as shown in figure below –example inserted id 1004,name rashi,salary 87000,department BI



When we click button “Delete” that particular row will be deleted from Show All Employee & message will come “DATA Deleted” as shown in below figure –Example delete rows 1003 by pressing delete button



When we click button “Update” that particular row will be come in Update employee from Show All Employee & as shown in figure & when updated employee button click the data will update in show All employee & message will display

Angular QuickStart x

localhost:3000

Angular 2 Operation

ADD EMPLOYEE

ID

Name

Salary

Department

Add Employee

Show All EMPLOYEE

ID	Name	Salary	Department	ACTION	
1001	Rahul	9000	Java	UPDATE	DELETE
1002	Sachin	19000	OraApps	UPDATE	DELETE
1004	Rashi	87000	BI	UPDATE	DELETE

Update EMPLOYEE

ID

Name

Salary

Department

Update Employee

Windows taskbar: 12:25 PM 3/9/2018

After Update employee button clicked data updated in show all employee & message came

Angular QuickStart x

localhost:3000

Angular 2 Operation

DATA Updated

ADD EMPLOYEE

ID

Name

Salary

Department

Add Employee

Show All EMPLOYEE

ID	Name	Salary	Department	ACTION	
1001	Rahul	9000	Java	UPDATE	DELETE
1002	Dayanand	19000	OraApps	UPDATE	DELETE
1004	Rashi	87000	BI	UPDATE	DELETE

Update EMPLOYEE

ID

Name

Salary

Department

Update Employee

Windows taskbar: 12:28 PM 3/9/2018

2.2. Creating a webpage to work with angular

Create a web page and display the results shown below :

ID	NAME	SALARY	DEPARTMENT
1001	Rahul	9000	JAVA
1002	Vikash	11000	ORAAPS
1003	Uma	12000	JAVA
1004	Sachin	11500	ORAAPS
1005	Amol	7000	.NET
1006	Vishal	17000	BI
1007	Rajita	21000	BI
1008	Neelima	81000	TESTING
1009	Daya	1000	TESTING

Click on any header --Id, name, salary, department it will display the list in sorted order of that header .

Suppose Clicking on **Name** header so we will get desired output

ID	NAME	SALARY	DEPARTMENT
1005	Amol	7000	.NET
1009	Daya	1000	TESTING
1008	Neelima	81000	TESTING
1001	Rahul	9000	JAVA
1007	Rajita	21000	BI
1004	Sachin	11500	ORAAPS
1003	Uma	12000	JAVA
1002	Vikash	11000	ORAAPS
1006	Vishal	17000	BI

So by clicking any header data will come in sorted order.

Internal Array: [

```
{empId:1001,empName:'Rahul',empSal:9000,empDep:'JAVA',empjoiningdate:'6/12/2014'},
{empId:1002,empName:'Vikash',empSal:11000,empDep:'ORAAPS',empjoiningdate:'6/12/2017'},
{empId:1003,empName:'Uma',empSal:12000,empDep:'JAVA',empjoiningdate:'6/12/2010'},
{empId:1004,empName:'Sachin',empSal:11500,empDep:'ORAAPS',empjoiningdate:'11/12/2017'},
{empId:1005,empName:'Amol',empSal:7000,empDep:'.NET',empjoiningdate:'1/1/2018'},
{empId:1006,empName:'Vishal',empSal:17000,empDep:'BI',empjoiningdate:'9/12/2012'},
{empId:1007,empName:'Rajita',empSal:21000,empDep:'BI',empjoiningdate:'6/7/2014'},
```

©2016 Capgemini. All rights reserved.

