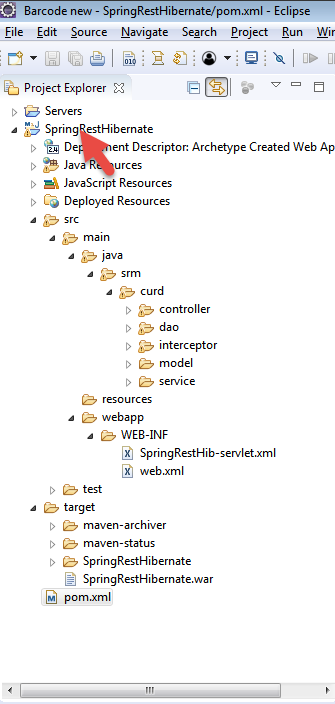
1.Create a dynamic web project With web.xml

+ convert as maven project



StudentController.java

**package** srm.curd.controller;

**import** java.util.List;

**import** org.hibernate.annotations.common.reflection.Filter;

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

**import** org.springframework.web.bind.annotation.PathVariable;

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

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

**import** org.springframework.web.bind.annotation.RequestMethod;

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

**import** org.springframework.web.bind.annotation.\*;

**import** javax.servlet.FilterChain;

**import** javax.servlet.FilterConfig;

**import** javax.servlet.ServletException;

**import** javax.servlet.ServletRequest;

**import** javax.servlet.ServletResponse;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** srm.curd.dao.StudentDAOImpl;

**import** srm.curd.model.Student;

**import** srm.curd.service.StudentServiceImpl;

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

@CrossOrigin(origins = "http://localhost:4200",maxAge = 4800, allowCredentials = "false")

@RestController

**public** **class** StudentController {

@Autowired

**private** StudentServiceImpl studentServiceImpl;

/\*\*\* Creating a new Student \*\*\*/

@RequestMapping(value = "/create", method = RequestMethod.***POST*** ) //, produces="application/json", consumes = "application/json")

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

**public** ResponseEntity<?> createStudent(@RequestBody Student student) {

studentServiceImpl.createStudent(student);

**return** ResponseEntity.*ok*().body("saved :" + student.toString());

//retur "{'status':'1'}";

// return new ResponseEntity<Student>(student, HttpStatus.OK);

}

/\*\*\* Retrieve a single Student \*\*\*/

@RequestMapping(value = "/student/{id}", method = RequestMethod.***GET***, produces = "application/json") //, produces = "application/json",

//Student

**public** Student getStudentById(@PathVariable("id") **long** id) {

Student student = studentServiceImpl.getStudentById(id);

**return** student;

//return new ResponseEntity<Student>(student, HttpStatus.OK);

}

/\*\*\* Retrieve all Students \*\*\*/

@RequestMapping(value = "/students", produces = "application/json", method = RequestMethod.***GET***)

**public** List getAllStudents() {

List studentList = studentServiceImpl.getAllStudents();

**return** studentList;

}

/\*\*\* Update a Student \*\*\*/

@RequestMapping(value = "/update", method = RequestMethod.***PUT***) //, produces = "application/json", consumes = "application/json")

**public** ResponseEntity<?> updateStudent(@RequestBody Student student) {

studentServiceImpl.updateStudent(student);

**return** ResponseEntity.*ok*().body("Updated :" + student.toString());

}

/\*\*\* Delete a Student \*\*\*/

@RequestMapping(value = "/delete/{id}", method = RequestMethod.***DELETE***) //, produces = "application/json")

**public** ResponseEntity<?> deleteStudent(@PathVariable("id") **long** id) {

studentServiceImpl.deleteStudent(id);

**return** ResponseEntity.*ok*().body("User has been deleted successfully.");

}

}

StudentDAO

package srm.curd.dao;

import java.util.List;

import srm.curd.model.Student;

public interface StudentDAO {

public void createStudent(Student student);

public Student getStudentById(long id);

public List<Student> getAllStudents();

public void updateStudent(Student student);

public void deleteStudent(long id);

}

StudentDAOImpl

package srm.curd.dao;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import srm.curd.model.Student;

