

INF354 Advanced Programming

Angular 1 2025 Practical slide

XXXXX



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

**Faculty of Engineering,
Built Environment and
Information Technology**

Fakulteit Ingenieurswese, Bou-omgewing en
Inligtingtegnologie / Lefapha la Boetšenere,
Tikologo ya Kago le Theknolotši ya Tshedimošo

Make today matter

www.up.ac.za

Outline

- Develop a simple angular app to demonstrate some of the features discussed in the theory lecture



**Faculty of Engineering,
Built Environment and
Information Technology**

Fakulteit Ingenieurswese, Bou-omgewing en
Inligtingtegnologie / Lefapha la Boetsenere,
Tikologo ya Kago le Theknolotši ya Tshedimošo

Make today matter

www.up.ac.za

Step 1: create a angular project

In command line prompt

Change directory to folder where you want to create your app

Create the application project using following commands:

(1) `ng new MyApp` (* MyApp is the project name, it can vary)

(2) `cd MyApp`

(3) `code.` (Opens angular project in VS code)

Run command: `ng serve -o` to launch your angular app



**Faculty of Engineering,
Built Environment and
Information Technology**
Fakulteit Ingenieurswese, Bou-omgewing en
Inligtingtegnologie / Lefapha la Boetšenere,
Tikologo ya Kago le Theknolotši ya Tshedimošo

Make today matter

www.up.ac.za

Step 2 : create the components

In this example we want to create two components (login and taskview)

- To create components:
 - In vs code: open terminal
- Enter create component command:
 - `ng generate component login` OR `ng g c login`
 - `ng generate component taskview` OR `ng g c taskview`



Faculty of Engineering,
Built Environment and
Information Technology
Fakulteit Ingenieurswese, Bou-omgewing en
Inligtingtegnologie / Lefapha la Boetšenere,
Tikologo ya Kago le Theknološi ya Tshedimošo

Configure routes

in `app.routes.ts`

- Set routes to the components
- In route array:

```
{path:'',redirectTo:'login', pathMatch:'full'}, //setting login page as default page
{path:"login", component:LoginComponent},
{path: "taskview", component:TaskviewComponent},
{path:'**',component:LoginComponent}, //set login page as default if wrong page
entered
```



Faculty of Engineering,
Built Environment and
Information Technology

Fakulteit Ingenieurswese, Bou-omgewing en
Inligtingtegnologie / Lefapha la Boetšenere,
Tikologo ya Kago le Theknolotši ya Tshedimošo

Make today matter

www.up.ac.za

Designing the UI: Bootstrap templates (html)

- You can create the UI from scratch
- Or you can use **Bootstrap templates** from <https://getbootstrap.com/docs/>
- A frontend toolkit, with predefined templates

Install bootstrap into angular:

In terminal, run command: `npm install bootstrap`

**installed bootstrap will be in node_modules folder*

Configure Angular to use Bootstrap templates

in angular.json

- Look for “styles” and “scripts” property :
- In style :
`"node_modules/bootstrap/dist/css/bootstrap.min.css",`
- In scripts
`"node_modules/bootstrap/dist/js/bootstrap.min.js"`



Faculty of Engineering,
Built Environment and
Information Technology
Fakulteit Ingenieurswese, Bou-omgewing en
Inligtingtegnologie / Lefapha la Boetšenere,
Tikologo ya Kago le Theknolotši ya Tshedimošo

Use bootstrap templates for quick UI design

Go to : <https://getbootstrap.com/docs>

Search for forms : select desired template for login page and paste in [login.component.html](#) file

NB : you can edit the template as you want

Style the template

in : login.component.css file

Style the template: for this example we used the following :

```
.logo{
  text-align: center;
  font-family: Verdana;
  font-weight: bold;
}
.login-Container{
  width:270px;
  height:200px;
  padding:10px;
  background-color: white;
  border-radius: 10px;
  position: absolute;
  top:100px;
  left:310px;
  box-shadow:5px 5px 20px #383838;
}
.btn{
  margin-top: 5px;
}

.error{
  position:absolute;
  top: 163px;
  right:15px;
  color: firebrick;
}
```

Add logic for the login page

- Create a service (in terminal)
 - `ng generate service auth`
- We don't have a database, we are going to set our username and password values in our service, we will later inject this into the component.

In `auth.service.ts`

```
login(uname:string, passw:string)
{
    if(uname==='john' && passw ==='12345'){
        return 200;
    }else {
        return 403;
    }
}
```

Add logic for the login page

- In: **login.component.ts**
- Implement the login method, which will be injected with the value from the service, and will do validation of values
- Imports:
 - Service: `import { authService } from '../services/auth.service';`
 - Forms module : `import {FormsModule} from '@angular/forms';`
- `@Component({`
- `selector: 'app-login',`
- `imports: [FormsModule],`
- `templateUrl: './login.component.html',`
- `styleUrl: './login.component.css'`
- `})`

*** we need to import FormsModule to use ngModel directive to 2 way bind the form input with the values in component


```

import { Component, OnInit } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { AuthService } from '../services/auth.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-login',
  imports: [FormsModule],
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
  //declare variables for the username and password to be
  injected
  username:string="";
  password:string="";
  errorMsg:string="";

  //inject service in constructo
  constructor(private auth:AuthService, private
  router:Router){}

  //create the onInit function: performs initialisation of
  components the use services
  ngOnInit(): void {
  }

```

```

//lets implement the login function with some
validation

async login()
{
  if(this.username.trim().length === 0){
    this.errorMsg = "Username is required";
    return;
  }else if(this.password.trim().length===0)
  {
    this.errorMsg="Password required";
    return;
  } try{
    const resp = await
    this.auth.login(this.username, this.password);//
    injected the service fuction login here

    if(resp===200)

    { this.router.navigate(['taskview']);/* to
    navigate to taskview page if login is successful
    */
    }

    else if(resp===403)
    {this.errorMsg= "Invalid Credentials";
    }

    else {this.errorMsg ="unexpected error occured";
    }}catch(error){
      console.error("Login error: ",error);

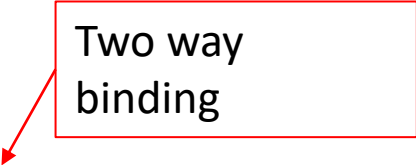
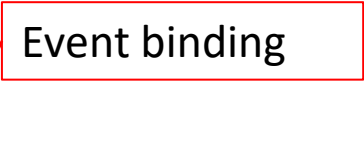
      this.errorMsg="An error occured while logging in,
      please try again later";
    }
}

```

Update the view to include error message

- Add following in: [login.component.html](#)

Update: inputs as follows

- `<input type="text" class="form-control" id="username" [(ngModel)]="username">`

- `<input type="password" class="form-control" id="password" [(ngModel)]="password"/>`
- `<div class="error">{{errorMsg}}</div>`
- `<button type="submit" class="btn btn-primary" (click)="login()">Submit</button>`


Remember intrapolation binding : `{{errorMsg}}`, event binding `(click)=`

NB> `errorMsg` comes from `login.component.ts`

Implementing the taskview page (DIY)

- Search for form on bootstrap and modify
- Update
- Search for table on bootstrap and modify
 - One form (login form and edit it)
 - One table (edit it)
- For the rest of the code ref to code on clickup



Faculty of Engineering,
Built Environment and
Information Technology
Fakulteit Ingenieurswese, Bou-omgewing en
Inligtingtegnologie / Lefapha la Boetšenere,
Tikologo ya Kago le Theknolotši ya Tshedimošo