The information contained in this document is proprietary and confidential. For Capgemini only. | 10 / 19

```
{empId:1008,empName:'Neelima',empSal:81000,empDep:'TESTING',empjoiningdate:'6/17/2015'},  
{empId:1009,empName:'Daya',empSal:1000,empDep:'TESTING',empjoiningdate:'6/17/2016'} ];
```

Lab 3: Angular with Form

Goals	▪ Understand the Angular form
Time	40 minutes

3. Create a registration form and perform validations on the data entered in the fields. [**Can make use of Angular Template Driven Forms or Model driven Forms**]

Product ID

Product Name

Product Cost

Product Online ☐ Yes ☐ No

Product Category

Available in Store ☐ Big Bazar ☐ DMart ☐ Reliance ☐ Mega store

Note:

- Make use of Angular 2 Forms-Model Driven or template driven.
- All fields are mandatory
- Product category will be –Grocery, Mobile, Electronics & Cloths
- When click on Add Product if all fields are validated the output will come on console
- We can select multiple store in Available list

Lab 4: Angular with Service & pipes

Goals	Working with angular http services & pipes
Time	120 minutes

4.1 Read the *booklist.json* file with the help of Angular service & display the images shown below;

Use [booklist.json](#)



Book ID	Book Title	Book Author	Book Year of Publish
1	Android for Expert	George R. R	1996
2	Complete Reference	George P. R. Martin	1998
3	BackBone JS	Kalvin R.V	2000
4	Knock Out JS	J.J Markin	2005
5	Pointer in C	Y P Kanitkar	2011
6	Big Data	Pearson D A	2011
7	A Dream of Spring	George R. R. Martin	2012
8	The Philosopher's Stone	J. K. Rowling	1997
9	The Chamber of Secrets	J. K. Rowling	1998
10	The Prisoner of Azkaban	J. K. Rowling	1999

4.2: Extended assignment 4.1 read booklist.json & apply search filter on id, title, year & author ,as shown in diagram



Book ID	Book Title	Book Author	Book Year of Publish
Id	Title	Author	Year
1	Android for Expert	George R. R	1996
2	Complete Reference	George P. R. Martin	1998
3	BackBone JS	Kalvin R.V	2000
4	Knock Out JS	J.J Markin	2005
5	Pointer in C	Y P Kanitkar	2011
6	Big Data	Pearson D A	2011
7	A Dream of Spring	George R. R. Martin	2012
8	The Philosopher's Stone	J. K. Rowling	1997
9	The Chamber of Secrets	J. K. Rowling	1998
10	The Prisoner of Azkaban	J. K. Rowling	1999

If we Search author, id, title, year only searched value will display in outputsearching K in author

Book ID	Book Title	Book Author	Book Year of Publish
Id	Title	K	Year
3	BackBone JS	Kalvin R.V	2000
4	Knock Out JS	J.J Markin	2005
5	Pointer in C	Y P Kanitkar	2011
8	The Philosopher's Stone	J. K. Rowling	1997
9	The Chamber of Secrets	J. K. Rowling	1998
10	The Prisoner of Azkaban	J. K. Rowling	1999

Searching Book TitleA

← → ↻ localhost:3000

Book ID	Book Title	Book Author	Book Year of Publish
Id	A	Author	Year
1	Android for Expert	George R. R	1996
3	BackBone JS	Kalvin R.V	2000
6	Big Data	Pearson D A	2011
7	A Dream of Spring	George R. R. Martin	2012
9	The Chamber of Secrets	J. K. Rowling	1998
10	The Prisoner of Azkaban	J. K. Rowling	1999

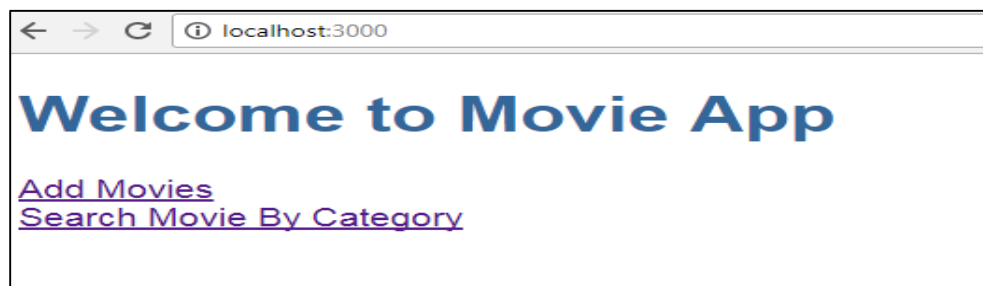
Lab 5: Angular CRUD with Java middle ware application (Spring REST) with Router