@Service

@Transactional(propagation = Propagation.REQUIRED)

public class StudentDAOImpl implements StudentDAO {

@PersistenceContext

private EntityManager entityManager;

@Override

public void createStudent(Student student)

{

entityManager.persist(student);

System.out.println("Saved successfully");

}

@Override

public Student getStudentById(long id)

{

System.out.println("get student");

return entityManager.find(Student.class,id);

}

@Override

public List<Student> getAllStudents()

{

System.out.println("all students");

return entityManager.createQuery("select s from Student s").getResultList();

}

@Override

public void updateStudent(Student student)

{

entityManager.merge(student);

}

@Override

public void deleteStudent(long id)

{

Student s = entityManager.find(Student.class,id);

entityManager.remove(s);

}

}

Student

package srm.curd.model;

import java.io.Serializable;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

import javax.xml.bind.annotation.XmlAccessType;

import javax.xml.bind.annotation.XmlAccessorType;

import javax.xml.bind.annotation.XmlRootElement;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

@Entity

@Table(name = "student")

@JsonIgnoreProperties({"hibernateLazyInitializer","handler"})

@XmlAccessorType(XmlAccessType.FIELD)

@XmlRootElement(name = "Student")

public class Student implements Serializable{

private static final long serialVersionUID = 1L;

@Id

@Column

private long id;

@Column

private String name;

@Column

private int age;

@Column

private String dob;

@Column

private String degree;

@Column

private String department;

@Column

private char grade;

public Student() {

super();

}

public Student(int id, String name, int age,String dob,String degree,String department,char grade) {

super();

this.id = id;

this.name = name;

this.age =age;

this.dob = dob;

this.degree = degree;

this.department = department;

this.grade = grade;

}

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String getDob() {

return dob;

}

public void setDob(String dob) {

this.dob = dob;

}

public String getDegree() {

return degree;

}

public void setDegree(String degree) {

this.degree = degree;

}

public String getDepartment() {

return department;

}

public void setDepartment(String department) {

this.department = department;

}

public char getGrade() {

return grade;

}

public void setGrade(char grade) {

this.grade = grade;

}

@Override

public String toString() {

return "Student [id=" + id + ", name=" + name + ", age=" + age + ", dob=" + dob + ", degree=" + degree + ", department=" + department + ", grade=" + grade + "]";

}

}

StudentService

package srm.curd.service;

import java.util.List;

import srm.curd.model.Student;

public interface StudentService

{ public void createStudent(Student student);

public Student getStudentById(long id);

public List<Student> getAllStudents();

public void updateStudent(Student student);

public void deleteStudent(long id);

}

StudentServiceImpl

package srm.curd.service;

import java.util.List;

import javax.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import srm.curd.dao.StudentDAO;

import srm.curd.model.Student;

@Service

@Transactional

public class StudentServiceImpl implements StudentService{

@Autowired

private StudentDAO studentDao;

@Override

public void createStudent(Student student) {

studentDao.createStudent(student);

}

@Transactional

@Override

public Student getStudentById(long id) {

return studentDao.getStudentById(id);

}

@Transactional

@Override

public List<Student> getAllStudents() {

return studentDao.getAllStudents();

}

@Transactional

@Override

public void updateStudent(Student student) {studentDao.updateStudent(student);

}

@Transactional

@Override

public void deleteStudent(long id)

{

studentDao.deleteStudent(id);

}

}

StudentAfterAdvice

package srm.curd.interceptor;

import java.util.Date;

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Aspect;

@Aspect

public class StudentAfterAdvice {

@After("execution(\* \*.\*(..)) && within(srm.curd.dao..\*)")

public void logBefore(JoinPoint joinPoint) {

long end = System.currentTimeMillis();

System.out.println("current time "+end);

System.out.println("logAfter() is running!");

System.out.println("student : " + joinPoint.getSignature().getName());

System.out.println("\*\*\*\*\*\*");

}

}

StudentAroundAdvice

package srm.curd.interceptor;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.Around;

import org.aspectj.lang.annotation.Aspect;

import org.springframework.stereotype.Component;

@Aspect

@Component

public class StudentAroundAdvice {

/\*@Around("execution(\* \*.\*(..)) && within(srm.curd.dao..\*)")

public Object profile(ProceedingJoinPoint pjp) throws Throwable {

long start = System.currentTimeMillis();

