

## #TodoApp - Rest Services Project

This project is a spring boot project. In this project we have used H2 in memory db.

This project has swagger ui integrated with it. (Not attached to the domains currently)

The project structure is based on the Domain Driven Design and is divided as 2 domains user and todo (task domain).

#

### ## User domain:

- This domain consist of 3 REST service which are used to authenticate user, create new user and delete user

- Rest Endpoints for the user:

- URI: /user

- Endpoints:

- /user/{username}/{password} : GET endpoint

- Returns userId if user is found else returns 0.

- The OAuth2 authentication could be used here in

order to add security

- /user :POST endpoint

- requires USER object and adds used to the db

- returns userId once the user is added

- /user/{userId} : DELETE mapping

- void method which deletes the

user from the db

### ## ToDo Domain:

- ToDo domain: This domain is created to maintain all the tasks list entered by the user. This domain has 4 REST endpoints to add task, get all the task for user, update task and delete task

- Rest Endpoints for the user:

- URI: /todo

- Endpoints:

- /todo/{userId} : GET endpoint

- Returns all the task for that userId.

- /todo/{userId} :POST endpoint and PUT endpoint

- requires Task object and creates new task/

updates existing task for that user

- returns userId once the user is added

- /todo/{taskId} : DELETE mapping

- void method which deletes the task for the user

from the db

### ## Development server

#### ### Running the project:

- To run this project we need to use maven. Following commands could be used for starting the server.

- mvn clean install (All the libraries are downloaded and jar file is created in target folder)

- mvn spring-boot:run

#### ### Testing the endpoints:

This project uses integration test for testing the endpoints and JUNIT test for testing the services. This test are very helpful as they can be ran while doing the development without having to start the server.

## # ToDoApp - Angular Frontend

This project is the frontend ui project created in using angular cli

The project structure is again similar to DDD which USER and TODO components in it and has various sub components under them.

### ## Development server:

Run ng serve for a dev server. Navigate to <http://localhost:4200/>. The app will automatically reload if you change any of the source files.

### ## Building the project:

Run ng build to build the project. The build artifacts will be stored in the dist/ directory. Use the --prod flag for a production build.

### # Running unit tests

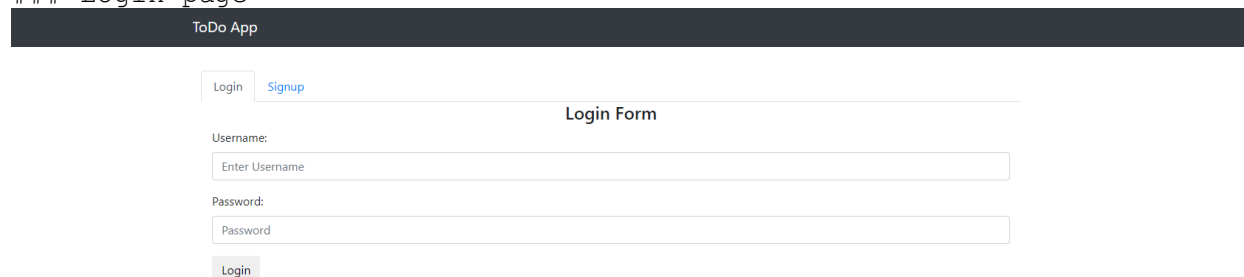
Run ng test to execute the unit tests via Karma.

### Given more time this project has a lot of scope for the improvement. Following are the task which I think I would have done given more time.

- Added more tests to cover maximum code
- Add security layer on top of the every REST endpoint to make it more secure.
- Move the above project related information to SWAGGER UI to give more clarity to consumer and also the developers.

## # Angular html pages

### ### Login page



ToDo App

Login Signup

Login Form

Username:  
Enter Username

Password:  
Password

Login

### ### Signup page

ToDo App

Login

Signup

Sign Up Form

Username:

Enter Username

Password:

Password

Login

Task screen:

ToDo App

Welcome user1Logout

Add Task

Status	Task	Description	Last Updated	Action
--------	------	-------------	--------------	--------

Creating Task:

ToDo App

Welcome user1Logout

Add Task

Create Task

Task

Description

Is Completed ☐

Submit

After task creation:

[Add Task](#)

Status	Task	Description	Last Updated	Action
<input checked="" type="checkbox"/>	Wash clothes	Wash clothes on Monday	2021-08-09T07:40:06.288+00:00	<a href="#">Edit</a> <a href="#">Remove</a>

## Task screen after Update

[Add Task](#)

Status	Task	Description	Last Updated	Action
<input type="checkbox"/>	Wash clothes	Wash clothes on Tuesday	2021-08-09T07:59:44.911+00:00	<a href="#">Edit</a> <a href="#">Remove</a>

## Task screen after Delete

[Add Task](#)

Status	Task	Description	Last Updated	Action
--------	------	-------------	--------------	--------