**Angular 6**

Closer : an inter function can have access to the outer function.

Ex: morningGreet.prototype

Check the available functions:

Ex: String.prototype

Memoize function: fetching result from cache memory

Functional programming usage:

They can be applied any kind of data and not restricted any particular data

ES 5

ES2015 🡪 ES 6 version enhancements

ESNext

Transpilers are used to convert ES2015 code to ES 5 Version

Babel

ES2015 Features;

1. Introduction of scoped variables and constants

let

const

1. Deconstructing

Let colors = {red, green,…blue}

Let[red, green, …] = colors

1. Arrow operator
2. New String literal

Symbol: **` `**

1. Promise API -> For Asynchrous calls

Node JS:

Is event loop based single threaded environment built on top of V8 JS engine

Used in; Netplix, youtube, hotstart.

To execute js file on Node environment:

C:\Users\Administrator\Desktop\Angular6\SG\_WS> node ./ex1.js

Package manager for Node js:

C:\Users\Administrator\Desktop\Angular6\SG\_WS\myapp> npm init

Test cmd:

test command: mocha --reporter spec

Adding dev dependences cmd:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\myapp> npm i --save-dev mocha chai request

For Prod dependences cmd:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\myapp> npm i --save mocha chai request

Run the MyApp application:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\myapp\app> node ./server.js

Mocha 🡪 testing framework

Chai --> Assertion library

To run the test cases report:

Myapp> mocha --reporter spec

Mocha installation at global (to execute command level mocha)

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\myapp\app> npm i -g mocha

To Run test cases:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\myapp\app> npm test

**TypeScript:**

SuperSet of Javascript, introduces types

Files are stored with “.ts” extension

typescript compilers are used to convert “ts” 🡪 “js”

Transpilers [BABEL] used to

Installation cmd:

C:\Users\Administrator>npm i -g typescript

To check the version of typescript:

C:\Users\Administrator>tsc –version

? 🡪 indication of optional field in type script

Execute/Run the type script file:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\tscex> tsc

To run .js file using node :

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\tscex\out> node .\thrid.js

Angualr Installation:

C:\Users\Administrator>npm install -g @angular/cli

npm config file: .npmrc

RWD ==🡺 responsive web design

@mediaquery CSS3 feature

We can use some f/wto do this;:

Bootstrap -🡪 Twitter

SPA ==🡺 Single Page Application

Challenges in SPA:

1. Data Binding (oneway binding and twoway binding)
2. Cache
3. Routing

(Multipage different pages has different URL)

<HTTP://MYSERVER.COM>

For single page we may still need different URI for views

<HTTP://MYSERVER.COM>/tv

<HTTP://MYSERVER.COM>/mobile

We need frameworks to achieve above:

Angualr, Backbone, ExtnJs, React, Vuejs,…

Angualr ------🡪 from Google

Angualr 6:

Building blocks of Angular:

**Module:**

At least one module is required per application

Module is to encapsulate components, services, directives, pipes….

Component:

* is a class which containts state and behavour like any other class
* contains templates to render which in turn can use state and behavior of that class
* styles
* selector
* contains @Component decorator

Create template project in Angular JS:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS> ng new customerapp

bootstrap: ---🡪 entry point component

main.ts 🡪 entry point of applin and loads appModule ---go to app.component.ts-🡪app.moduel.ts (bootstrap: ---🡪 entry point component)

To start the application:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> ng serve

Access application:

<http://localhost:4200>

create customer module:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> ng g class common/customer

Command short cuts:

c -> component

s-> service

g -> global

create component cmd:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> ng g c customers

\*🡪 structural directive (which can manipulate DOM)

Install bootstrap:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> npm i --save bootstrap@3

Place all images in:

C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp\src\assets

Parent ----🡪 child property data is passing using @Input

Child -----🡪 Parent and info has to be passed it can be done using @Output() and EventEmitter

[] --🡪 property

() --🡪 Event

coverageIstanbulReporter ---🡪 Code coverage representation

TestBed --🡪 provides the test environment for any component

To run the test cases:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> ng test

To check the code coverage:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> ng test --code-coverage

To reload all node\_module dependences using below command:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> npm i

In angular.cli.json

Entry point is “main.ts” and start file is “index.html”

Main.ts

-🡪 loads “AppModule” bootstrap

App.module.ts

--🡪 loads all components,services, directives, piipes

🡪 bootstrap “AppComponent” app.component.ts

App.component.html:

<app-customers></app-customers>

Loads CustomnerComponents

Customer.componet.html

-🡪app-customer-card

-🡪 loads customercardComponent

In index.html

<app-root>

Which loads AppComponent

Directive unlike component will not have template and styles.

<img src=”” width=”50px” />

Here img is an element/component

Src, width are attribues/directives

Creating directives:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> ng g directive hover

* directive - main purpose to view decoration.

**Service:**

* are injectable into components
* can contain business logic
* interaction with RESTFul web services
* sharing data between components

Installing json server (restful service data provider):

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> npm i -g json-server

To Run json server:

S C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp\src\app\common> json-server --port 3000 --watch .\data.json

Now we can access restful data:

<http://localhost:3000/customers?gender=female>

Angular service for RESTful web services:

* we can use promise API or Observable
* Include HTTPClientModule in app.module.ts
* Provides HttpCllient Service
* Add it in app.moduel.ts as => providers: [DataService],

Reactive programming - reference doc site is:

RxJSMarbles

(Observable follows Reactive programming)

Creating services:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> ng g s common/data

Creating pipe:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> ng g pipe caps

Creating service:

PS C:\Users\Administrator\Desktop\Angular6\SG\_WS\customerapp> ng g s common/shared

* Converter for int in node is : +