

Spring Boot with angular Rest A pi Example:-

Angular Front End step by step :-

Create angularpro Project folder and open in visual code you will see project structure like this :-

EXPLORER		...	TS toc
📄	✓ ANGULARPRO		src >
🔍	> node_modules		1
🔗	✓ src		2
🔗	✓ app		3
🔗	✓ adduser		4
🔗	# adduser.component.css		5
🔗	<> adduser.component.html		6
🔗	TS adduser.component.spec.ts		7
🔗	TS adduser.component.ts		8
🔗	✓ toaster-demo		10
🔗	# toaster-demo.component.css		11
🔗	<> toaster-demo.component.html		12
🔗	TS toaster-demo.component.sp...		13
🔗	TS toaster-demo.component.ts		14
🔗	✓ userclass		16
🔗	# userclass.component.css		17
🔗	<> userclass.component.html		18
🔗	TS userclass.component.spec.ts		19
🔗	TS userclass.component.ts		20
🔗	TS app-routing.module.ts		21
🔗	# app.component.css		22
🔗	<> app.component.html		23
🔗	TS app.component.spec.ts		24
🔗	TS app.component.ts		25
🔗	TS app.module.ts		26
🔗	TS app-routing.module.ts		27
🔗	TS userclass-service.service.spec.ts		28
🔗	TS userclass-service.service.ts		29
🔗			30

Create main `app.component.html` file:-

```
<> app.component.html ×
src > app > <> app.component.html > router-outlet
  1
  2 <!-- <app-toaster-demo></app-toaster-demo> -->
  3 <!-- <app-adduser></app-adduser> -->
  4
  5 <!-- Navigates to /adduser page on button click -->
  6 <button routerLink="/adduser">Registration1</button>
  7 <button [routerLink]="['/adduser']">Registration2</button>
  8
  9 <a class="nav-link" routerLink="/adduser">Registration3</a>
 10 <br>
 11
 12 <router-outlet></router-outlet>
```

`app.module.ts` :-

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserModuleAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModuleModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { ToastrModule } from 'ngx-toastr';
import { animation } from '@angular/animations';
import { ToasterDemoComponent } from './toaster-demo/toaster-demo.component';
import { AdduserComponent } from './adduser/adduser.component';
import { HttpClientModule } from '@angular/common/http';
import { UserclassComponent } from './userclass/userclass.component';

@NgModule({
  declarations: [
```

```
    AppComponent,  
    ToasterDemoComponent,  
    AdduserComponent,  
    UserclassComponent  
  ],  
  imports: [  
    BrowserModule,  
    AppRoutingModule,  
    FormsModule,  
    ReactiveFormsModule,  
    BrowserModuleAnimationsModule,  
    ToastrModule.forRoot(),  
    HttpClientModule  
  ],  
  providers: [],  
  bootstrap: [AppComponent]  
})  
export class AppModule { }
```

TS app.component.ts X

src > app > TS app.component.ts > ...

```
1  import { Component } from '@angular/core';
2  
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'angularpro';
10 }
11
```

TS app-routing.module.ts X

src > app > TS app-routing.module.ts > ...

```
1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3  import { AdduserComponent } from './adduser/adduser.component';
4  import { UserclassComponent } from './userclass/userclass.component';
5
6  const routes: Routes = [
7    {path:"users",component:UserclassComponent},
8    {path:"adduser",component:AdduserComponent},
9  ];
10
11 @NgModule({
12   imports: [RouterModule.forRoot(routes)],
13   exports: [RouterModule]
14 })
15 export class AppRoutingModule { }
16
```

Now create component `adduser` :-

`adduser.component.html` file:-

```
<div class="form-group">
  <label>Enter Username</label>
  <input placeholder="Enter Username" [(ngModel)]="u.username" class="form-
control"/>
</div>
<div class="form-group">
  <label>Enter Email</label>
  <input placeholder="Enter Email" [(ngModel)]="u.email" class="form-control"/>
</div>
<div class="form-group">
  <label>Enter Password</label>
  <input type="password" placeholder="Enter Password" [(ngModel)]="u.password"
class="form-control"/>
</div>
<div class="form-group">
  <button class="btn btn-primary" (click)="addUser()">Register</button>
</div>
```

`adduser.component.ts` file :-

```
import { Component, OnInit } from '@angular/core';
import { UserclassServiceService } from '../userclass-service.service';
import { UserClass } from '../userclass';
import { Router } from '@angular/router';
import { ToastrService } from 'ngx-toastr';

@Component({
  selector: 'app-adduser',
  templateUrl: './adduser.component.html',
  styleUrls: ['./adduser.component.css']
})
export class AdduserComponent implements OnInit {
  u=new UserClass();

  constructor( private useClassService:UserclassServiceService,private
toastr:ToastrService,
  private router:Router )
  {}

  addUser()
```

```

{
  this.u.role="user";
  this.userClassService.addUser(this.u).subscribe((result)=>
  {
    this.toastr.success("User Added Successfully!!!");
    this.router.navigate(["/users"]);
  });
}

ngOnInit(): void {
}
}

```

Now create **userclass component**:-

userclass.component.html file :-