System.out.println("Going to call the method.");

Object output = pjp.proceed();

System.out.println("Method execution completed.");

long elapsedTime = System.currentTimeMillis() - start;

System.out.println("Method execution time: " + elapsedTime + " milliseconds.");

return output;

}\*/

@Around("execution(\* \*.\*(..)) && within(srm.curd.dao..\*)")

public Object profile(ProceedingJoinPoint pjp) throws Throwable {

long start = System.currentTimeMillis();

System.out.println("Going to call the method.");

Object output = pjp.proceed();

System.out.println("Method execution completed.");

long elapsedTime = System.currentTimeMillis() - start;

System.out.println("Method execution time: " + elapsedTime + " milliseconds.");

return output;

}

}

StudentBeforeAdvice

package srm.curd.interceptor;

import java.util.Date;

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

@Aspect

public class StudentBeforeAdvice {

@Before("execution(\* \*.\*(..)) && within(srm.curd.dao..\*)")

public void logBefore(JoinPoint joinPoint) {

long start = System.currentTimeMillis();

System.out.println("current time "+start);

System.out.println("logBefore() is running!");

System.out.println("student : " + joinPoint.getSignature().getName());

System.out.println("\*\*\*\*\*\*");

}

}

SpringRestHib-servlet.xml

<beans xmlns="http://www.springframework.org/schema/beans"

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

xmlns:mvc="http://www.springframework.org/schema/mvc" xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/tx

http://www.springframework.org/schema/tx/spring-tx.xsd">

<context:component-scan base-package="srm.curd" />

<mvc:annotation-driven />

<!-- <aop:aspectj-autoproxy />

<bean id="student" class="srm.curd.service.StudentServiceImpl"></bean> -->

<!-- Aspect -->

<!-- <bean id="StudentBeforeAspectj" class="srm.curd.interceptor.StudentBeforeAdvice" />

<bean id="StudentAfterAspectj" class="srm.curd.interceptor.StudentAfterAdvice"></bean> -->

<!-- <bean id="StudentAroundAspectj" class="srm.curd.interceptor.StudentAroundAdvice"></bean> -->

<!-- Database Configurations -->

<!-- <bean id="dataSource"

class="org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name="driverClassName" value="org.postgresql.Driver" />

<property name="url" value="jdbc:postgresql://localhost:5432/student" />

<property name="username" value="postgres" />

<property name="password" value="root" />

</bean> -->

<bean id="dataSource"

class="org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name="driverClassName" value="com.mysql.jdbc.Driver" />

<property name="url" value="jdbc:mysql://localhost:3306/concretepage" />

<property name="username" value="root" />

<property name="password" value="root" />

</bean>

<!-- Hiberante Sesssion Factory Configurations -->

<!-- <bean id="sessionFactory" class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">

<property name="dataSource" ref="dataSource" /> <property name="annotatedClasses">

<list> <value>com.javainterviewpoint.model.Student</value> </list> </property>

<property name="hibernateProperties"> <props> <prop key="hibernate.dialect">org.hibernate.dialect.Oracle10gDialect</prop>

<prop key="hibernate.show\_sql">true</prop> <prop key="hibernate.hbm2ddl.auto">update</prop>

</props> </property> </bean> -->

<!-- Hiberante Entity Manager Configurations -->

<bean id="entityManagerFactory"

class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">

<property name="dataSource" ref="dataSource" />

<property name="packagesToScan"

value="srm.curd.model,srm.curd.controller,srm.curd.dao,srm.curd.service" />

<property name="jpaVendorAdapter">

<bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter" />

</property>

<property name="jpaProperties">

<props>

<!-- org.hibernate.dialect.MySQLDialect

org.hibernate.dialect.PostgreSQLDialect

-->

<prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>

<prop key="hibernate.show\_sql">true</prop>

<prop key="hibernate.hbm2ddl.auto">update</prop>

</props>

</property>

</bean>

<bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">

<property name="entityManagerFactory" ref="entityManagerFactory" />

</bean>

<tx:annotation-driven proxy-target-class="true"

transaction-manager="transactionManager" />

<mvc:cors>

<mvc:mapping path="/\*\*"

allowed-origins="http://localhost:4200"

allowed-methods="POST, GET, PUT, OPTIONS, DELETE"

allowed-headers="X-Auth-Token, Content-Type"

exposed-headers="custom-header1, custom-header2"

allow-credentials="false"

max-age="4800" />

</mvc:cors>

</beans>

Web.xml

<web-app xmlns=*"http://java.sun.com/xml/ns/javaee"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* version=*"2.5"*>

