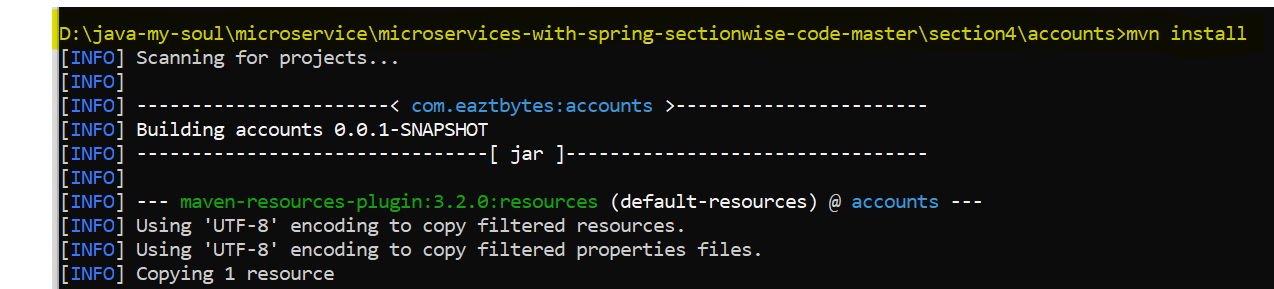
1. Sign to [www.docker.com](http://www.docker.com)
2. Download Docker Desktop
3. Access Docker Documents from <https://docs.docker.com/>
4. To Access H2 Database
   1. Ensure the following properties set to true in application.properties.

spring.h2.console.enabled=true

* 1. http://localhost:<app:port>/h2-console

Go to the command line

* Build the account,application
  + mvn clean install



* Run the account application
  + mvn spring-boot:run // type it from the folder containing pom.xml
  + OR java -jar target/accounts-0.0.1-SNAPSHOT.jar
* Test the application using rest client.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Repeat the same with other applications & test them

Stop all the services

* Your microservices are ready…

**Building Dockers Images [ Step by Steps ]**

1. Create Docker Files with the name Dockerfile with no extension under your project “Account ” workspace.

#Start with a base image containing Java runtime

FROM openjdk:11-slim as build

#Information around who maintains the image

MAINTAINER javaoursoul

# Add the application's jar to the container

COPY target/accounts-0.0.1-SNAPSHOT.jar accounts-0.0.1-SNAPSHOT.jar

#execute the application

ENTRYPOINT ["java","-jar","/accounts-0.0.1-SNAPSHOT.jar"]

1. Create Account Service Docker Image
   1. Confirm if any docker image available in your system currently. Run the command below :

**docker images**

docker build . -t nitin/accounts

docker images

docker image inspect <docker-id>

1. Start the container

docker run -p 8081:8080 nitin/accounts

docker ps // to check running instance

1. Call the rest service using restclient at <http://localhost:8081>/myAccount
2. You can run multiple instance of container using the same images…
   1. docker run -p 9081:8080 nitin/accounts

OR docker run -d -p 9081:8080 nitin/accounts [ without log ]

* 1. docker ps // to check the running instance
  2. Call the rest service using restclient at <http://localhost:9081>

Similarly, you can create container services for the other two microservices.

docker logs <container-id> // show you all logs

docker logs -f <container-id>

docker stop <container-id>

docker ps -a

docker start <cont-id>

docker container pause <cont-id>

docker container unpause <cont-id>

docker container inspect <cont-id>

docker kill <cont-id>

docker stats

docker rm <cont-id>

**Introduction to buildpacks to avoid building manually images files…**

[Cloud Native Buildpacks · Cloud Native Buildpacks](https://buildpacks.io/)

By Pivotal and Heroku

* Go to loan application pom.xml file and add following

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<configuration>

<image>

<name>nitin/${project.artifactId}</name>

</image>

</configuration>

</plugin>

</plugins>

</build>

* Go to command prompt

**mvn spring-boot:build-image**

**Pushing Images to Docker Hub**

**docker tag nitin/accounts:latest srinit16/sample-repo:accounts**

**docker push srinit16/sample-repo:accounts**

Introduction to Dockers Compose

[Overview of Docker Compose | Docker Documentation](https://docs.docker.com/compose/)

Verify if it is already installed run the below command.

docker-compose version

* Create a docker-compose.yaml file
* docker-compose up // will run containers for all images specified in yml file.
* Docker-compose stop // to stop