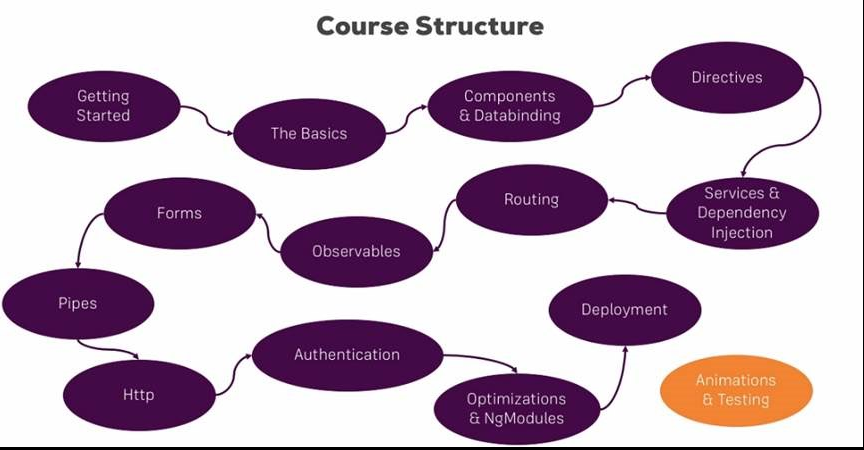
My node git credentials

Username- hemanthkaravalla



Specifically you will learn:

* + Which architecture Angular uses
  + How to use TypeScript to write Angular applications
  + All about directives and components, including the creation of custom directives/ components
  + How databinding works
  + All about routing and handling navigation
  + What Pipes are and how to use them
  + How to access the Web (e.g. RESTful servers)
  + What dependency injection is and how to use it
  + How to use Modules in Angular
  + How to optimize your (bigger) Angular Application
  + We will build a major project in this course
  + and much more!

Course started on may 22nd,should complete around june 18 may june 30th.

Start date -May 22nd

End date -June 18 th

Duration - 30 days

**1 week- 6:30 hrs. -revision on Thursday**

Basics, components and data binding, Directives, services, dependency injection, Type Script introduction, Course Roundup, Angular changes and new features.

**2 week-6:30 hrs. -revision on Thursday**

Routing, observables, forms **.**

**3rd week-6.30 hr -revision on thursday**

Pipes, Making HTTP requests, authentication and route protecting in angular app, using angular modules and optimizing apps, deploying and http client, Custom project and work flow setup.

**4th week-6.30 hr -revision on Thursday**

Working with NGRX in our project, Angular Universal, Angular Animations, Adding office capabilities with service workers, A basic introduction to unit testing in Angular apps

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| topic | No of lectures | days | Time required  (30 hrs) | Examples practice time |  |
|  |  | |  | | --- | | **1 week- 300min -6:30 hrs**  Basics,components and data binding,Directives,services,dependency injection, Type Script introduction, Course Roundup, Angular changes and new features. | |  |  |  |
| Getting started | 10 |  | 36 min |  |  |
| The basics | 33 | Wednesday | 1.53 hrs |  |  |
| Course project basics | 18 | Thursday | 1.03 hrs |  |  |
| debugging | 3 |  | 12.09min |  |  |
| Components and data binding deep dive | 20 | Friday | 1.24 hrs |  |  |
| Course project component and data binding | 6 | Saturday | 31 min |  |  |
| Directives deep dive | 11 | Saturday | 47 min |  |  |
| Project directives | 1 | Sunday | 6.25min |  |  |
| Using services and dependency injection | 12 | Sunday | 43 min |  |  |
| Course project services and dependency injection | 8 | sunday | 30.48 min |  |  |
|  |  | |  | | --- | | 2nd week - 6 :30HRS  Routing,observables,forms | |  |  |  |
| Changing pages with routing | 31 | Wednesday, Thursday | 2.18 hrs |  |  |
| Course project Routing | 15 | Friday | 45min |  |  |
| Understanding observables | 11 | Saturday | 38min |  |  |
| Course project observables | 1 | Saturday | 4 min |  |  |
| Handling forms in angular app | 34 | Sunday | 2.02 hrs |  |  |
| Course project forms | 22 | Monday and Tuesday | 1.14 hrs |  |  |
|  |  | |  | | --- | | 3 rd week - 6 :30HRS  Pipes,Making HTTP requests,authentication and route protecting in angular app,using angular modules and optimizing apps,deploying and http client | |  |  |  |
| Using pipes to transform output | 10 | Wednesday | 37 min |  |  |
| Making HTTP request | 14 | Thursday | 40 min |  |  |
| Course project Http | 5 | Thursday | 21 min |  |  |
| Authentication and route protecting in angular app | 15 | Friday | 45 min |  |  |
| Using angular modules and optimizing apps | 25 | Saturday | 1.27 hrs |  |  |
| Deploying an angular app | 4 | Sunday | 11.49 min |  |  |
| **Bonous**-The http Client | 14 | sunday | 52min |  |  |
|  |  | |  | | --- | | 4TH WEEK – 6 .30HRS | |  |  |  |
| Working with NGRX in our project | 49 | Wednesday,thrusday,Friday | 3.41 hrs |  |  |
| Angular Universal | 9 | Saturday | 27 min |  |  |
| Angular Animations | 12 | Saturday | 39.22 min |  |  |
| Adding office capabilities with service workers | 5 | Sunday | 27 min |  |  |
| A basic introduction to unit testing in Angular apps | 11 | Sunday | 45.19 |  |  |
|  |  | Topics remaining adjusted in every week |  |  |  |
| Angular changes and new features | 3 | 1 WEEK -Monday | 35 |  |  |
| Course Roundup | 2 | 1 WEEK -Monday | 2 min |  |  |
| Custom project and work flow setup | 13 | 3 RD WEEK -Monday | 51 min |  |  |
| Type Script introduction | 7 | 1 –WEEK –Tuesday and overall recall | 26 min |  |  |

