Marwadi University

Marwadi University

01CE0412-Advanced Web Technology

Unit-5 Angular

Prof. Vaibhav K Matalia
Computer Engineering Department

Introduction

- Front end Framework. So This is not a language, it is a framework of javascript.
- Single page application(Open https://angular.io/ and gtu.ac.in)

Why Angular?

- Fast
- If we will create a website in core javascript it will take so much time.Angular provide multiple functions, packages from which we can easily built a website.
- Developed by Google
- MVC Framework
- Open Source
- Use NPM and command line interface.With NPM we are able to download packages. With NPM we can add Bootstrap,jQuery,validation library etc.We can use command line interface to download all packages.

- First released in 14-Sept-2016
- The current version is 17.0.0

Angular vs AngularJS

- Angular and Angularjs are not the same.
- Angularjs was released in 2014.It was developed by google.After 1 year structure of angular was totally changed. Then it was named Angular.
- Angularjs is Oldest Version not in use now

What is Angular CLI

- It provide some command from which we are able to
  - setup angular app
  - install routes,module,component
  - execute application

Marwadi University

Install angular CLI:- **npm install -g @angular/cli**

- We can download it from any directory. We can access it from any directory, because it is globally installed.
- If you want to check it is properly installed or not use **ng version** command.
- ng is command of angular CLI.

Add New Project:-**ng new app_name**(ng new first).Similar to npx create-react-app first

Execute Project:**-ng serve**

- Open browser and add localhost:4200 in URL.
- Open package.json file if you want to check which version of angular is installed.
- Version of Angular CLI and Angular will same.

How to install old version of angular:-npm install -g@angular/cli@10.0.0

Project Structure in Angular

.vscode:-This is supporting folder of vs code. This folder is not related to the angular project. If you run the project, some file will be automatically created in this folder.

node_modules:-When you create a new project, many packages are installed. The details of these packages will be in the node folder.

app:-This is one component(default). All the files you create related to the project will be created in the app folder.

app-routing.module.ts:-Used for navigation.

app-component.html:-Template file.

app-component.css:-Used for web page designing.

app-component.ts:-Logic,module,property

app-component.spec.ts:-Used for testing purpose

app.config.ts:-store configuration related to application

app.config.server.ts:-store configuration related to server.

asset:-Used to store website images,icon.

favicon.ico:-

editorconfig:-Used when multiple developer works on same project. Team leader define some rule inside this file and all other developers need to follow that rules. If they don’t follow rules compile time error occurs.

.gitignore:-

angular.json:-configuration file

tsconfig:-configuration file of typescript.

Component

- Angular application cannot be developed without components.
- Minimum one component always present inside an angular application.
- If we create an angular application, it always generate one default component.
- Only the index file will run. All the component will load and unload based on requirement inside index file.

Create component:-

ng generate component user-list OR ng g c user-list

- All the components consist of four files
    1. component.ts:-This is main file. We need to write logic,module,properties etc in this file.
    2. component.html:-All the component has its own template file.
    3. component.css:-
    4. component.spec.ts:-This file is used for testing purpose. If you delete this file, there will be no difference in the project

Step 1:-Create one component named as user-list by below command

Ng generate component user-list

Step 2:-open template file of user-list component and add

<h1>User List Component Called</h1>

Step 3:-Copy selector of user-list and add inside app.component.html

<app-user-list></app-user-list>

Step 4:-Copy class name from user-list.ts file and add it inside imports metadata in app.component.ts

component with inline style and template

- If you use inline style and template, the css and html that comes with component is not needed.

When to use inline style?

- Less number of style
- Don’t need to generate an extra file

Step 1:-Open app.component.ts file and comment styleURL metadata and add below code

```
styles:`h1{color:red;background-color:yellow}
  p{font-size:24px}`
```

Step 2:-Open app.component.ts file and comment templateURL metadata and add below code

```
template:`<h1>Hello world</h1>`
```

Routing

1.Generate new file structure of Angular by using below command
ng new routing-blog

2.Add below component. If user click on hyperlink below component will load
ng g c aboutus
ng g c contactus
ng g c dashboard

3.Add below code inside aboutus component
Aboutus.html
<h2>About Us</h2>
Lorem ipsum
aboutus.css
h2
{background-color: brown;
}

4.Follow step 3 for contactus and dashboard component.

