domain class:

**package** com.mintopp.bricks.domain;

**public** **class** Order {

**private** **int** orderNum;

**private** String description;

**private** String status;

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**private** **int** qty;

**public** Order()

{

}

**public** Order(**int** orderNum,String description,String status,**int** qty)

{

**this**.orderNum=orderNum;

**this**.description=description;

**this**.status=status;

**this**.qty=qty;

}

**public** **int** getOrderNum() {

**return** orderNum;

}

**public** **void** setOrderNum(**int** orderNum) {

**this**.orderNum = orderNum;

}

**public** String getDescription() {

**return** description;

}

**public** **void** setDescription(String description) {

**this**.description = description;

}

**public** **int** getQty() {

**return** qty;

}

**public** **void** setQty(**int** qty) {

**this**.qty = qty;

}

}

-------------------Order Controller -----------

**package** com.mintopp.bricks.resources;

**import** java.util.List;

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

**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** com.mintopp.bricks.domain.Order;

**import** com.mintopp.bricks.services.OrderService;

@RestController

**public** **class** OrderResource {

OrderService orderService= **new** OrderService();

@RequestMapping("/")

**public** String welcome() {//Welcome page, non-rest

**return** "Welcome to RestTemplate Example.";

}

//stage 3

//method for implementing order status

@RequestMapping(value = "/orders/{id}", method = RequestMethod.***GET***, headers = "Accept=application/json")

**public** Order getOrderById(@PathVariable **int** id) {

**return** orderService.getOrder(id);

}

@RequestMapping(value = "/orders", method = RequestMethod.***GET***,headers = "Accept=application/json")

**public** List<Order> getAllOrders()

{

**return** orderService.getAllOrders();

}

@RequestMapping(value="/orders", method=RequestMethod.***POST***,headers = "Accept=application/json")

**public** Order addOrder(@RequestBody Order order) {

**return** orderService.addOrder(order);

}

@RequestMapping(value="/orders", method=RequestMethod.***PUT***,headers = "Accept=application/json")

**public** String updateOrder(@RequestBody Order order) {

**return** orderService.updateOrder(order);

}

@RequestMapping(value="/orders", method=RequestMethod.***DELETE***,headers = "Accept=application/json")

**public** ResponseEntity deleteOrder(@RequestBody Order order) {

**return** orderService.deleteOrder(order);

}

}

----------------------

------------Order services ------------

**package** com.mintopp.bricks.services;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map;

**import** javax.xml.ws.Response;

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

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

**import** org.springframework.stereotype.Service;

**import** com.mintopp.bricks.domain.Order;

@Service

**public** **class** OrderService {

**private** Response respone;

**private** **static** Map<Integer,Order> *orders*=**new** HashMap<Integer,Order>();

**public** OrderService()

{

*orders*.put(**new** Integer(123),**new** Order(123,"Dell Laptop","delivered",10));

*orders*.put(**new** Integer(567),**new** Order(567,"Dell Laptop","dispatched",20));

*orders*.put(**new** Integer(234),**new** Order(234,"Lenovo Laptop","inproduction",30));

*orders*.put(**new** Integer(345),**new** Order(345,"Mac Laptop","dispatched",10));

}

**public** List<Order> getAllOrders()

{

List<Order> ordersList = **new** ArrayList(*orders*.values());

**return** ordersList;

}

//order is searched in the map with order id

**public** Order getOrder(**int** orderId)

{

Order order= *orders*.get(orderId);

**return** order;

}

**public** Order addOrder(Order order)

{

order.setOrderNum(*getMaxId*()+1);

*orders*.put(order.getOrderNum(), order);

**return** order;

}

**public** String updateOrder(Order order)

{

**if** (order.getStatus().equalsIgnoreCase("dispatched"))

{

**return** ResponseEntity.*status*(HttpStatus.***BAD\_REQUEST***).toString();

}

**else**{

*orders*.put(order.getOrderNum(), order);

}

**return** "order updated " + order.getOrderNum() ;

}

**public** **static** **int** getMaxId()

{ **int** max=0;

**for** (**int** id:*orders*.keySet()) {

**if**(max<=id)

max=id;

}

**return** max;

}

**public** ResponseEntity deleteOrder(Order order)

{

*orders*.remove(order.getOrderNum());

**return** ResponseEntity.*ok*("order deleted" + order.getOrderNum());

//order.getOrderNum();

}

}// end of the service class

------------------------

-------------------------web.xml ------------

<!DOCTYPE web-app PUBLIC

"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

"http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<web-app>

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

<servlet>

<servlet-name>bricks</servlet-name>

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

</servlet>

<servlet-mapping>

<servlet-name>bricks</servlet-name>

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

</servlet-mapping>

</web-app>

---------------------------bricks-servlet.xml---------

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

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

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-3.0.xsd "*>

<context:annotation-config />

<mvc:annotation-driven />

<context:component-scan base-package=*"com.mintopp.bricks"* />

</beans>

------------------------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>com.mintopp.brick</groupId>

<artifactId>bricks</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>bricks Maven Webapp</name>

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

<properties>

<springframework.version>4.3.0.RELEASE</springframework.version>

<jackson.library>2.7.5</jackson.library>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${springframework.version}</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

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

<version>3.1.0</version>

</dependency>

<dependency>

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

<artifactId>jackson-databind</artifactId>

<version>${jackson.library}</version>

</dependency>

<dependency>

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

<artifactId>jackson-dataformat-xml</artifactId>

<version>${jackson.library}</version>

</dependency>

</dependencies>

<build>

<finalName>bricks</finalName>

</build>

</project>

--------------end point urls

http://localhost:8080/bricks/orders/ (for getting all the orders)

http://localhost:8080/bricks/orders/123 (for particular order)