Goals	▪ Angular2 with Spring restful with JPA
Time	240 minutes

5. Create a “Movie App” wherein movies could be maintained according to genres like Fiction, Drama and Satire. There is also a search option allowing to search for movies.

Use [Movies.java](#)

Refer below screen shot:



a. When the user clicks on “Add Movies” -> Allows a User to add a movie.

The user can add movie(s) depending on Genre (drop-down list loaded with values [Drama, Fiction, Satire])

The user could add multiple movies. On Adding Movies will be added in Database with the help of Spring Web services (Restful). Use Spring restful with JPA concept Refer below screen shot:

← → ↻ ⓘ localhost:3000/addMovie

Welcome to Movie App

[Add Movies](#)
[Search Movie By Category](#)

Movie Name:

Give Rating:

Select Genre: ▼
 Select Genre!!
 Drama
 Fiction
 Satire

Refer below screens for validations:

Welcome to Movie App

[Add Movies](#)
[Search Movie By Category](#)

Movie Name:

Movie Name is required !!

Give Rating:

Rating is required !!

Select Genre: ▼
 Select Genre!!!
 Drama
 Fiction
 Satire

localhost:3000/addMovie

Welcome to Movie App

[Add Movies](#)
[Search Movie By Category](#)

Movie Name:
Movie Name should be only alphanumeric

Give Rating:
Rating needs to be a number: eg: 2.5. Rating is on Scale 1-5

Select Genre:
Select Genre!!!

- b. When the user clicks on “Search Movie By Category” -> Allows user to search movie(s) by Category. [Depending on Genre (Drama,Fiction,Satire)]

There could be more than one movies depending on same Genre.
Refer below screen shots:

localhost:3000/searchMovie

Welcome to Movie App

[Add Movies](#)
[Search Movie By Category](#)

Search Movie

Thus if Searching Movie on ‘Drama’ displays below Result:

A screenshot of a web browser at localhost:3000/searchMovie. The page has a title 'Welcome to Movie App'. Below it are links for 'Add Movies' and 'Search Movie By Category'. There is a 'Search Movie' section with a dropdown menu set to 'Drama' and a 'Search Movie' button. Below this is a table showing search results for Drama movies.

Movie Name	Rating
HomeAlone	4
HorrorSeries	4

Thus if Searching Movie on 'Fiction' displays below Result:

A screenshot of the same web browser at localhost:3000/searchMovie, but the dropdown menu is now set to 'Fiction'. The search results table below shows movies in the Fiction category.

Movie Name	Rating
IronMan	4
Avengers	4

Note:

- f. In above assignment movies are added via “Add Movies” and Searched via “Search Movie By Category”. ***These are different components.***
- g. Make use of Angular 2 Forms.
- h. Make use of services.
- i. Make use of Router Module. Which should be created as a separate module and included in the main App Module.
- j. Use Spring restful & JPA concept to add movies & search movies.
- k. All adding & searching movies done through database

Appendix A: Table of Figures

Figure 1.....	Error! Bookmark not defined.
Figure 2.....	Error! Bookmark not defined.
Figure 3.....	Error! Bookmark not defined.
Figure 4.....	Error! Bookmark not defined.
Figure 5.....	Error! Bookmark not defined.
Figure 6.....	Error! Bookmark not defined.
Figure 7.....	Error! Bookmark not defined.
Figure 8.....	Error! Bookmark not defined.
Figure 9.....	Error! Bookmark not defined.
Figure 10.....	Error! Bookmark not defined.
Figure 11.....	Error! Bookmark not defined.