5.Open app.component.html and add below code

```
<a routerLink="home">Home</a>|
<a routerLink="aboutus">About Us</a>|
<a routerLink="contactus">Contact Us</a>
<router-outlet></router-outlet>
```

The component will be loaded wherever you place the router-outlet.

6.Open app.routes.ts file and add below code

```
import { DashboardComponent } from './dashboard/dashboard.component';
import { AboutusComponent } from './aboutus/aboutus.component';
import { ContactusComponent } from './contactus/contactus.component';
export const routes: Routes = [
    {
        path:'home',component:DashboardComponent
    },
    {
        path:'aboutus',component:AboutusComponent
    },
    {
        path:'contactus',component:ContactusComponent
    }
];
```

Step 8:-Open app.component.ts and modify imports metadata.Add RouterLink inside imports metadata.

Step 9:-setpage title

Open app.routes.ts file and add title attribute

Step 10:-wildcard route

a) This is used to display page not found error message.First create new component to handle pagenotfound error message.

b) Type ng g c pagenotfound and create new component.Add some HTML and CSS.

c) Open app.routes.ts file and add below object

```
{
path:'**',component:PagenotfoundComponent,title:'page not found'
}
```

- Compiler will not check the remaining routes once it find **. So At last we need to add wildcard routes.

Step 11:set BaseURL

- Add below object inside app.routes.ts file

```
{
    //path:'',redirectTo:'home',pathMatch:'full'
    path:'',component:DashboardComponent
}
```

- path:’’,redirectTo:’home’ means If the path is empty, i.e. if the user has requested for the home page, redirect the user to the home route. pathMatch check path is fully empty or not.

Step 12:-activate link

- RouterLinkActive is a directive for adding or removing classes from HTML element that is bound to RouterLink.The main use of this directive is to highlight which route is currently active.
- Open styles.css and add below code

```
.active
{
    background-color: green;
    color: white;
    padding:10px;
    font-size: 22px;
}
```

- Open app.component.html and add below code

```
<a routerLink="home" routerLinkActive="active">Home</a>|
<a routerLink="aboutus" routerLinkActive="active">About Us</a>|
<a routerLink="contactus" routerLinkActive="active">Contact Us</a>
<router-outlet></router-outlet>
```