**1 week- 300min -6:30 hrs -revision on Thursday**

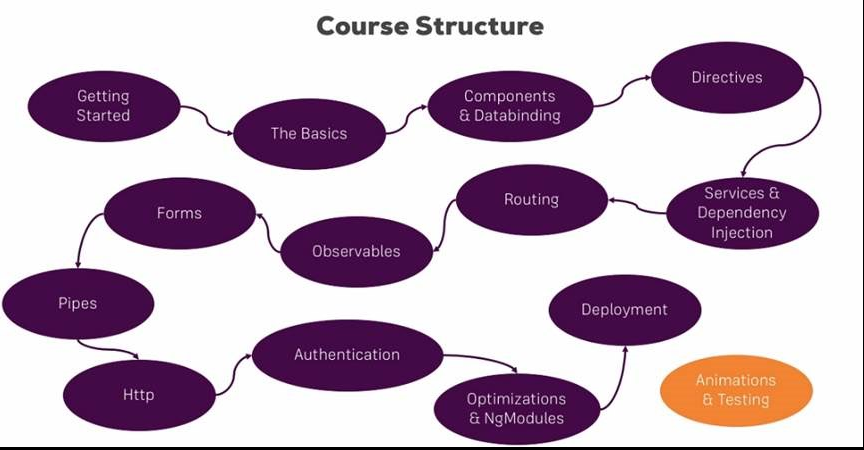
**1 week- 300min -6:30 hrs -revision on Thursday**

**1 week- 300min -6:30 hrs -revision on thursday**

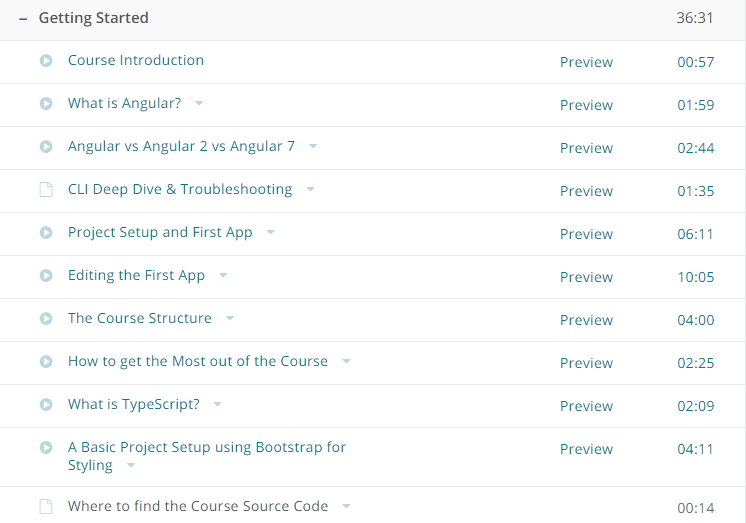
1. Introduction
   1. Angular Introduction
   2. Introduction to SPA framework
   3. Node Installation
   4. AngularJS vs Angular Comparison
2. TypeScript
   1. Introduction to Type Script
   2. TypeScript compiler
   3. TypeScript Data Types
   4. TypeScript Statements
   5. TypeScript functions and Classes
   6. Modules and Namespaces
   7. Decorators
   8. Generics
   9. Example codes
3. Angular Project setup
   1. Introduction to Angular/CLI
   2. Angular Project Structure
4. Module and Component
   1. Module
      1. Module Introduction
      2. Module Decorator
      3. Import and Export of Modules
      4. Example
   2. Component
      1. Component Introduction
      2. Component Decorator
      3. Template
      4. Typescript class
      5. Example
5. Databinding
   1. String Interpolation
   2. Property
   3. Event
   4. Two Way Data Binding
6. Component Communication
   1. Introduction to Component Hierarchy
   2. @Input Decorator
   3. @Output Decorator
   4. Event Emitter
7. Dependency Injection
   1. Introduction to DI
   2. Introduction to Services
   3. Component communication using Services
8. Directives
   1. Introduction to Directives
   2. Built-in-Directives
   3. Attribute Directives
   4. Structural Directives
9. Custom Directives
   1. @Directive decorator
   2. Custom Attribute Directives
   3. Custom Structural Directives
   4. @Input and @Output usage
   5. @HostListner
   6. Example Code
10. Lifecycle Hooks
    1. Component Lifecycle Hooks Overview
    2. Lifecycle hooks Sequence
11. Pipes
    1. Pipes introduction
    2. Built-In-Pipes
    3. Custom Pipes
    4. Angular-pipes
    5. Custom pipes Example
12. Forms
    1. Introduction to Forms
    2. NgForms
    3. Reactive Forms
       1. Forms Controls
       2. Form Group
       3. Form Array
       4. Validators

