Installation Guid for Warehouse Management

Front End Application:

Prerequisite:

Angular cli version 8.3.19

Git Repository:

https://github.com/praveenkumarmidd/warehouse-management-fe.git

- Steps to run (Git):
  - ✓ npm install
  - ✓ “npm build –prod” or “ng build –prod”
  - ✓ “npm start” or “ng serve”

Application URL: http://localhost:4200/

Docker Hub Repository:

docker image - praveenkumarmiddi/warehouse-management-frontend:v1

- Step to run (docker image)

To Run only Front-End Warehouse management:

- ✓ docker pull praveenkumarmiddi/warehouse-management-frontend:v1
- ✓ docker run -p 80:80 praveenkumarmiddi/warehouse-management-frontend:v1

To Run both Front end and Back-end warehouse:

- ✓ Execute the command in the "docker-compose -f docker-compose.yml up" (project root directory)

Application URL: http://localhost:80

Back End Application:

Prerequisite:

1. Java 8
2. apache Maven 3.3.9

Tech-Stack:

1. Java 8
2. Springboot 2.5.8
3. apache Maven 3.3.9
4. H2 Database
5. Swagger - 2.9.2
6. Junit
7. Docker

Git hub Repository:

https://github.com/praveenkumarmidd/warehouse-service-backend.git

Steps to run (Git)

1. Build the warehouse Service project using "mvn clean install" in the project root directory
2. Run the application using "mvn spring-boot:run"
3. The warehouse-service is accessible via localhost:8080
4. Login credentials username - Demo and password - Demo123

Docker Hub Repository:

docker image praveenkumarmiddi/warehouse-service-backend:v1

Steps to run (Docker image):

1. To run only warehouse backend service: docker pull praveenkumarmiddi/warehouse-service-backend:v1 docker run -p 8080:8080 praveenkumarmiddi/warehouse-service-backend:v1

2. To run Frontend and Backend: "docker-compose -f docker-compose.yml up" in the project root directory

Application URL:

Swagger URL: http://localhost:8080/swagger-ui.html#/ Api Docs: http://localhost:8080/v2/api-docs

Authentication: Basic Auth Username: Demo Password: Demo123

Endpoints:

1. Create Box endpoint to create the box of the product
   URL http://localhost:8080/warehouse/v1/box/createBox

   HTTP Method: Post

   Json Request:

   { "boxName": "testBox", "boxCapacity": 1 }

2. Add Product endpoint to add product to the given box
   URL: http://localhost:8080/warehouse/v1/product/addProduct

   HTTP Method: Put

   Json Request:

   { "productName": "testProductName", "boxName": "testBox" }

3. Get Products by product name search
   URL: http://localhost:8080/warehouse/v1/product/getProducts?productName=testProductName

   HTTP Method: Get

4. Fetch boxes with minimum available capacity
   URL: http://localhost:8080/warehouse/v1/box/getAvailableBox

   HTTP Method: Get

Application Monitoring Endpoints:

- ✓ http://localhost:8080/actuator/health
- ✓ http://localhost:8080/actuator/prometheus
- ✓ http://localhost:8080/actuator/metrics