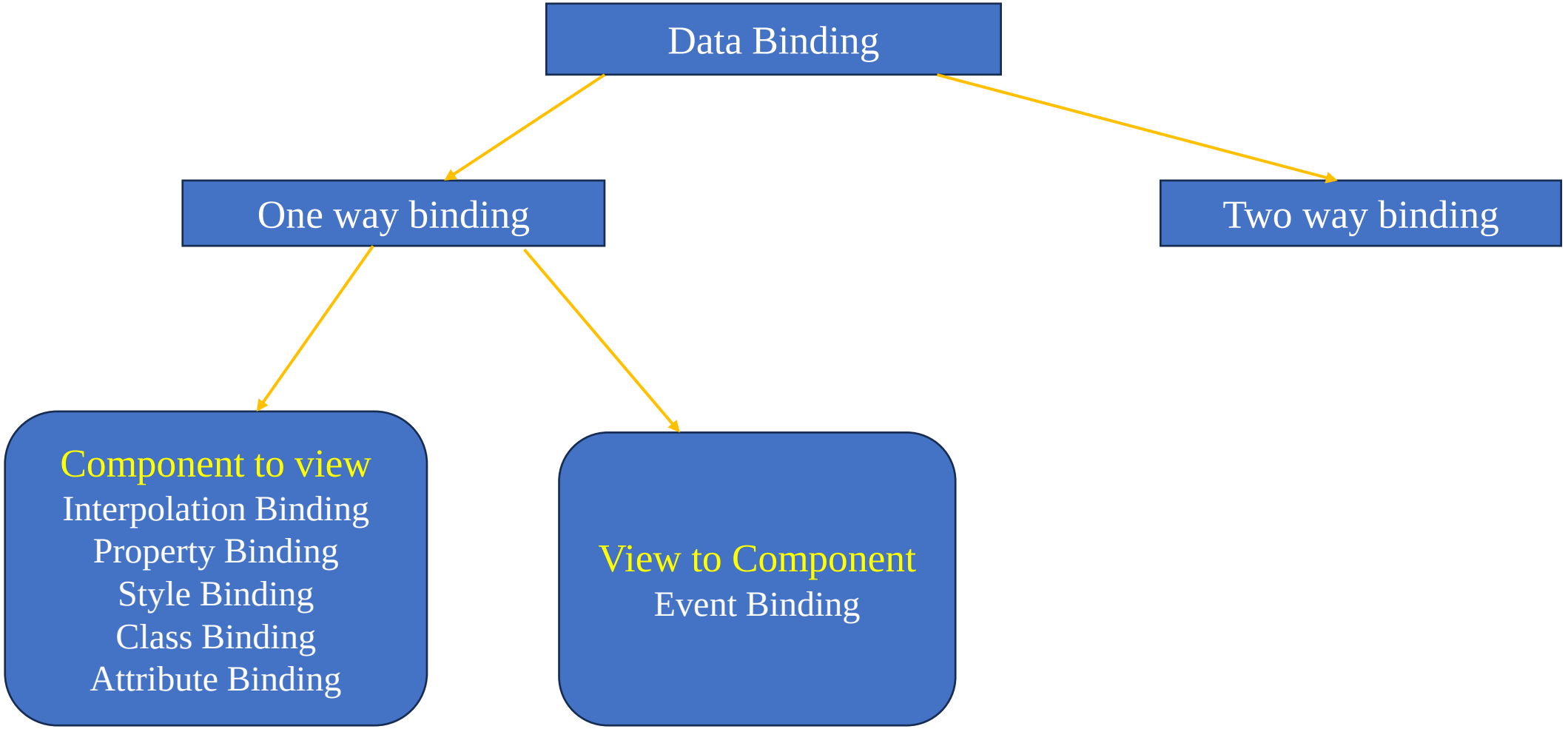
- Import routerLinkActive in app.component.ts

Data Binding

- Data Binding means interacting with data.So,we can say that the interaction between templates and business logic is called data binding.
- Communication will take place between typescript and HTML file.

Possibility 1:-We can access the properties, functions in HTML file which are defined inside typescript file.

Possibility 2:-HTML file has one textbox, whatever input the user gives should come in a typescript file.


Data Binding
One way binding
Two way binding
Component to view
Interpolation Binding
Property Binding
Style Binding
Class Binding
Attribute Binding
View to Component
Event Binding

Interpolation

- Used to display dynamic data.
- When we declare any property or data inside app.component.ts file then it is known as dynamic data.

Example 1:-

app.component.ts:-

```
export class AppComponent {
  title = 'first';
}
```

App.component.html:-

```
<h1>{{title}}</h1>
```

Example 2:-

app.component.ts

```
export class AppComponent {
  title = 'first';
  fName="virat";
  lName="kohli"
}
```

app.component.html:-

```
<h1>{{fName+' '+lName}}</h1>
```

Example 3:-

<h1>2+2</h1>

<h1>{{2+2}}</h1>

<h1>{{title==“first”}}</h1>

<h1>{{title=“second”}}</h1> <!-- Error-->

<h1>{{typeof title}}</h1><!-- Error-->

<h1>{{data++}}</h1><!– First add data property in typescript file Error-->

Property Binding:-

- It is used to change the properties of input field or html field.

Example:-

```
export class AppComponent {
  title = 'first';
  x='virat’;
}
<input type="text" value={{x}}/><br/><br/>
<input type="text" [value]=x/>
```

Difference between Interpolation and Property Binding:-

Interpolation accept string and numeric value. Boolean values cannot be added.

```
export class AppComponent {
  title = 'first';
  x=false;
}
<input type="text" disabled={{x}}/><br/><br/>
<input type="text" [disabled]=x/>
```

- Boolean value will not work in interpolation. Here false is considered as a string, so the textbox will remain disabled.

Event Binding:-

- In Angular we have to write event name inside round bracket.
- The event name will be in lowercase.

