# Task Management App

Modules:

1. Dropwizard REST API
2. ReactJS UI app

1. REST API application setup (taskrestservice)

## Pre-requisite

1. Persistent database like MYSQL
2. Maven Build Tool
3. JDK 8 or above
4. IDE

## How to start the taskrestservice application

1. git clone https://github.com/anilkumar4412/task-app.git
2. update the database connection details in task-app/taskrestservice/local\_properties.yaml
3. create a table in database using script task-app/taskrestservice/task\_table\_ddl.sql
4. cd task-app/taskrestservice
5. Run mvn clean install to build your application
6. Start application with java -jar target/taskrestservice-1.0-SNAPSHOT.jar server local\_properties.yaml
7. To check that your application is running enter url http://localhost:8080

To create docker image and run as docker container

1. CD into project folder
2. docker build -t taskrestservice .
3. docker run -it -p 8080:8080 -p 8081:8081 -e taskrestservice taskrestservice

JUNIT Test cases (task-app/junit-test-report/jacoco/index.html)

API List

* 1. Create task:

**url**: <http://138.2.67.170:8080/task>

**method**: POST

**http-headers:** Content-Type: application/json

**request-body**:

{

"description": "Task Three",

"taskDate": "2022-01-03"

}

Response:

{"id":4,"description":"Task Three","taskDate":"2022-01-03"}

* 1. Get tasks:

**url**: <http://138.2.67.170:8080/task>

**method**: GET

**http-headers:** Content-Type: application/json

**request-body**: N/A

Response:

[{"id":1,"description":"Task One","taskDate":"2022-01-01"},{"id":3,"description":"Task Two","taskDate":"2022-01-02"},{"id":4,"description":"Task Three","taskDate":"2022-01-03"}]

* 1. Edit task:

**url**: <http://138.2.67.170:8080/task>/1

**method**: PUT

**http-headers:** Content-Type: application/json

**request-body**:

{

"id": 1,

"description": "Task One-1",

"taskDate": "2022-01-01"

}

Response:

{"id":1,"description":"Task One-1","taskDate":"2022-01-01"}

* 1. Delete task:

**url**: <http://138.2.67.170:8080/task>/1

**method**: DELETE

**http-headers:** Content-Type: application/json

**request-body**: N/A

Response:

HttpStatus: 200