* Readme document for the application task manager Spring Boot.
* Basic deployment information
* Use dev profile currently as database is set up only for dev profile

java -jar target\taskmanager-0.0.1-SNAPSHOT.jar -Dspring-boot.run.profiles=dev

mvn clean package -P <profile-id>  
mvn spring-boot:run -P <profile-id>

Please use following as part of postman query ; data has been set up that way that is all tasks are under project 2.

http://localhost:8080/tasks?isProject=false&parent=2

If you are using source code then use following command.

mvn clean install spring-boot:run -Dspring-boot.run.profiles=prod

Directions on running the development REST server

o Short notes on application usage

All user stories are implemented. They are listed below.

Use postman json file for testing.

For testing using browser go to localhost:8080/tasks and other get apis.

For running tests use following command

mvn test

Notes while using postman

1. parameter is isProject please do not use isproject
2. Use this api as default value for tasks isproject under parent 2 are set up as false. <http://localhost:8080/tasks?isProject=false&parent=2>

Project notes document

* List of features that were implemented and are working

User stories implemented – All of the following stories are implemented

Note: Filter is case sensitive and it turns on when you select the checkbox

Swagger UI is available at

<http://localhost:8080/swagger-ui.html>

H2 console

Connection string: jdbc:h2:mem:testdb

Username:sa

Password:

4.0 Following User Stories are implanted

# User Story US\_01

Users should be able to access the microservice API using RESTful requests.

Acceptance criteria: The API uses URIs to identify resources. Requests return HTTP Status Codes. HTTP Methods (GET, POST, PUT, DELETE) map to microservice operations.

US\_02 Users should be able to get an individual task record.

Acceptance criteria: GET /tasks/{id} should return the task with the specified id. If the task exists, the HTTP Status Code should be 200. If the task does not exist, the HTTP Status Code should be 404.

US\_03 Users should be able to get all task records. Acceptance criteria: GET /tasks should return all tasks. If tasks exist, the HTTP Status Code should be 200. If there are no tasks, the HTTP Status Code should be 404.

US\_04 Users should be able to get all task records associated with a project.

Acceptance criteria: GET /tasks?parent={id} should return the tasks associated with the specified task. If there are associated tasks the HTTP Status Code should be 200. If the task does not exist or there are no associated tasks, the HTTP Status code should be 404.

US\_05 Users should be able to get all project (task) records.

Acceptance criteria: GET /tasks?isproject=true should return all project (task) records. If projects exist, the HTTP Status Code should be 200. If there are no projects, the HTTP Status Code should be 404.

US\_06 Users should be able to get all task records that are not project records.

Acceptance criteria: GET /tasks?isproject=false should return all task records that are not project records. If there no tasks that are not projects, the HTTP Status Code should be 200. If there are no tasks that are not projects, the HTTP Status Code should be 404.

US\_07 Users should be able to add a new task. Acceptance criteria: POST /tasks should create a new task from the request body and insert it in the database. If the task is inserted, the HTTP Status Code should be 201. If the task is inserted, the Location-Header should be set to the URI of the newly inserted task.

US\_08 Users should be able to update a task.

Acceptance criteria: PUT /tasks/{id} should update the specified task with the request body. If the task cannot be found, the task will not be inserted, and the HTTP Status code should be 404. If the task is updated, the HTTP Status Code should be 200.

US\_09 Users should be able to delete a task.

Acceptance criteria: DELETE /tasks/{id} should delete the specified task. If the task is deleted, the HTTP Status code should be 200. If the task cannot be found, the HTTP Status Code should be 404.

US\_10 Users should be able to run the microservice with dev profile only. Acceptance criteria: When the dev profile is active, the microservice must run and pass tests. If the active profile is not dev, the microservice must run but will not pass tests. In-memory database (H2) configuration must be defined in the dev profile only.

Currently if user runs – mvn clean install spring-boot:run for both profiles tests are being run , disabling tests is not added in pom file

US\_11 Users should be able to use the Angular front end to invoke the operations of the microservice.

Acceptance criteria: When a user performs operations in the front-end, modifications should be reflrected in the back-end H2 console