<display-name>Archetype Created Web Application</display-name>

<servlet>

<servlet-name>SpringRestHib</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/SpringRestHib-servlet.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>SpringRestHib</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

Pom.xml

<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 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>WEF</groupId>

<artifactId>SpringRestHibernate</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>SpringRestHibernate Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.0.6.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/cglib/cglib -->

<dependency>

<groupId>cglib</groupId>

<artifactId>cglib</artifactId>

<version>3.2.6</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.3.1.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.9.5</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet.jsp/javax.servlet.jsp-api -->

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>javax.servlet.jsp-api</artifactId>

<version>2.3.2-b02</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.0.6.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-tx -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>5.0.6.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/postgresql/postgresql -->

<!-- <dependency>

<groupId>postgresql</groupId>

<artifactId>postgresql</artifactId>

<version>9.1-901.jdbc4</version>

</dependency> -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.10</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.0.6.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjrt -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>1.9.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.1</version>

</dependency>

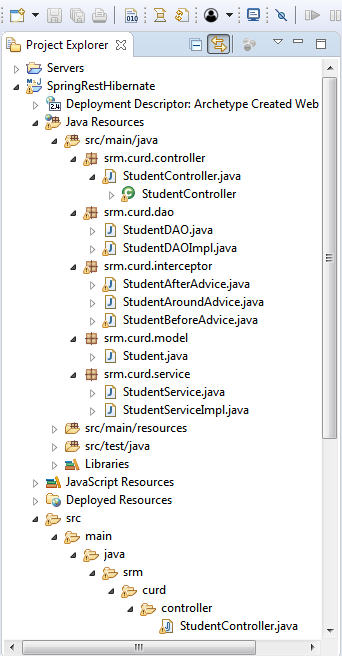
</dependencies>

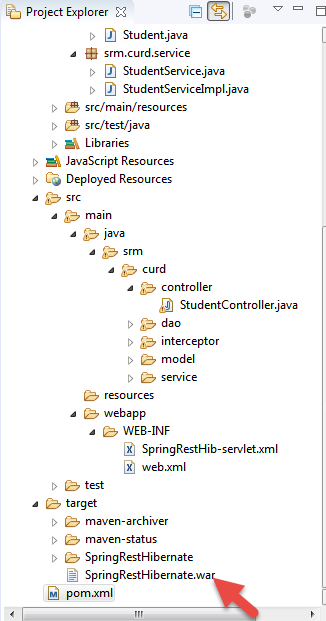
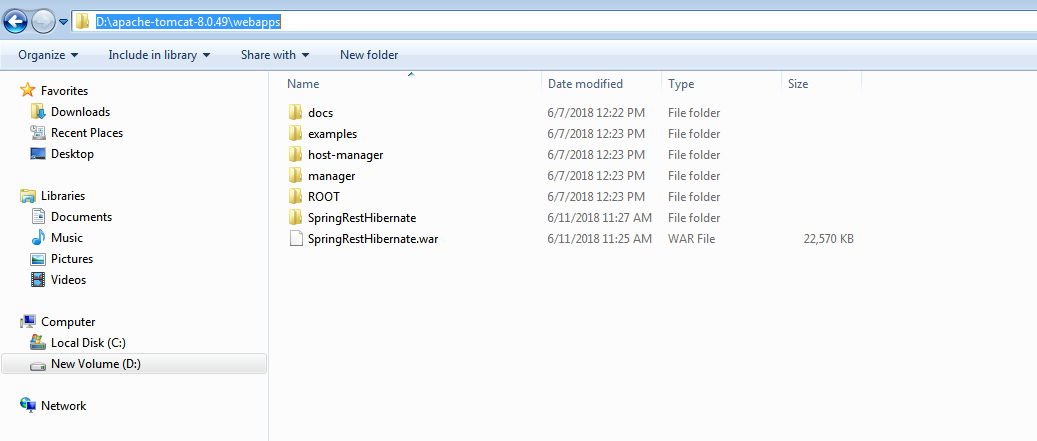
<build>

