**ANGULAR**

**- intro**

Angular is a frontend/client side javascript framework. Created and maintained by Google. Angular is used to build powerful single page apps (SPAs).

Why use Angular?

* Rapid development and code generation
* Code organisation and productivity
* Dynamic content (can be used in HTML)
* Cross platform issues taken care of automatically
* Comes with testing capabilities

What is typescript?

* Typescript is a superset of javascript with added features
* Created by Microsoft
* Class based object-oriented programming
* Resembles languages like java and c#

**Components**

Components are sections of the user interface, e.g. navbar, blog posts.

An angular app is a tree of angular components

A component must be imported first:

Import {component} from ‘@angular/core’;

A decorator is followed:

@component ({

Selector: ‘my-app’,

Template: ‘<h1>Hello {{name}}</h1>’

})

Decorators allow us to mark a class as an angular component and provide meta-data that determines how the component should be processed, instantiated and used at run time.

Component class:

Export class AppComponent {

Name = “Angular”;

}

*You can use a separate HTML file for template using ‘templateURL’ as opposed to ‘template’ in the decorator. And putting the folder path as the property.*

**Services**

Services are used when wanting to share data across multiple components, instead of repeating code.

Import {Injectable} from ‘@angular/core’;

Import {User} from ‘./user’;

Import {USERS} from ‘./mock-users’;

@Injectables()

Export class UserService {

GetUsers(): user[ ] {

Return USERS;

}

}

**Installation process**

*DL: node,js (and npm), IDE (such as visual code studio), and gitbash (command line)*

New project

1. Open gitbash and go to directory you want your code folder to be using => *cd to folderlocation*
2. *ng new nameofapp* – wait for installation
3. *cd to nameofapp*
4. *ng serve –open* => this opens browser on the server

Files application in IDE

|  |  |
| --- | --- |
| **File** | **Description** |
| Package.json | Json file which shows dependencies and other information |
| App.modules.ts | Think of this as a meeting place for everything in your app. I.e. components, services, modules all needs to be imported in to this file and added to the ngModule directive. Each thing goes in a different array:   * components => declarations * modules => imports * services => providers |
| .html and .ts files | This is where you will be doing most of your work. Content for the page is created on these files |

**To create new component**

Example

Create new folder in ‘app’ folder called ‘components’.

Open terminal => ng g component components/user

g = generate, component = component, components/user = file destination

this will generate all the files (HTML, ts etc…) and will also update the app.module file automatically – imports and adds to declaration.

In user.components.ts file:

Under *export class* add: name = “Ali Issaee”;

In user.components.html file

Add <p>Hello {{name}}</p>

This is called ***interpolation.***

Ensure that the selector is in the main HTML file for the app (app.component.html). The selector will be in the user.components.ts file. This selector is what you call the html tag <selectorname></selectorname> in the html file.

User.components.ts

Import {component.OnInit} from ‘@angular/core’;

@Component ({

Selector: ‘app-user’,

TemplateUrl: ‘./user.component.html’,

styleUrls: [‘./user.component.css’]

})

Export class UserComponent implements OnInit {

Name: string;

Age: number;

Email: string;

Address: {

Street: string,

City: string,

State: string

}

Constructor() {

}

NgOnInit() {

This.name = “Ali Issaee”;

This.age = 26;

This.address = {

Street: “150 Grosvenor rd”,

City: “Hull”,

State: “EY”

}

}

}

|  |  |
| --- | --- |
| **Types of data** | |
| Obect | { } |
| String | String |
| Number | Number |
| Array | Any[ ] |
|  | String[ ] |
|  | Number[ ] |
| Any character | any |

If you have a large object you can create code to clean up at the bottom of the page. E.g. instead of the address object in the above example, you could change it to:

Address: address; //under export class

//at bottom of the page

Interface address {

Street: string,

City: string,

State: string

}

This can also be in a different file if needed and then imported in to keep the code clean.

In user.component.html

<h1>Hello {{name}} </h1>

<ul>

<li>age: {{age}}</li>

<li>email: {{email}}</li>

<li>{{address.street}}, {{address.city}}, {{address.state}}</li>

</ul>

<h1>Hobbies</li> //ensure hobbies array in class in .ts file.

<ul>

<li ngFor=”let hobby of hobbies”>{{hobby}}</li>

</ul>

ngFor = angulars way of looping.

Let = set variable key word

Hobby = is the key in the variable, this can be called anything

Of = keyword – to say *key of array-item*

Hobbies = name of the array.

The above can be extended to include the index number to:

<li ngFor = “let key of hobbies; let i=index”>

{{i+1}} : {{hobby}}

</li>

**Events**

Example

<button (click) = “name()”>Click me</button>

(click) = event type in parenthesis

“name()” = name of function

The function should be in the .ts file under the ngOnInit()function.

Name(){

This.name = “Lauren Foster”;

}

Example 2

HTML file

<form (submit)=”add Hobby(hobby.value)”>

<div>

<label for=”hobby”>Hobby</label>

<input type=”text” #hobby> //#hobby = id

</div>

</form>

.ts file (in ngOnInit())

add Hobby(hobby) {

console.log(hobby);

this.hobbies.push(hobby); //hobbies = name of array

return false; //e.preventDefault()

}

**Two-way data binding**

HTML file

<h1>EDIT USER</h1>

<form>

<div>

<label for=”name”>Name: </label><br>

<input type=”text” [(ngModel)] = “name” name=”name”>

</div>

<div>

<label for=”street”>Street: </label><br>

<input type=”text” [(ngModel)]=”street” name=”address.street”>

</div>

</form>

*\*[(ngModel)] = sets up 2 way data binding\**

App.modules.ts

Import {FormsModule} from ‘@angulr/forms’;

//add FormsModule under imports

**Summary**

1. What is a component?
2. What is a decorator?
3. What are services?
4. How do you create a new angular project?
5. Create a basic angular app using interpolation
6. Create a new component with name, age, email, address{ }, and hobbies [ ]
7. Use an interface at the bottom of the page for the address object to clean the code up
8. Add the example in the HTML page using interpolation
9. On the HTML page use a loop to put all list items from hobbies array on a new bullet point in a <ul>
10. Create a basic event using button click to change name of name on HTML to ‘Lauren Foster’
11. Create a more advanced event so on submit, the value from a text input is added on to a list when form is submitted
12. Give an example of 2 way data binding