package com.ecomerce.webapp;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class EcomResource {

@GetMapping("/greeting")

public String greeting(@RequestParam(value = "name", defaultValue="World") String name) {

return "Today is a wonderfull day ! , " + name;

}

@GetMapping("/")

public String index() {

return "Hello, Welcome ecommerce webservice application ! ";

}

@GetMapping("/hello")

public String hello() {

return "Hello, wonderfull job!";

}

}

package com.ecomerce.webapp;

public class Employee {

public Employee() {

}

public Employee(Integer id, String firstName, String lastName, String email) {

super();

this.id = id;

this.firstName = firstName;

this.lastName = lastName;

this.email = email;

}

private Integer id;

private String firstName;

private String lastName;

private String email;

//Getters and setters

public Integer getId() {

return this.id;

}

public String getFirstName() {

return this.firstName;

}

public String getLastName() {

return this.lastName;

}

public String getEmail() {

return this.email;

}

public void setId(Integer id) {

this.id= id;

}

public void setFirstName(String firstName) {

this.firstName= firstName;

}

public void setLastName(String lastName) {

this.lastName= lastName;

}

public void setEmail(String email) {

this.email= email;

}

@Override

public String toString() {

return "Employee [id=" + id + ", firstName=" + firstName + ", lastName=" + lastName + ", email=" + email + "]";

}

}

package com.ecomerce.webapp;

import java.net.URI;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import com.ecomerce.webapp.EmployeeDAO;

import com.ecomerce.webapp.Employee;

import com.ecomerce.webapp.Employees;

@RestController

@RequestMapping(path = "/employees")

public class EmployeeController

{

@Autowired

private EmployeeDAO employeeDao;

@GetMapping(path="/", produces = "application/json")

public Employees getEmployees()

{

return employeeDao.getAllEmployees();

}

@PostMapping(path= "/", consumes = "application/json", produces = "application/json")

public ResponseEntity<Object> addEmployee(@RequestBody Employee employee)

{

Integer id = employeeDao.getAllEmployees().getEmployeeList().size() + 1;

employee.setId(id);

employeeDao.addEmployee(employee);

URI location = ServletUriComponentsBuilder.fromCurrentRequest()

.path("/{id}")

.buildAndExpand(employee.getId())

.toUri();

return ResponseEntity.created(location).build();

}

}

package com.ecomerce.webapp;

import org.springframework.stereotype.Repository;

import com.ecomerce.webapp.Employee;

import com.ecomerce.webapp.Employees;

@Repository

public class EmployeeDAO

{

private static Employees list = new Employees();

static

{

list.getEmployeeList().add(new Employee(1, "Lokesh", "Gupta", "lokeshgupta@gmail.com"));

list.getEmployeeList().add(new Employee(2, "Alex", "Kolenchiskey", "alex@gmail.com"));

list.getEmployeeList().add(new Employee(3, "David", "Kameron", "david@gmail.com"));

list.getEmployeeList().add(new Employee(3, "Nisha", "Singh", "nisha@gmail.com"));

}

public Employees getAllEmployees()

{

return list;

}

public void addEmployee(Employee employee) {

list.getEmployeeList().add(employee);

}

}

package com.ecomerce.webapp;

import java.util.ArrayList;

import java.util.List;

public class Employees

{

private List<Employee> employeeList;

public List<Employee> getEmployeeList() {

if(employeeList == null) {

employeeList = new ArrayList<>();

}

return employeeList;

}

public void setEmployeeList(List<Employee> employeeList) {

this.employeeList = employeeList;

}

}

Docker File

FROM openjdk:8-jdk

COPY ./target/ecom-webservice-0.0.1-SNAPSHOT.jar ecom-webservice-0.0.1-SNAPSHOT.jar

CMD ["java" ,"-jar","ecom-webservice-0.0.1-SNAPSHOT.jar"]

RUN echo "jenkins ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

#-> Create a image for springboot webservice

# --> Dockerfile

# - -> docker image build -t <imagetagname> .

#-> run container by using custom image

# -> docker container run --name <ecom-web-service> -p 8081:8081 -d <ecom-webservice>

#-> verify log

# -> docker container log <container-name>

pipeline {

agent any

triggers {

pollSCM('\* \* \* \* \*')

}

// Got permission denied while trying to connect to the Docker daemon socket at unix.

// sudo usermod -a -G docker jenkins

// restart jenkins server -> sudo service jenkins restart

stages {

stage('Maven Compile') {

steps {

echo '----------------- This is a compile phase ----------'

sh 'mvn clean compile'

}

}

stage('Maven Test') {

steps {

echo '----------------- This is a compile phase ----------'

sh 'mvn clean test'

}

}

stage('Maven Build') {

steps {

echo '----------------- This is a build phase ----------'

sh 'mvn clean package -DskipTests'

}

}

stage('Docker Build') {

steps {

echo '----------------- This is a build docker image phase ----------'

sh '''

docker image build -t ecom-webservice .

'''

}

}

stage('Docker Deploy') {

steps {

echo '----------------- This is a docker deploment phase ----------'

sh '''

(if [ $(docker ps -a | grep ecom-webservice | cut -d " " -f1) ]; then \

echo $(docker rm -f ecom-webservice); \

echo "---------------- successfully removed ecom-webservice ----------------"

else \

echo OK; \

fi;);

docker container run --restart always --name ecom-webservice -p 8081:8081 -d ecom-webservice

'''

}

}

}

}

Pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 <https://maven.apache.org/xsd/maven-4.0.0.xsd>">

<modelVersion>4.0.0</modelVersion>

<parent>

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

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.4.1</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<groupId>com.ecomerce.webapp</groupId>

<artifactId>ecom-webservice</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>ecom-webservice</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>11</java.version>

</properties>

<dependencies>

<dependency>

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

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

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

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>