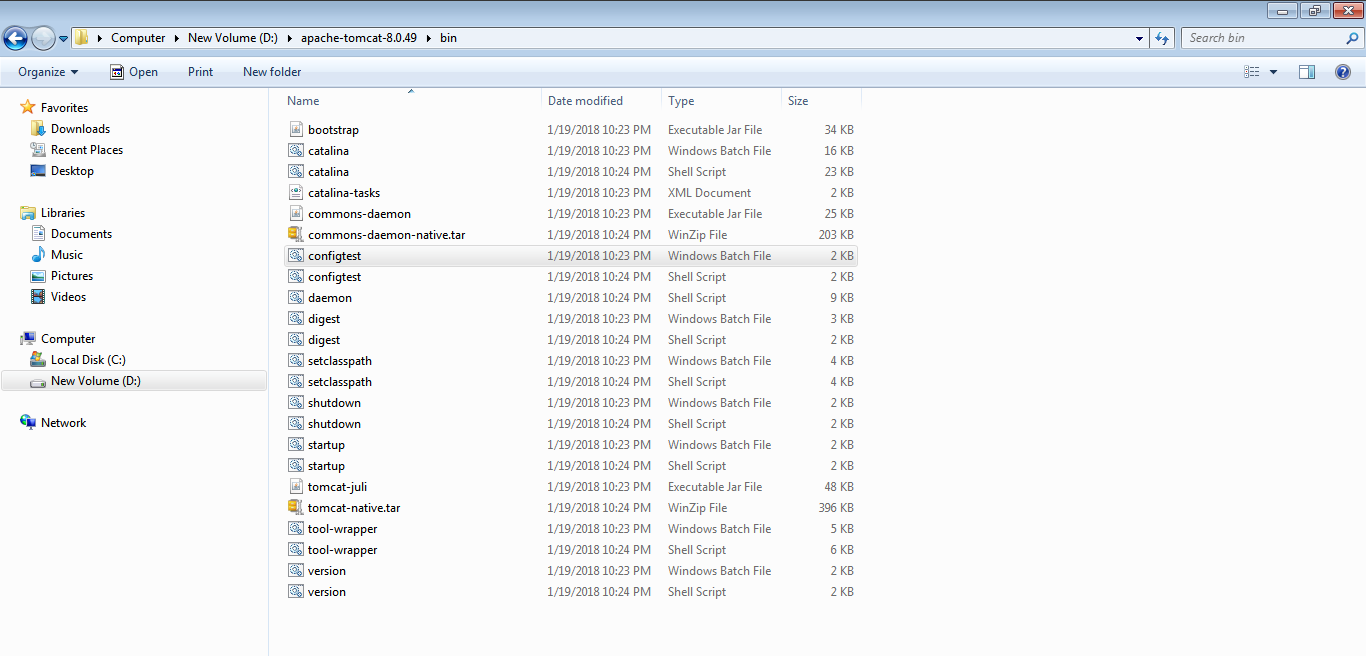
<finalName>SpringRestHibernate</finalName>

