

## Practice Task 1

Create a Spring Boot project named practice-test.

Include the following dependencies:

- Web
- Validation
- Thymeleaf
- Devtools
- Redis
- Redis-Client
- JSON-P

Create an empty github repo and link you project to github.

Create the relevant folders as follows to organise your code:

- Controller
- RestController
- Configuration
- Model
- Service
- Repo
- Utils

Configure your application runs on port 3000.

## Practice Task 2

Using the CommandLineRunner, load the following data found in todo.txt as a List or Map object to Redis with the help of JSON-P library.

## Practice Task 3

Create the following model with Spring Validation.

Todo

Attribute	Data Type	Validation	Remarks
id	String	Required Maximum length = 50	ID should be generated using UUID
name	String	Required Min length = 10 Max length = 50	
description	String	Max length = 255	
Due date	Date	Date must be equals to or	Date should be stored as epochmilliseconds in Redis.

		greater than today	
priority	String		Low/medium/hight
Status	String		Pending/Started/Progress/Completed
createdAt	Date		Date of creation (Date should be stored as ephochmilliseconds in Redis.)
updatedAt	Date		Updated Date each time record is updated (Date should be stored as ephochmilliseconds in Redis.)

#### Practice Task 4

Create a page listing.html to display the list of ToDo tasks loaded in Practice Task 2.  
The page should allow you to filter records based on the Status:  
Pending/Started/Progress/Completed.

#### Practice Task 5

Create a page add.html to allow creation of new ToDo task using the Model created in Practice Task 3.  
The page should allow you to create a new ToDo record and put it to the Redis List or Map in Practice Task 2.

#### Practice Task 6

Create a login.html page.  
The page should allow user to enter their fullname and age.  
On submission, the fullname and age information should be stored as a Http session.  
User will be redirected to the listing.html page created in Practice Task 4.  
Modify the listing.html page to only display if Httpsession are present and valid.  
If Httpsession are not present or session invalidated, redirect to a custom error page refused.html. (You will define the content of refused.html page.)  
Refused.html page should have a link button to direct user back to login.html page.

#### Practice Task 7

Modify your listing.html page to include:

- Deletion of Todo record
- Update of Todo record

## Practice Task 8

On login, put in addition logic as follows:

- If user age is below 10, redirect to a underage.html page. (underage.html page should have a link button to direct user back to login.html page.)
- If user age is above 10, proceed to display listing.html page created in Practice Task 4.

## Practice Task 9

Test and run your application. Ensure that your program is working locally.

Dockerise your application using single stage docker Dockerfile.

Dockerise your application using multi-stage docker Dockerfile.

Setup your railway environment.

1. Redis Service
2. Empty Service

Deploy your solution to railway.