TODO APP-

1) Requirements-

User can sign in using unique login and password securely (this can be hard coded to a default user list, at list one user e.g. with username: test, password: pwd123)
User can view her/his task list
User can add/remove task
All changes can be persistent to allow view them in next sign in by the same user
Each task should display the date of last updates and description
User can check/uncheck any task on their list
Consider performance

2) technologies Used

- in memory H2 db,
- Maven
- spring mvc,
- JPA,
- spring boot,
- spring security,
- html.

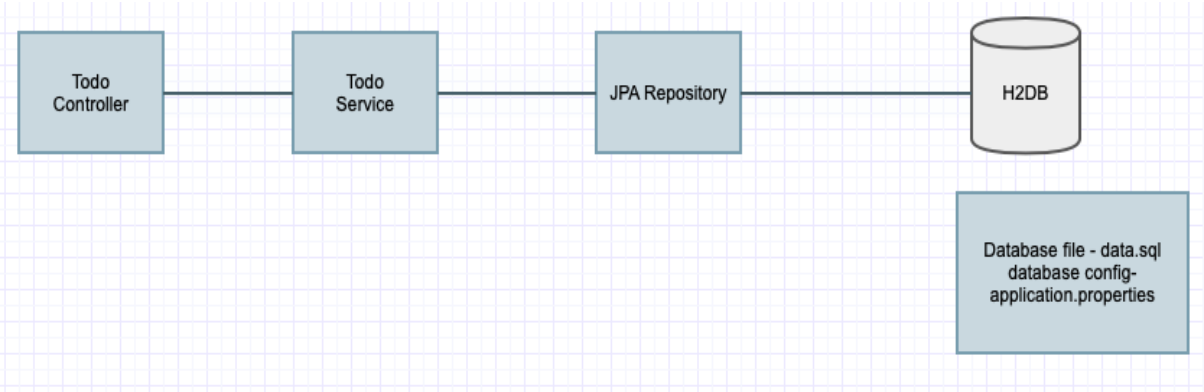
3) Approaches Considered-
Approach 1 : Rest Based- every thing will be fetched using rest API.
pros -clearner approach
cons-extra LOC for every API, performance issue for auto updation.

Approach 2 : Form action based
pros- simpler
cons:- performance issue for auto updation.

Approach 3: Using webSocket
pros - elegant solution, subscription based.
cons- complex, lower performnace.

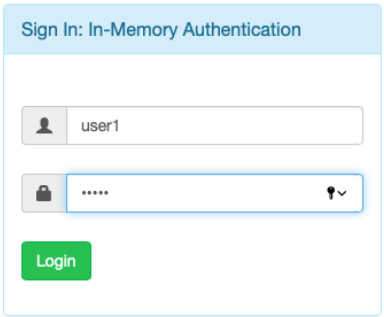
Approach taken- 2 (form based) easy to implement with a UI compared to other two approaches.

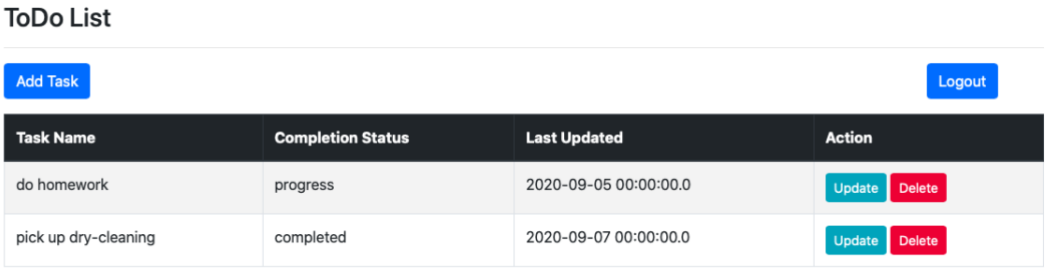
4) Architecture : It is a maven based Spring boot+spring mvc application with the following architecture.



Application Overview:

When the application is loaded, the user is asked for login details.(which are mentioned below).Upon successful login the user can see his/her tasks list. These tasks can be updated as completed or still in progress with the update button. The update also lets you to change the task description. To delete a task, the delete button is used and to add a new task add button can be used. The user entries are maintained when the user logs in another time.





5) Request Mappings-

- Login-username/password.
- fetchTasksyUsername – it lists all tasks for a particular user.
- updateTasks – task can be updated to 'completed' or description can be changed.
- Add NewTasks – user can add a new task
- Delete Task – you can delete a task
- Logout

6) Database Table-
ToDo _List:
Username, Item Id (primary key) auto incremental, Item DESCRIPTION, Date created, Date Edited, Status

7)Security: login security has been applied via spring security.

Username password combinations that are hard coded :

- Username – user1
- Password -user1

- Username-admin
- Password -admin

Url to run the war file :
Localhost:8080/warfilename
Preferred tomcat version 8 for deployment.

Anything extra you would have done give more time ?

- If I had more time, I would have liked to work more on validations and test cases.
- And, Observer pattern using websocket to update tasks for same user logged in on different tabs
- A sepe rate thread for accessing DB, so that main thread post events to DB thread