</build>

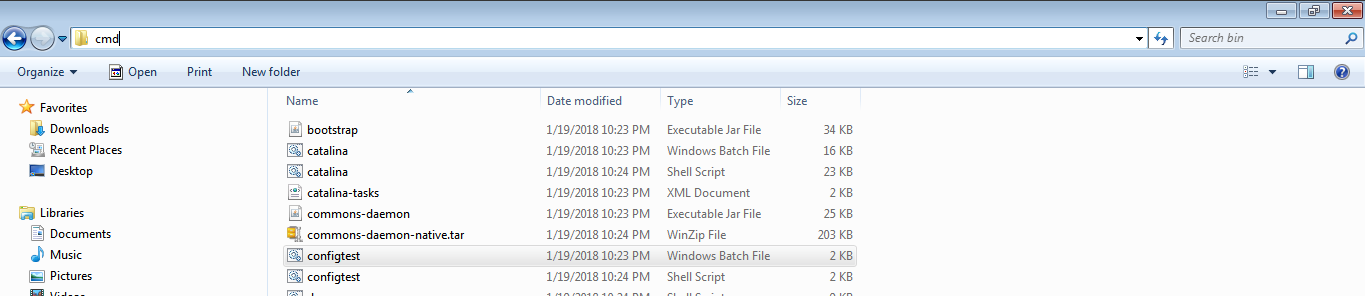
</project>



 paste the war 



Then Type CMD in



**USE POST MAN TO Check whether The REST Call Working**

List

http://localhost:8080/SpringRestHibernate/students --get

Insert

http://localhost:8080/SpringRestHibernate/create --post Raw Json

{"id":13,"name":"asdfm1","age":14,"dob":"14/10/96","degree":"Bsc","department":"dfdf","grade":"B"}

