

TODO List Code

App.components.ts

```
import { Component } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Task } from './task';
import { User } from './user';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'projectTODO';

  constructor(private http:HttpClient)
  {
  }

  // obj: Task = new Task();
  // user : User=new User();
  // num:number=0;
  //url:string='http://localhost:8081/';
  url:string='http://65.0.99.161:8090/todo2/';
  // username:string='';
  // password:string='';
  // login()
  // {
  //   console.log('in login');
  //   let url = this.url + 'login' + this.username + 'and' +
  this.password;

  //   this.http.get(url).subscribe((data:any)=>{
  //     if(data==-1)
  //     {
  //       window.alert('something wrong');
  //     }
  //     else if(data==0)
  //     {
  //       this.num=1;
  //     }
  //     else if(data==2)
  //     {
  //       window.alert('wrogn password');
```

```

//     }
//     else if(data==1)
//     {
//         this.num=1;
//         // this.user=data.user;
//     }
//     else if(data==3)
//         window.alert('wrong')

//     });
// }

// add()
// {
//     console.log('add');
//     let url=this.url+'addNewTask/'+this.user.id;
//     this.http.post(url,this.obj.details).subscribe((data:any)=>
//     {
//         this.obj=data;
//         this.user.task.push(data);
//         console.log('data saved');
//         this.obj=new Task();
//         console.log(data);
//     });
// }

// // delete(obj:Task)
// // {
// //     let url2=this.url+'delete'+this.user.id+'and'+obj.id;
// //     this.http.get(url2).subscribe((data:any)=>
// //     {
// //         if(data==true)
// //         {
// //             let index=this.user.task.indexOf(obj);
// //             this.user.task.splice(index,1);
// //             window.alert("successfully delete");
// //         }
// //         else{
// //             window.alert("something is wrong")
// //         }
// //     })

```

```

//    // }

login:number=0;
username:string='';
password:string='';

user:Userrr=new Userrr();
task:Task=new Task();
list:Task[]=[];
length:number=0;
searchNo:number=0;
details:any;
id:any
loginMethod()
{
    let url2=this.url+'login'+this.username+'and'+this.password;
    this.http.get(url2).subscribe((data:any)=>
    {
        if(data==0)
        {
            window.alert('wrong username..!');
        }
        else if(data==1)
        {
            window.alert('wrong password..!');
        }
        else if(data==2)
        {
            window.alert('login successful');
            this.login=1;
        }

        let url2=this.url+'getOneUser'+this.username;
        this.http.get(url2).subscribe((data:any)=>
        {
            this.user=data;
        });
    });
    else
    {
        window.alert('something went wrong');
    }
});

```

```

}

logout()
{
  this.login=0;
}

addTask()
{
  let url2=this.url+'addNewTask'+this.user.id;
  this.http.post(url2,this.task.details).subscribe((data:any)=>
  {
    if(data==null)
    {
      window.alert('something went wrong');

    }
    else
    {
      this.task=data;
      this.user.tasks.push(data);
      this.task=new Task();
      console.log('in add');
    }

  });
}

delete(task:Task)
{
  let url2=this.url+'delete'+task.id+'and'+this.user.id;
  this.http.get(url2).subscribe((data:any)=>{
    if(data==true)
    {
      let index=this.user.tasks.indexOf(task);
      this.user.tasks.splice(index,1);
      window.alert('successfully delete');

    }
    else
    {
      window.alert('something wrong');
    }
  });
}

```

```

findByLength()
{
  let url2=this.url+'findByLength'+this.length;
  this.http.post(url2,this.user.tasks).subscribe((data:any)=>{
    this.searchNo=1;
    this.list=data;

  });
}
}

```

App.module.ts

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { HttpClientModule } from '@angular/common/http';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { FormsModule } from '@angular/forms';
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,HttpClientModule,FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

app.component.html

```

<body>
  <div *ngIf="login==0" class="log">
    Username : <input type="text" [(ngModel)]="username"><br><br>
    Password : <input type="text" [(ngModel)]="password"><br><br>

```

```

<button (click)="loginMethod()">Login</button>
<br><br><br>
dummy username:Admin

password: admin

</div>

<div *ngIf="login==1">
  <button (click)="logout()">Logout</button><br><br>
  welcome :{{user.name}}
  <br>
  <br>
  <table border="1">
    <tr>
      <td>Sr. No.</td>
      <td>Details</td>
      <td>Delete</td>
    </tr>
    <tr *ngFor="let task of user.tasks;let i=index">
      <td>{{i+1}}</td>
      <td>{{task.details}}</td>
      <td><button (click)="delete(task)">Delete</button></td>
    </tr>
  </table>
  <div>
    Add New Task : <input type="text" [(ngModel)]="task.details">
    <button (click)="addTask()">Add</button>
  </div>

  <br><br>

  <div>

    <table border="1" *ngIf="searchNo==1">
      <tr>
        <th>Sr. No.</th>
        <th>Details</th>

      </tr>
      <tr *ngFor="let task of list;let i=index">
        <td>{{i+1}}</td>
        <td>{{task.details}}</td>

```

```
        </tr>
      </table>
    </div>

</div>
</body>
```

Task.ts

```
export class Task {
  id:any;
  details:any;
  length:any;
}
```

User.ts

```
export class User {
  tasks: any;
  id:any;
  name:any;
  username:any;
  password:any;
}
```