Example 1:-

```
<button (click)="f1()">Click Me!</button>
export class AppComponent {
  title = 'first';
  f1()
  {
    //console.log('event triggered');
    alert('event triggered');
  }
}
```

Example 2:-

```
<input type="text" #box (keyup)="f1(box.value)" placeholder="Event keyup"/>
export class AppComponent {
  title = 'first';
  f1(val:any)
  {
    console.log(val);
  }
}
```

- Add keydown,blur event instead of keyup in above example.

Example 3:-

```
<h1 (mouseover)="f1()">App component called</h1>
f1()
  {
    console.log('Mouse over event triggered');
  }
```

Example 4:-get InputBox value and print it in browser not in console

```
<input type="text" #box (keyup)="f1(box.value)">
<h2>{{userData}}</h2>
userData:any=";//userData="
  f1(val:any)
  {
    //console.log(val);
    this.userData=val;}
```

Example 5:Display value when user clicks on button

```
<input type="text" #box><br/><br/>
<button (click)="f1(box.value)">Click Me!</button><br/><br/>
{{userData}}
```

Typescript file:-

```
userData='';
  f1(val:string)
  {
    //console.log(val);
    this.userData=val;
  }
```

Example 6:-Counter Example

```
<button (click)="f1('add')">++</button> <button (click)="f1('minus')">--
</button><br/><br/>
<h1>{{count}}</h1>
```

Typescript file:-

```
count=0;
  f1(val:string)
  {
    val=='add'?this.count++:this.count--;
  }
```

Example:-show and hide password

```
showpassword:boolean=false;
  password:string='';
  f1()
  {
    this.showpassword=!this.showpassword;
  }
<input type="checkbox" (change)="f1()"/> 
<input type="{{showpassword?'text':'password'}}"/>
```

Example:-Copy Temporary Address in Permanent Address

```
<textarea [(ngModel)]="address1"></textarea><br><br>
<input type="checkbox" [(ngModel)]="copyAddress" (change)="f1()"><br><br>
<textarea [(ngModel)]="address2"></textarea>
```

```
copyAddress:boolean=false;
  address1:any='';
  address2:any='';
  f1()
  {
    if(this.copyAddress)
      {
        this.address2=this.address1;
      }
```

```
else
    {
      this.address2='';
    }
  }
```

- The **ngModel directive** is a directive that is used to bind the values of the HTML controls (input, select, and textarea) or any custom form controls, and stores the required user value in a variable and we can use that variable whenever we require that value.

Directive

- Directives extend the power of HTML by making HTML dynamic.
- The directive changes the structure and behavior of the DOM.
- In other language we used conditional statements, loops and switch case, but in HTML we are not able to use it.
- It is not possible in other technology that we can use looping, conditional statement inside HTML.
- Directives are used in all applications in Angular.
- The directive is divided into 3 parts
  Component Directive
  Structural Directive
  Attribute Directive

Component Directive:-

- Every component is a directive.
- This is the special directive which is always present in an angular app where each of our HTML page associated with this directive.
- Component Directive is a class with @Component decorator function.As we know that angular app contains at least one component called AppComponent which is present inside App-component.ts file.
- In this file, we see with the help of selector,@Component which is a decorator function is used to create a component directive.

Structural Directive

- Directly changes the structure of the component's DOM.
- There are basically 3 built in structural directive available in angular
  NgIf(@ngIf)
  NgFor(@ngFor)
  NgSwitch(@ngSwitch)

Example 1:-

```
@if(true)
{
   <div>
      Hello world
   </div>
}
```

Example 2

```
<button (click)="f1()">LogIn</button> 
<button (click)="f2()">LogOut</button>
@if(isLoggedIn)
{
    <h2>User is LoggedIn</h2>
}
@else
{
    <h2>User is LoggedOut</h2>
}
```


```
isLoggedIn:boolean=false;
  f1()
  {
    this.isLoggedIn=true;
  }
  f2()
  {
    this.isLoggedIn=false;
  }
```

Attribute Directive:-

- The attribute directive changes the appearance or behaviour of a DOM element.
- With attribute directive we can change fontcolor,background color etc dynamically at run time.

