It is the backend of the Walmart Task Replenisher Application which can be deployed on any cloud platform and function as an independent api.

Currently, the backend itself is deployed at url : http://18.218.226.1:8080/

Below are the specifications which needs to be taken care of before the api functionalities are utilized:

1. Assuming that MySql server is already installed in the running system, start mysql service:

mysql -u root -p password

2. Either change the sql creadentials to:

username - root

password - password

OR

Make changes in application.properties file inside path 'task-replenisher-api/src/main/resources' and update the mysql user credentials

3. Run following command on mysql prompt to create a database schema:

mysql>CREATE DATABASE task\_replenisher

4. Inside file application.properties(from step 2), change

spring.jpa.hibernate.ddl-auto=update

TO

spring.jpa.hibernate.ddl-auto=create

5. Build the project by running 'mvn clean install' at the root folder and run the jar inside the target using:

java -jar task-replenisher-api-0.0.1-SNAPSHOT.jar

6. This will create the tables in the database.

7. Run below command on sql prompt to load initial role data:

insert into role values('ADMIN'),('BUSINESS'),('INDIVIDUAL');

8. Again, inside file application.properties, revert

spring.jpa.hibernate.ddl-auto=create

TO

spring.jpa.hibernate.ddl-auto=update

9. Build and run again using step 5.

Once the jar has been executed, the service is up and running at http://localhost:8080/ and can be consumed using any REST client.

Below are the possible requests which can be made :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S. No. | Usage | Request Type | URL | RequestBody | Value constraints | Authentication | Note |
| 1 | Create admin user, who in turn is responsible for creating any system user | POST | <http://localhost:8080/admin> | {  "name":"admin", "password":"pass", "role":{"role":"ADMIN"}  } | role can only be ADMIN | N/A |  |
| 2 | Create system users. This can only be done by existing admin, hence use the admin credentials in authorization | POST | <http://localhost:8080/authorized/users> | { "name":"business2", "password":"pass", "role":{"role":"BUSINESS"}  } | role can either be ADMIN, BUSINESS or INDIVIDUAL, and is mandatory | Basic Auth username : admin password : pass | auth credentials will be same as the ones created in request 1 |
| 3 | Get list of admin and system users. To be run by admin, hence admin's credentials | GET | <http://localhost:8080/authorized/users> | N/A |  | Basic Auth username : admin password : pass | auth credentials will be same as the ones created in request 1 |
| 4 | Get tasks in the database. This can either be used by BUSINESS or INDIVIDUAL user, which are identified based on their respective credentials in the basic-authorization.  BUSINESS USERS can access all the tasks in the system, while INDIVIDUAL users can only see the tasks assigned to them | GET | <http://localhost:8080/authorized/tasks/> | N/A |  | Basic Auth username : business2 password : pass | Auth credentials of any BUSINESS or INDIVIDUAL user |
| 5 | Create task in the database and assign it to an existing user (BUSINESS OR INDIVIDUAL)  BUSINESS user can assign it to any system user, while INDIVIDUAL can only assign it to themselves. | POST | <http://localhost:8080/authorized/tasks/> | { "description": "dance", "status": "CREATED", "priority": 4 "assignedTo": { "name": "individual2" } } | Status can be either CREATED/ASSIGNED/STARTED/FINISHED  if the status is left empty, it is automatically saved as CREATED or ASSIGNED based on whether assignedTo is empty or not  priority is mandatory and can range between 1 and 10  description is mandatory | Basic Auth username : business1 password : pass | Auth credentials of any BUSINESS or INDIVIDUAL user |
| 6 | Update task using specific task id in the url  Only the updated parameter needs to be supplied in the request body. If the status of the task changes, STARTTIME and ENDTIME are updated accordingly | PUT | <http://localhost:8080/authorized/tasks/1> | { "status":"STARTED" } | Status when changes to STARTED, STARTTIME becomes current time  Status when changes to FINISHED, ENDTIME becomes current time  Status when changes to CREATED/ASSIGNED, STARTTIME and ENDTIME become null | Basic Auth username : business1 password : pass | Auth credentials of any BUSINESS or INDIVIDUAL user |