Capsule Task Manager Case Study:

Application Structure: Parent maven project(task-manager-aggregator) with two dependent modules - For building and packaging both service and ui into a single jar

task-manager-ui - Maven Spring boot project for exposing rest endpoints + MY SQL task-manager-ui - Angular CLI for building UI and connects with rest endpoints exposed.

Git Repository:

https://github.com/PremkumarFSD/FSD-TaskManager.git/

Maven Build Commands for the final artifacts:

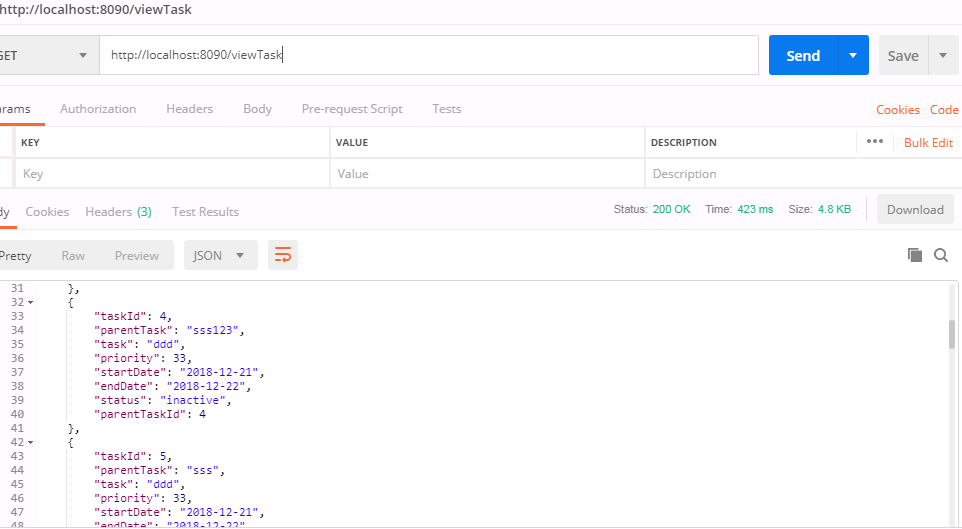
mvn clean install -e [run the command for task-manager-aggregator project which will build UI and service project and create the final jar with required resources] mvn package docker:build [run the command for task-manager-ui project which will copy the jar from target and create image in the remote docker container] Commands to run the created image and validate the image:

Connect to the remote docker machine To check whether the image is created in docker container dockerx image ls To run the created image in docker dockerx run -p 8586:8586 prem-task-manager:latest To check whether the image is running in docker[open new cmd prompt and run the cmd] dockerx ps To validate whether the application is working fine using curl command

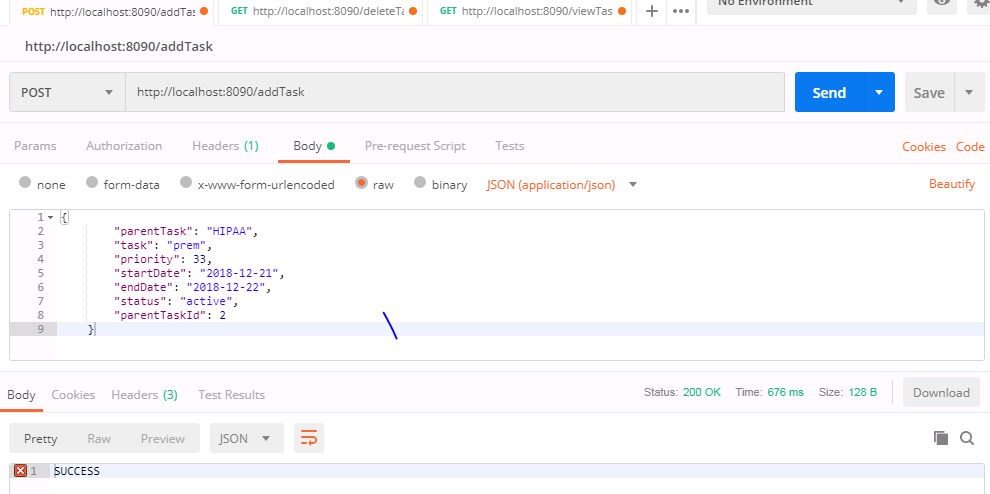
i) connect to bash shell in the container. [take container id of the image created from dockers ps] dockerx exec -it [CONTAINER\_ID] bash

ii) check whether application is working [it will return custom technical error from the service exposed since my sql is not available in docker] curl -H "Accept:/" -H "Content-Type:application/json" -X [http://localhost:8090/viewTasks](http://localhost:8085/task-manager/viewtasks) Commands for local development:

taskmanager-ui Check whether the endpoint is working fine in postman Endpoint : [http://localhost:8090/viewTasks](http://localhost:8085/task-manager/viewtasks)  Request: {} Method: GET Headers: Accept:/ Content-Type:application/json Response:



Add Task:



Open the folder src/main/web of project task-manager-ui in visual studio and then run below commands npm install npm start Hit the url [http://loalhost:4200](http://loalhost:4200/) and see whether the page is getting loaded Jenkins command:

Make sure Jenkins installed and running Configure Maven and JDK in jenkins with name maven3 and jdk1.8 Create Jenkins project with Pipeline option and configure the below information i) Github repository with credentials ii) Branch to build: \*/develop iii) Path to Jenkinsfile -> Note:

All the configurations are available in pom.xml of respective projects

Docker Image:

