**SpringBoot-SpringDataJPA-MONGODB-SOAPUI Sample Application**

**Prerequisites**

We must have installed the following

* JDK 1.7 or later
* Maven 3 or later

**Project Structure**

├── src

│   └── main

│   ├── java

│   │   └── com

│   │   └── springboot

│   │   └── springdatajpa

│   │   ├──Controller

│   │   │ └──BookingController.java

│ │ ├──models

│ │ │ ├── Booking.java

│   │ │ └── BookingRepository.java

│  │  SpringBootJpaSpringDataApplication.java

│   └── resources

│     ├── application.properties

│ └

│

└── pom.xml

**Project dependencies**

|  |
| --- |
| **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 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>com.springboot.springdatajpa.mongodb</groupId>  <artifactId>springbootmongo</artifactId>  <version>0.0.1-SNAPSHOT</version>  <packaging>jar</packaging>  <name>SpringBootMongoDB</name>  <description>SpringBootMongoDB project for Spring Boot with MongoDB providing APIs</description>  <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>1.4.0.RELEASE</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  <java.version>1.8</java.version>  </properties>  <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-data-mongodb</artifactId>  </dependency>  <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>  </plugins>  </build>  </project> |

**Application Configuration**

|  |
| --- |
| **SpringBootJpaSpringDataApplication.java** |
| **Path 🡪 src/main/java/com/test/sdjpa/ SpringBootJpaSpringDataApplication.java** |
| package com.test.sdjpa;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  @SpringBootApplication  public class SpringBootJpaSpringDataApplication {  public static void main(String[] args) {  SpringApplication.run(SpringBootJpaSpringDataApplication.class, args);  }  } |

**Controller File**

|  |
| --- |
| **BookingController.java** |
| **Path 🡪 src/main/java/com/test/sdjpa/** **BookingController.java** |
| package com.springboot.springdatajpa.controller;  import java.util.Date;  import java.util.HashMap;  import java.util.List;  import java.util.Map;  import javax.validation.Valid;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.web.bind.annotation.RequestBody;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RequestParam;  import org.springframework.web.bind.annotation.RestController;  import com.springboot.springdatajpa.models.Booking;  import com.springboot.springdatajpa.models.BookingRepository;  @RestController  @RequestMapping("/booking")  public class BookingController {    @Autowired  BookingRepository bookingRepository;  /\*\*  \* GET /create --> Create a new booking and save it in the database.  \*/      @RequestMapping("/create")  public Booking create(@Valid @RequestBody Booking booking) {    try{    booking.setTravelDate(new Date());    booking = bookingRepository.save(booking);      }catch(Exception e){  System.out.println("Connection Exception"+e.getMessage());  }  return booking;  }      /\*\*  \* GET /read --> Read a booking by booking id from the database.  \*/    @RequestMapping("/read")  public Booking read(@Valid @RequestBody Booking booking) {  Booking book = bookingRepository.findOne(booking.getId());    return book;  }  /\*\*  \* GET /update --> Update a booking record and save it in the database.  \*/  @RequestMapping("/update")  public Booking update(@Valid @RequestBody Booking book) {  Booking booking = bookingRepository.findOne(book.getId());  booking.setPsngrName(book.getPsngrName());  booking.setDeparture(book.getDeparture());  booking.setDestination(book.getDestination());  booking.setTravelDate(new Date());  booking = bookingRepository.save(booking);    return booking;  }    /\*\*  \* GET /delete --> Delete a booking from the database.  \*/  @RequestMapping("/delete")  public Map<String, Object> delete(@Valid @RequestBody Booking book) {  Map<String, Object> dataMap = new HashMap<String, Object>();  bookingRepository.delete(book.getId());    dataMap.put("booking", book);  dataMap.put("Status", "Successfully Deleted");    return dataMap;  }    /\*\*  \* GET /read --> Read all booking from the database.  \*/  @RequestMapping("/read-all")  public List<Booking> readAll() {    List<Booking> bookings = bookingRepository.findAll();  return bookings;  }  } |

**Model File**

|  |
| --- |
| **Booking.java** |
| **Path 🡪 src/main/java/com/test/sdjpa/** **Booking.java** |
| package com.springboot.springdatajpa.models;  import java.util.Date;  import org.springframework.data.annotation.Id;  import org.springframework.data.mongodb.core.mapping.Document;  @Document  public class Booking{  @Id  String id;  String psngrName;  String departure;  String destination;  Date travelDate;  public String getId() {  return id;  }  public void setId(String id) {  this.id = id;  }  public String getPsngrName() {  return psngrName;  }  public void setPsngrName(String psngrName) {  this.psngrName = psngrName;  }  public String getDeparture() {  return departure;  }  public void setDeparture(String departure) {  this.departure = departure;  }  public String getDestination() {  return destination;  }  public void setDestination(String destination) {  this.destination = destination;  }  public Date getTravelDate() {  return travelDate;  }  public void setTravelDate(Date travelDate) {  this.travelDate = travelDate;  }      } |

|  |
| --- |
| **BookingRepository.java** |
| **Path 🡪 src/main/java/com/test/sdjpa/** **BookingRepository.java** |
| package com.springboot.springdatajpa.models;  import org.springframework.data.mongodb.repository.MongoRepository;  import org.springframework.transaction.annotation.Transactional;  @Transactional  public interface BookingRepository extends MongoRepository<Booking, String> {    /\*\*  \* This method will find an Boooking instance in the database by its departure.  \* Note that this method is not implemented and its working code will be  \* automatically generated from its signature by Spring Data JPA.  \*/  public Booking findByDeparture(String departure);  } |

**Application.properties**

|  |
| --- |
| **application.properties** |
| **Path 🡪 src/main/resources/application.properties** |
| server.port: 8083  spring.data.mongodb.database=test  spring.data.mongodb.host=localhost  spring.data.mongodb.port=27017 |

**Steps to Test the Application**

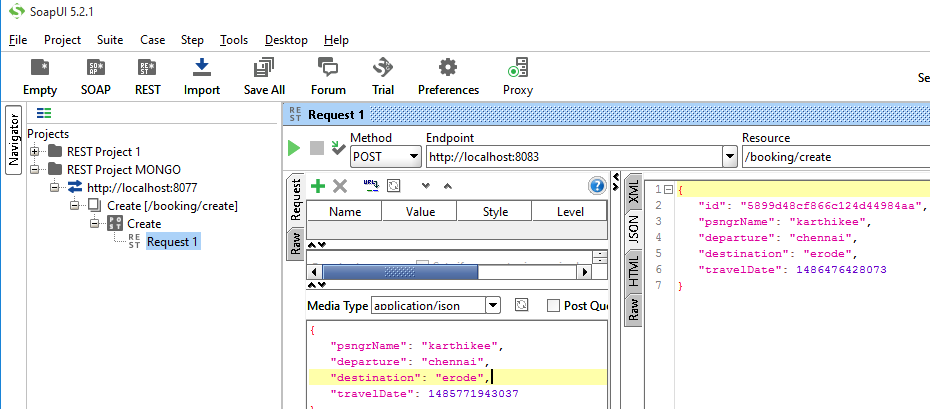
D:\Karthik-ws\code\SpringBoot-SpringDataJPA-MongoDB>mvn spring-boot:run

Once the server has been started. Enter the below url in the SOAPUI and select POST method.

**Insert**

[http://localhost:8083/booking/create](http://localhost:8181/booking/create)

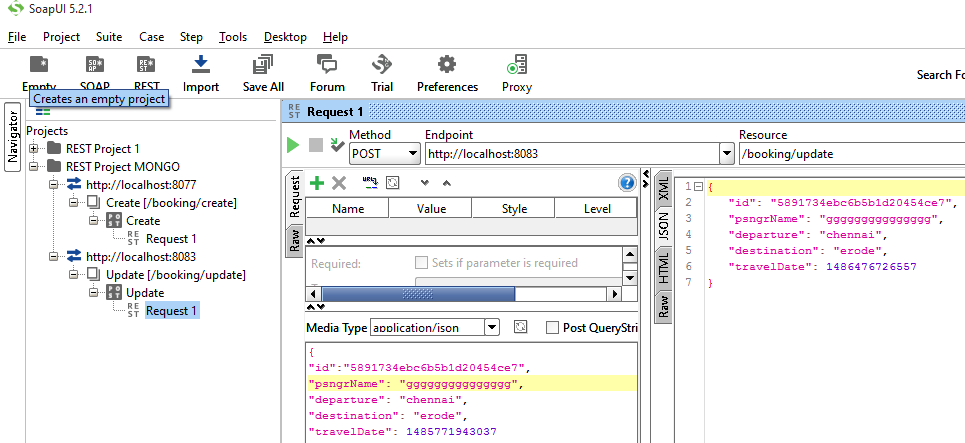
|  |
| --- |
| JSON Input |
| {  "psngrName": "karthik",  "departure": "chennai",  "destination": "erode",  "travelDate": 1485771943037  } |



**Update**

[http://localhost:8083/booking/update](http://localhost:8181/booking/update)

