

Angular reminder

Add FormsModule and HttpClientModule in app.module.ts → imports.

```
import {HttpClientModule} from '@angular/common/http';
```

Where HttpClientModule is (VS Code doesn't know on it's own)

```
<a routerLink="path">Anything</a>
```

Routing tag in html of component:

```
<router-outlet></router-outlet>
```

Add routes in app-routing.module.ts:

```
const routes: Routes = [  
  {path: "about", component: AboutComponent}  
];
```

Variable from component.ts:

```
{{variableName}}
```

Variable which will be given in html tag:

In specific class add:

```
@Input() mybook: Book = new Book()
```

Usage:

```
<app-bookdetails [mybook]="book"></app-bookdetails>
```

Model for Service and more:

```
export class Book{  
  name: string = "";  
  writer: string = "";  
  pages: number = 0;  
}
```

Service class (ng c s imeServisa)

```
export class BookService {  
  
  constructor() { }  
  
  getAllBooks(): Book[]{  
    return [  
      {name: "Book1", writer: "zika", pages: 123},  
      {name: "Book2", writer: "pera", pages: 32}  
    ]  
  }  
  
  findBooks(searchParam: string): Book[]{  
    return this.getAllBooks().filter(book=>  
      book.name.includes(searchParam))  
  }  
}
```

Component class (with onInit) (ng c c imeKomponente):

```
export class WritersComponent implements OnInit {

  constructor(private writerService: WriterService){

  }

  ngOnInit(): void {
    this.allWriters = this.writerService.getAllWriters()
  }
  allWriters: Writer[] = []

  sortByAwards(){
    this.allWriters = this.writerService.sortByAwards()
  }

  sortByBirth(){
    this.allWriters = this.writerService.sortByBirth()
  }
}
```

Angular for loop:

```
<div *ngFor="let book of allBooks">
  <app-bookdetails [mybook]="book"></app-bookdetails>
</div>
```

Angular if statement:

```
<div *ngIf="allWriters.length>0; else noWriters">
  Show if True
</div>
<ng-template #noWriters>
  No writers for you
</ng-template>
```

Add css class if condition is true:

```
<tr *ngFor="let writer of allWriters" [class.red]="writer.awards>3">
  <td>{{writer.anything}}</td>...
</tr>
```

Ng Model(searchParam is in component class(ime.component.ts)):

```
<input type="text" name="param" [(ngModel)]="searchParam">
```

Button and click method(search Is a method in component class(ime.component.ts)):

```
<button (click)="search()">Search</button>
```

How to navigate from component class to another page(inject router in constructor first):

```
pushToPage(){
  this.router.navigate(["/register"])
}
```

Http Request:

1.Add in Service:

```
export class UserService {

  constructor(private http: HttpClient) { }

  login(username: string, password: string){
    const data={
      username: username,
      password: password
    }
    return this.http.post<User>("http://localhost:4000/users/login", data)
  }

  register(user: User){
    return this.http.post<Message>("http://localhost:4000/users/register",
      user)
  }
}
```

2.Add in component class(ime.component.ts) – handle the response):

```
export class LoginComponent {

  constructor(private servis: UserService){}

  username: string = ""
  password: string = ""

  login(){
    this.servis.login(this.username, this.password).subscribe(
      data=>{
        if(data==null) alert("Nema korisnika")
        else alert("Hello " + data.firstname)
      }
    )
  }
}
```

Path variable in angular:

```
this.router.navigate(['k1', {indeksParam: indeks[0]+indeks[1]}]);
```

We read it like this:

```
let indeksParam = this.route.snapshot.paramMap.get("indeksParam");
```

In component in which we are reading parameter, these two need to be in constructor:

```
private route: ActivatedRoute, private router: Router
```

Method to be in .pipe(catchError(handleLoginError)).

Method is in class and needs to be private!!!

Method needs to return new Observable!!!

```
private handleLoginError(error: HttpResponse){
  console.log("Login Failed")
  switch(error.status){
    case 306:
      console.log("User not found")
      break;
    case 307:
      console.log("User not yet active")
      break;
    case 308:
      console.log("Password not mathcing!")
      break;
    case 304:
      console.log("Unknown error! Login Failed.")
  }
  return new Observable;
}
```