Pipes

- Transfers the format of data from one form to another.
- Pipes will be used in html file, cannot be used in typescript file.

```
export class AppComponent {
  title = 'first';
  subject="Advanced Web Technology";
  today=Date();
}
<h1>{{subject}}</h1>
<h1>{{subject|uppercase}}</h1>
<h1>{{subject|lowercase}}</h1>
<h1>{{subject|titlecase}}</h1>
<h1>{{today}}</h1>
```

```
<h1>{{today|date}}</h1>
<h1>{{today|date:'fullDate'}}</h1>
```

Example 2:-Without pipes we can achieve above functionality

```
export class AppComponent {
  title = 'first';
  capString(item:string)
  {
    return item.toUpperCase();
  }
}
<h1>{{capString(title)}}</h1>
```

Example 3:-

```
export class AppComponent {
  title = 'first';
  subject="Advanced Web Technology";
  cricketer={
    'name':'virat kohli',
    'team':'RCB',
    'city':'mumbai'
  }
}
```

```
<h1>{{subject|slice:3}}</h1>
<h1>{{subject|slice:3:6}}</h1>
<h1>{{subject|slice:3:6|uppercase}}</h1><!--we can use two pipes at a time-->
<h1>{{cricketer}}</h1>
<h1>{{cricketer|json}}</h1>
<h1>{{20 | currency:'USD'}}</h1>
<h1>{{20 | currency:'INR'}}</h1>
```

Module

- Module in Angular referes to a place where you can group the components,directives,pipes and services which are related to the application.
- By creating different modules, components can be added as per the requirement of the application.