12.4 Example Code

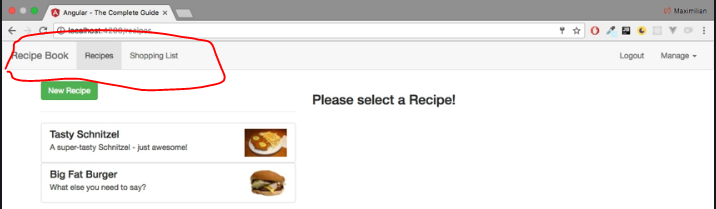
1. Pages with Routing
   1. Introduction to Routing
   2. Setting up and Loading Routing
   3. Navigating with Router Links
   4. Working with Route Parameters
   5. Router Outlet
2. Observables
   1. Introduction to Observables
   2. Reactive Programing
3. Http
   1. Introduction to Http
   2. CURD Operations
   3. GET
   4. POST
4. HttpClient
5. Build and Optimization
   1. ng build
   2. ng build vs ng serve
   3. AOT
   4. JIT
   5. Build Optimizer
6. Debugging
   1. Understanding angular messages
   2. Debugging code using Augury
7. Angular Best Practices



36min -



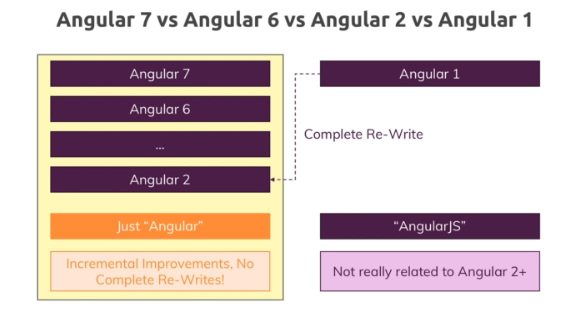
-Angular is a framework which allows you to create reactive, single page applications



you can see that we seem to visit different pages,but in the end, our page never changes.

* It's only one HTML file and a bunch of JavaScript code we got from the server and everything which you see here, every change, is rendered in the browser.
* JavaScript is much faster than having to reach out to a server for every page change and for every new,piece of data you want to display. Therefore, this approach allows you to create web applications, which look and feel almost like mobile applications; very fast!
* Well, JavaScript changes the DOM, changes whatever is displayed here (in the browser), by changing the HTML code during runtime,(so to say). That is why you never see the refresh icon on the top-left spin; because we're only changing

->the currently-loaded page. You can even see that if you inspect the source code of a page like this.That is the HTML fileand as you can see, it doesn't seem to contain the content you are seeing on this page.We only have one single HTML element which doesn't seem to be a built-in one (a native one), but that's Angular doing its job.



-complete re-write of angular1

- When it came to using browser side javascript to reach rendered a Dom to update the dom at runtime and,therefore provide highly interactive user experiences without reloading the page angular or was a complete

-angular 2 and it later versions started in 2016

-

Creating first project in Angular

npm i -g @angular/cli

ng new projectname ex-ng new my-app