Delete

http://localhost:8080/SpringRestHibernate/create/delete/{id} -Delete

http://localhost:8080/SpringRestHibernate/delete/1

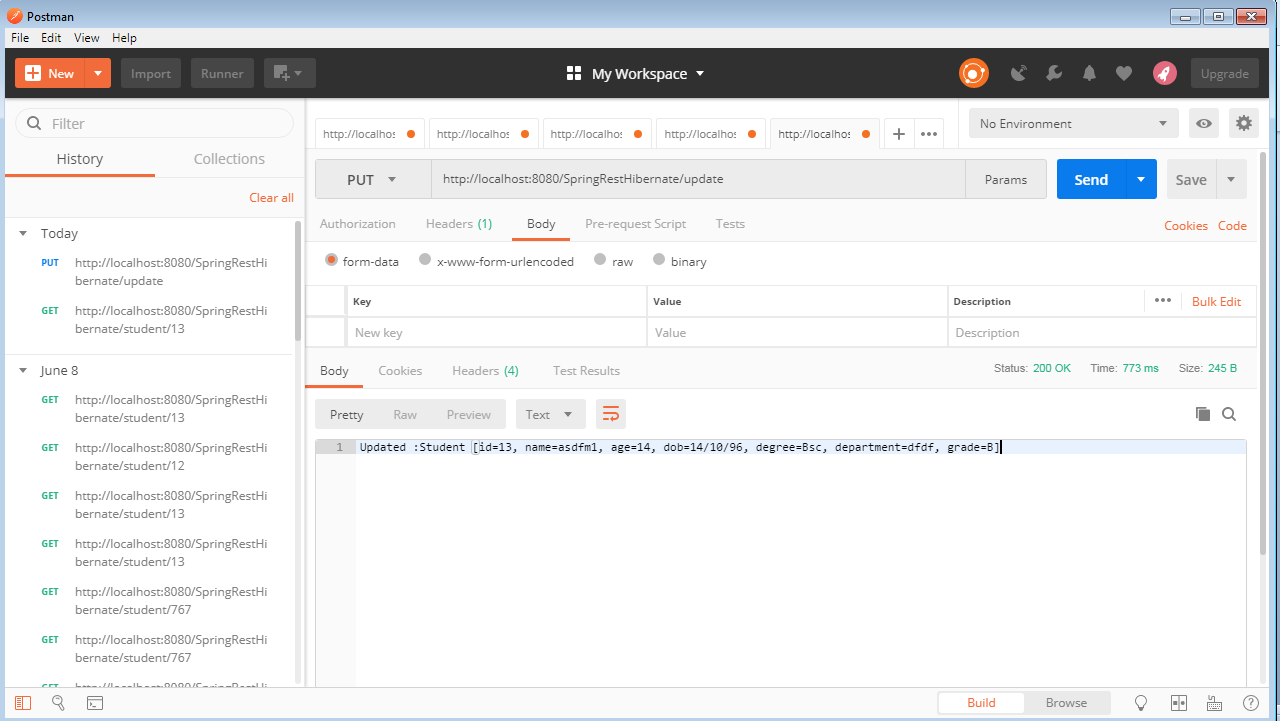
Getstudentbyid

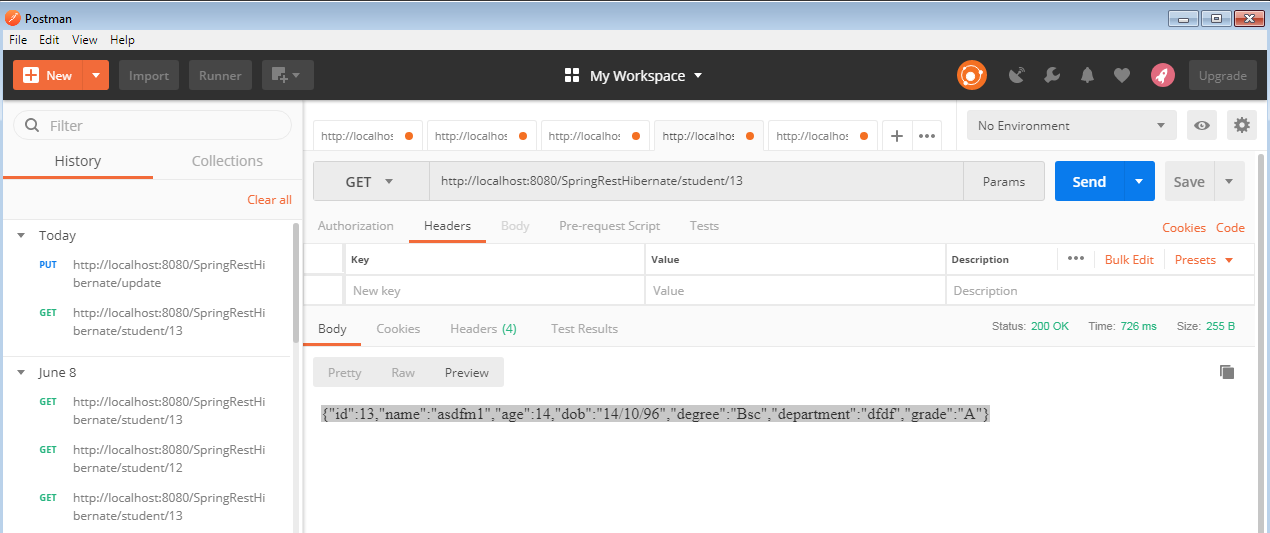
http://localhost:8080/SpringRestHibernate/student/1

Update

http://localhost:8080/SpringRestHibernate/update -put

{"id":13,"name":"asdfm1","age":14,"dob":"14/10/96","degree":"Bsc","department":"dfdf","grade":"B"}





Reffered

Try to understand this code

@Autowired

private BookService bookService;

/\*---Add new book---\*/

@PostMapping("/book")

public ResponseEntity<?> save(@RequestBody Book book) {

book.setAge(bookService.age(book.getDate()));

long id = bookService.save(book);

return ResponseEntity.ok().body("New Book has been saved with ID:" + id);

}

/\*---Get a book by id---\*/

@GetMapping("/book/{id}")

public ResponseEntity<Book> get(@PathVariable("id") long id) {

Book book = bookService.get(id);

return ResponseEntity.ok().body(book);

}

/\*---get all books---\*/

@GetMapping("/book")

public ResponseEntity<List<Book>> list() {

List<Book> books = bookService.list();

return ResponseEntity.ok().body(books);

}

/\*---Update a book by id---\*/

@PutMapping("/book/{id}")

public ResponseEntity<?> update(@PathVariable("id") long id, @RequestBody Book book) {

bookService.update(id, book);

return ResponseEntity.ok().body("Book has been updated successfully.");

}

/\*---Delete a book by id---\*/

@DeleteMapping("/book/{id}")

public ResponseEntity<?> delete(@PathVariable("id") long id) {

bookService.delete(id);

return ResponseEntity.ok().body("Book has been deleted successfully.");

}

<http://njoonk.blogspot.com/2015/08/sonarqube-how-to-install-and-get-start.html>

ResponseEntity :

https://www.logicbig.com/tutorials/spring-framework/spring-web-mvc/request-response-entity.html