Create Module:-ng generate module user-auth

Create Component inside module:-

ng generate component user-auth/register

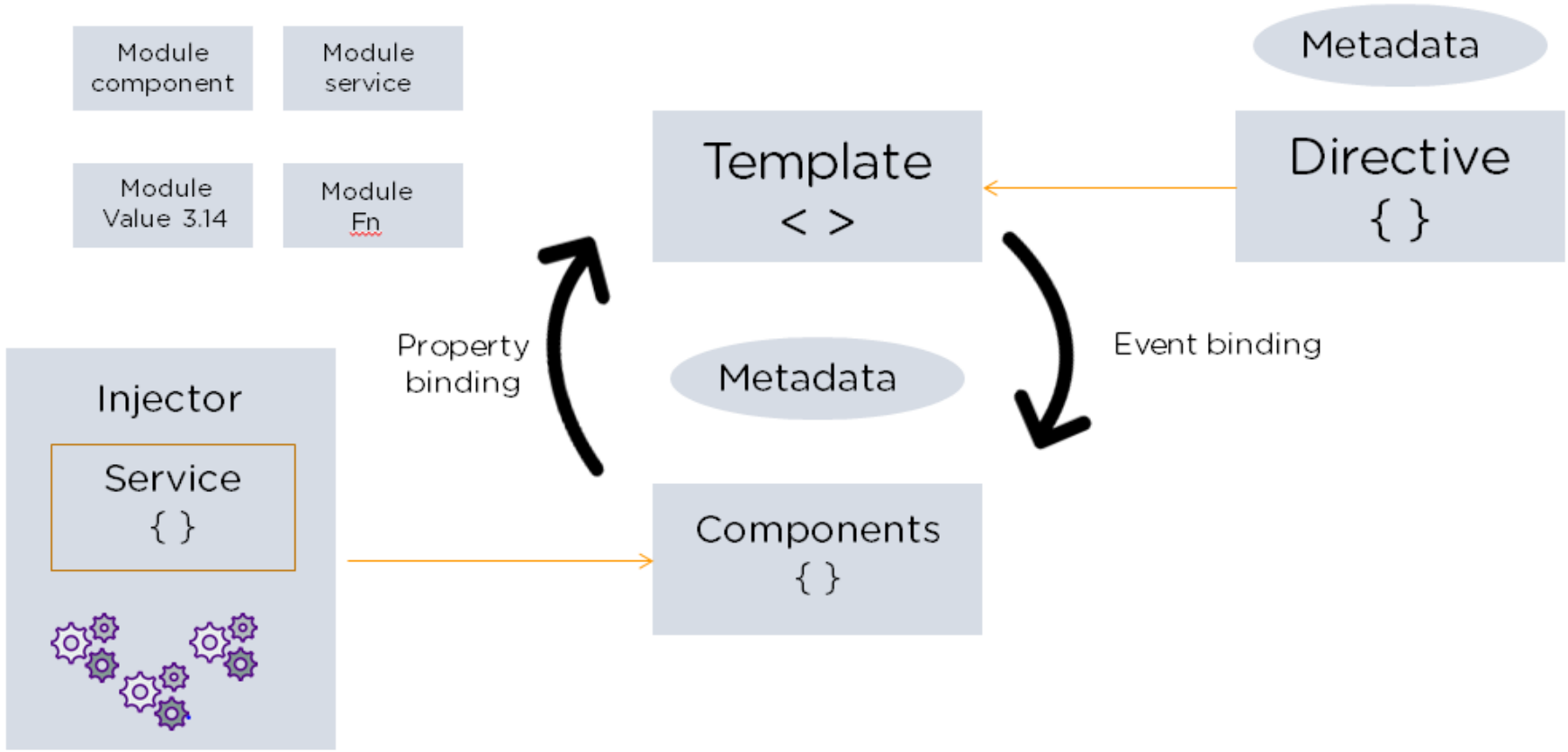
ng generate component user-auth/login

ng generate component user-auth/forger-password

ng generate component user-auth/change-password

Architecture of Angular

Marwadi University



Form

- First import formsmodule in typescript file.

```
<form #basicForm="ngForm" (ngSubmit)="getData(basicForm)">
    <input type="text" name="fname" placeholder="Enter your firstname" ngModel/><br/><br/>
    <input type="text" name="lname" placeholder="Enter your lastname" ngModel/><br/><br/>
    <input type="text" name="email" placeholder="Enter your email" ngModel/><br/><br/>
    <input type="submit">
</form>
<h2>First Name {{userData.fname}}</h2>
<h2>Last Name {{userData.lname}}</h2>
<h2>Email {{userData.email}}</h2>
```

```
export class TemplateformComponent {
  userData:any={}
  getData(basicForm:NgForm)
  {
    //console.log(basicForm.value);
    this.userData=basicForm.value;
  }
}
```

Form validation

Import commonModule package.

```
<form #basicForm="ngForm" (ngSubmit)="getData(basicForm)">
    <input type="text" name="fname" placeholder="Enter your firstname" required ngModel #answer1="ngModel"/>
    <span *ngIf="answer1.touched && !answer1.valid">First name is required</span><br/><br/>
    <input type="text" name="lname" placeholder="Enter your lastname" required ngModel #answer2="ngModel"/>
    <span *ngIf="answer2.touched && !answer2.valid">Last name is required</span>
    <br/><br/>
```


```
<input type="text" name="email" placeholder="Enter your email" required ngModel
#answer3="ngModel"/>
    <span *ngIf="answer3.touched && !answer3.valid">Last name is required</span>
    <br/><br/>
    <input type="submit" [disabled]         ="basicForm.invalid">
</form>
```

Todo App

Marwadi University

app.component.html

```
<input type="text" placeholder="Enter Task" #task><br><br>
<button (click)="addTask(task.value)">Add Task</button><br><br>
<ul *ngFor="let item of list">
    <li>{{item.name}} <button (click)="remove(item.id)">Remove</button></li>
</ul>
```

app.component.ts:-

```
export class AppComponent {
  title = 'first';
  list:any=[];
  addTask(item:string)
  {
    this.list.push({id:this.list.length+1,name:item});//console.log(this.list);
  }
  remove(id:number)
  {//console.log(id);
    this.list=this.list.filter((item:any)=>item.id!=id);}}
```

Services

- It is used to reuse the code.
- If there is any functionality in the website which needs to be used in many components then services can be used for this.
- We can write logic inside service file.

Step 1:-Create service file

Ng generate service message

Note:-Injectable decorator turns any class into a service.

Step 2:-Add getmessage() inside message.service.ts file

```
export class MessageService {
  fullName='Virat Kohli';
  constructor() { }
  getmessage()
  {
    return this.fullName;}
}
```

Step 3:-Open app.component.ts and add below code

```
export class AppComponent {
  title = 'second';
  getFullName:string='';
  constructor(private obj:MessageService)
  {
    this.getFullName=obj.getmessage();
  }
}
```

Step 4:-Open app.component.html file and add {{getFullName}}

Marwadi University

Thank You