MyController.java

```
package com.example.demo;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Service;
```

```
@Service
```

```
public class MyImpl implements UserService {
```

```
    @Autowired
```

```
    TaskRepository taskRepository;
```

```
    @Autowired
```

```
    UserRepo userRepo;
```

```
    @Override
```

```
    public Task addNewTask(int userId, String details) {
```

```
        try
```

```
        {
```

```
            System.out.println("In add task ");
```



```

        Task task= new Task(0,details,details.length());
        Task taskdb = taskRepository.save(task);

        User user = userRepo.findById(userId).get();

        user.getTasks().add(taskdb);

        userRepo.save(user);

        return taskdb;
    }

    catch (Exception e)
    {

        e.printStackTrace();

        return null;
    }
}

```

@Override

```

public int login(String username, String password) {

    List<User> list = userRepo.findAll();

    for (User admin : list)

```

```
{  
    if (username.equals(admin.getUsername()))  
    {  
        if (password.equals(admin.getPassword()))  
        {  
            return 2;  
        }  
        else  
            return 1;  
    }  
}  
  
return 0;  
}  
  
@Override  
  
public User getOneUser(String username) {  
    return userRepo.findByUsername(username);  
}
```

```

@Override

public boolean deleteTask(int taskId,int userId)

{

    User user=
userRepo.findById(userId).orElse(null);

    Task
task=taskRepository.findById(taskId).orElse(null);

    if(user!=null&&task!=null)

    {

        user.getTasks().removeIf(t-
>t.getId()==taskId);

        userRepo.save(user);

        return true;

    }

    else

        return false;

}
}

```

MyImpl.java

```
package com.example.demo;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service

public class MyImpl implements UserService {

    @Autowired

    TaskRepository taskRepository;

    @Autowired

    UserRepo userRepo;

    @Override

    public Task addNewTask(int userId, String details) {

        try

        {

            System.out.println("In add task ");

            Task task= new Task(0,details,details.length());
```

```

        Task taskdb = taskRepository.save(task);

        User user = userRepo.findById(userId).get();

        user.getTasks().add(taskdb);

        userRepo.save(user);

        return taskdb;
    }

    catch (Exception e)
    {

        e.printStackTrace();

        return null;
    }
}

@Override

public int login(String username, String password) {

    List<User> list = userRepo.findAll();

    for (User admin : list)
    {

```

```
        if (username.equals(admin.getUsername()))
        {
            if (password.equals(admin.getPassword()))
            {
                return 2;
            }
            else
                return 1;
        }
    }

    return 0;
}
```

@Override

```
public User getOneUser(String username) {

    return userRepo.findByUsername(username);

}
```

```

@Override

public boolean deleteTask(int taskId,int userId)

{

    User user=
userRepo.findById(userId).orElse(null);

    Task
task=taskRepository.findById(taskId).orElse(null);

    if(user!=null&&task!=null)

    {

        user.getTasks().removeIf(t-
>t.getId()==taskId);

        userRepo.save(user);

        return true;

    }

    else

        return false;

}
}

```

Task.java

```
package com.example.demo;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class Task {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    int id;

    String details;

    int length;

    public Task() {

        super();

        // TODO Auto-generated constructor stub

    }
```



```
public Task(int id, String details, int length) {  
  
    super();  
  
    this.id = id;  
  
    this.details = details;  
  
    this.length = length;  
  
}  
  
public int getId() {  
  
    return id;  
  
}  
  
public void setId(int id) {  
  
    this.id = id;  
  
}  
  
public String getDetails() {  
  
    return details;  
  
}  
  
public void setDetails(String details) {
```

```
        this.details = details;

    }

    public int getLength() {

        return length;

    }

    public void setLength(int length) {

        this.length = length;

    }

    @Override

    public String toString() {

        return "Task [id=" + id + ", details=" + details + ",
length=" + length + "];"

    }

}
```

TaskRepository.java

```
package com.example.demo;

import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface TaskRepository extends JpaRepository<Task,
Integer> {

    public List<Task> findByLength(int len);

}
```

User.java

```
package com.example.demo;

import java.util.List;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.OneToMany;

@Entity
```

```
public class User {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    int id;

    String name;

    String username;

    String password;

    @OneToMany

    List<Task> tasks;

    public User() {

        super();

        // TODO Auto-generated constructor stub

    }

    public User(int id, String name, String username, String
password, List<Task> tasks) {

        super();

        this.id = id;
```

```
        this.name = name;

        this.username = username;

        this.password = password;

        this.tasks = tasks;
    }

    public int getId() {

        return id;
    }

    public void setId(int id) {

        this.id = id;
    }

    public String getName() {

        return name;
    }

    public void setName(String name) {

        this.name = name;
```

```
}
```

```
public String getUsername() {
```

```
    return username;
```

```
}
```

```
public void setUsername(String username) {
```

```
    this.username = username;
```

```
}
```

```
public String getPassword() {
```

```
    return password;
```

```
}
```

```
public void setPassword(String password) {
```

```
    this.password = password;
```

```
}
```

```
public List<Task> getTasks() {
```

```
    return tasks;
```

```
}
```

```
public void setTasks(List<Task> tasks) {
```

```

        this.tasks = tasks;

    }

    @Override

    public String toString() {

        return "User [id=" + id + ", name=" + name + ",
username=" + username + ", password=" + password + ", tasks="
        + tasks + "]";

    }

}

```

UserRepo.java

```

package com.example.demo;

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepo extends JpaRepository<User, Integer> {

    public User findByUsername(String username);

    public int countByUsername(String username);

}

```

UserService.java

```
package com.example.demo;
```

```
public interface UserService {
```

```
    public Task addNewTask(int userId, String details);
```

```
    public int login(String username, String password);
```

```
    public User getOneUser(String username);
```

```
    public boolean deleteTask(int taskId, int userId);
```

```
}
```

Link and QR Code

<http://65.0.99.161:8090/toddo/>