```

<h2>UserClass Details!</h2>
<table class="table table-bordered">
  <tr>
    <th>User Id</th>
    <th>UserName</th>
    <th>Email</th>
    <th>Password</th>
    <th>Role</th>
  </tr>
  <tr *ngFor="let user of users">
    <td>{{user.userid}}</td>
    <td>{{user.username}}</td>
    <td>{{user.email}}</td>
    <td>{{user.password}}</td>
    <td *ngIf="user.role">{{user.role}}</td>
    <td *ngIf="!user.role">No Role</td>
    <td>
      <button class="btn btn-warning" (click)="edit(user)">Edit</button>
      <button class="btn btn-danger" (click)="delete(user)">Delete</button>
    </td>
  </tr>
</table>

```

```

<div *ngIf="selectedUser">
  <div class="form-group">
    <label hidden>Enter Userid</label>
    <input placeholder="Enter UserId" hidden
[(ngModel)]="selectedUser.userid" class="form-control"/>
  </div>
  <div class="form-group">
    <label>Enter Username</label>
    <input placeholder="Enter Username" [(ngModel)]="selectedUser.username"
class="form-control"/>
  </div>
  <div class="form-group">
    <label>Enter Email</label>
    <input placeholder="Enter Email" [(ngModel)]="selectedUser.email"
class="form-control"/>
  </div>
  <div class="form-group">
    <label>Enter Password</label>
    <input placeholder="Enter Password" [(ngModel)]="selectedUser.password"
class="form-control"/>
  </div>
  <div class="form-group">
    <button (click)="updateUser()" class="btn btn-warning">Update
User</button>
    <button (click)="cancel()" class="btn btn-primary">Cancel</button>
  </div>
</div>

```

userclass.component.ts file :-

```

import { Component, OnInit } from '@angular/core';
import { Userclass } from '../userclass';
import { UserclassServiceService } from '../userclass-service.service';
import { ToastrService } from 'ngx-toastr';
import { Router } from '@angular/router';

@Component({
  selector: 'app-userclass',
  templateUrl: './userclass.component.html',
  styleUrls: ['./userclass.component.css']
})
export class UserclassComponent implements OnInit {

```



```

users:UserClass[]=[];
selectedUser:UserClass | undefined;

constructor(private useClassService:UserclassServiceService,
private toastr:ToastrService,
private router:Router){}

getUserDetails()
{
  this.userClassService.getUserDetails().subscribe(result=>
  {
    this.users=result
  },
  (err: { message: string | undefined; })=>
  {
    this.toastr.error(err.message);
  }
  );
}

edit(u:UserClass)
{
  this.selectedUser=u;
}

cancel()
{
  this.selectedUser=undefined;
}

updateUser()
{
  this.userClassService.updateUser(this.selectedUser).subscribe((result)=>
  {
    this.toastr.success("User Updated successfully!!!");
  });
}

delete(u:UserClass)
{
  this.userClassService.deleteUser(u).subscribe((result)=>
  {
    this.toastr.success("User Deleted successfully!!!");
    this.getUserDetails();
  },

```

```
(err)=>
{
  this.toastr.error(err.message);
});
}

ngOnInit(): void {
  this.getUserDetails();
}
}
```

Now create `userclass.ts` file :-

```
TS userclass.ts X
src > app > TS userclass.ts > UClass
1  export class UClass
2  {
3      userid:number | undefined;
4      username:string | undefined;
5      email:string | undefined;
6      password:string | undefined;
7      role:string | undefined;
8  }
```

now create `userclassserviceservice`

`userclass-service.service.ts` file code:-

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { UserClass } from './userclass';

@Injectable({
  providedIn: 'root'
})
export class UserclassServiceService {

  constructor(private http:HttpClient) { }

  getUserDetails()
  {
    return this.http.get<UserClass[]>("http://localhost:8080/user/");
  }
  addUser(u:UserClass)
  {
    return this.http.post("http://localhost:8080/user/add",u);
  }
  updateUser(u:UserClass | undefined)
  {
    return this.http.put("http://localhost:8080/user/update",u);
  }
  deleteUser(u:UserClass)
  {
    return this.http.delete("http://localhost:8080/user/delete/"+u.userid)
  }
}
```

Create toaster-demo component:-

```
<> toaster-demo.component.html X
src > app > toaster-demo > <> toaster-demo.component.html > button
1   <h1>Angular Toastr Notifications Example</h1>
2
3   <button (click)="showToasterSuccess()">
4     |   Success Toaster
5   </button>
6
7   <button (click)="showToasterError()">
8     |   Error Toaster
9   </button>
10
11  <button (click)="showToasterInfo()">
12    |   Info Toaster
13  </button>
14
15  <button (click)="showToasterWarning()">
16    |   Warning Toaster
17  </button>
```

Write code for `toaster-demo.component.ts` file code :-

```
import { Component } from '@angular/core';
import { ToastrService } from 'ngx-toastr';

@Component({
  selector: 'app-toaster-demo',
  templateUrl: './toaster-demo.component.html',
  styleUrls: ['./toaster-demo.component.css']
})
export class ToasterDemoComponent {
  title = 'toaster-not';

  constructor(private toastr: ToastrService) { }
```

```
showToasterSuccess(){
  this.toastr.success("Data shown successfully !!",this.title)
}

showToasterError(){
  this.toastr.error("Something is wrong", this.title)
}

showToasterInfo(){
  this.toastr.info("This is info", this.title)
}

showToasterWarning(){
  this.toastr.warning("This is warning", this.title)
}
}
```