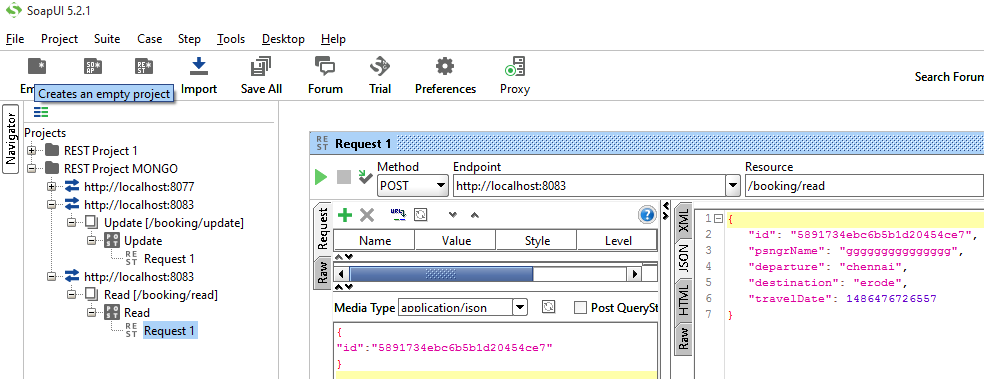
|  |
| --- |
| JSON Input |
| {  "id":"5891734ebc6b5b1d20454ce7",  "psngrName": "ggggggggggggggg",  "departure": "chennai",  "destination": "erode",  "travelDate": 1485771943037  } |



**View**

http://localhost:8083/booking/read

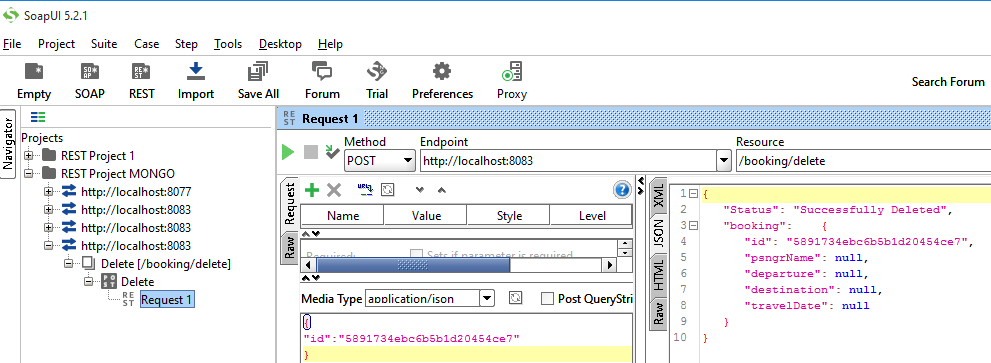
|  |
| --- |
| JSON Input |
| {  "id":"5891734ebc6b5b1d20454ce7"  } |



**Delete**

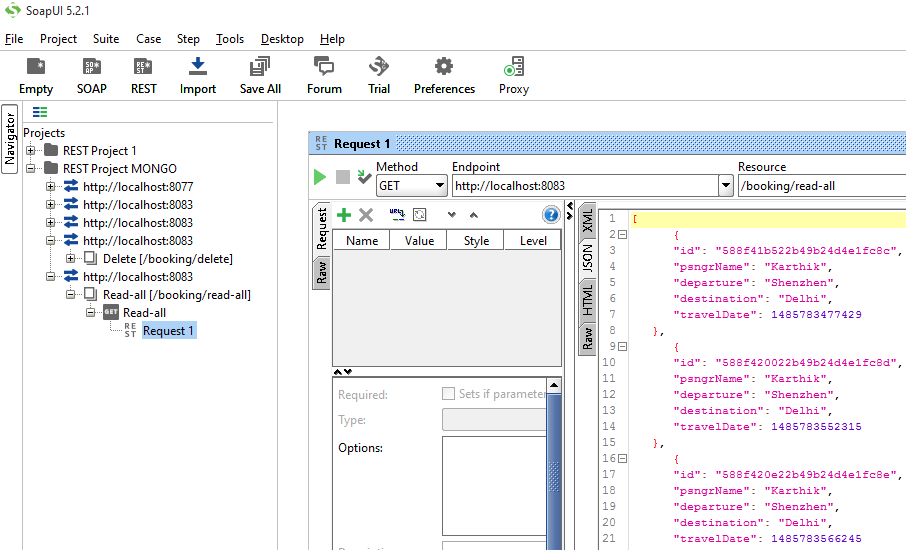
http://localhost:8083/booking/delete

|  |
| --- |
| JSON Input |
| {  "id":"5891734ebc6b5b1d20454ce7"  } |



**View All**

<http://localhost:8083/booking/read-all>



**Test in mysql Client**

