# Application.java

* package com.cognizant.fsd.restfulapiassignment1;
* import org.springframework.boot.SpringApplication;
* import org.springframework.boot.autoconfigure.SpringBootApplication;
* @SpringBootApplication
* public class Application {
* public static void main(String[] args) {
* SpringApplication.run(Application.class, args);
* }
* }

# BookStoreController.java

* package com.cognizant.fsd.restfulapiassignment1.controller;
* import java.util.List;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.web.bind.annotation.CrossOrigin;
* import org.springframework.web.bind.annotation.DeleteMapping;
* import org.springframework.web.bind.annotation.GetMapping;
* import org.springframework.web.bind.annotation.PathVariable;
* import org.springframework.web.bind.annotation.PostMapping;
* import org.springframework.web.bind.annotation.PutMapping;
* 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.cognizant.fsd.restfulapiassignment1.model.Book;
* import com.cognizant.fsd.restfulapiassignment1.service.BookStoreService;
* @CrossOrigin(origins = "http://localhost:3000")
* @RestController
* public class BookStoreController {
* @Autowired
* private BookStoreService bookStoreService;
* @GetMapping("/bookstore/books/{bookId}")
* public Book getBookById(@PathVariable String bookId) {
* Book bookDetails = bookStoreService.getBookById(Long.parseLong(bookId));
* return bookDetails;
* }
* @GetMapping("/bookstore/books")
* public List<Book> getAllBook(){
* List<Book> allBookList = bookStoreService.getAllBooks();
* return allBookList;
* }
* @RequestMapping(method = RequestMethod.POST, value="/bookstore/books")
* public void addBook(@RequestBody Book book) {
* System.out.println("Adding Book..");
* bookStoreService.addBook(book);
* }
* @RequestMapping(method=RequestMethod.PUT, value="/bookstore/books/{bookId}")
* public void editBook(@RequestBody Book book,@PathVariable String bookId) {
* bookStoreService.editBook(book,Long.parseLong(bookId));
* }
* @RequestMapping(method=RequestMethod.DELETE,value="/bookstore/books/{bookId}")
* public void deleteBook(@PathVariable String bookId) {
* bookStoreService.deleteBook(Long.parseLong(bookId));
* }

* }

# Book.java

* package com.cognizant.fsd.restfulapiassignment1.model;
* import java.io.Serializable;
* import java.time.LocalDate;
* public class Book implements Serializable{
* private static final long serialVersionUID = 1L;
* private long bookId;
* private String title;
* private double price;
* private Integer volume;
* private LocalDate publishDate;

* public Book() {
* super();
* // TODO Auto-generated constructor stub
* }
* public Book(long bookId, String title, double price, Integer volume, LocalDate publishDate) {
* super();
* this.bookId = bookId;
* this.title = title;
* this.price = price;
* this.volume = volume;
* this.publishDate = publishDate;
* }
* public long getBookId() {
* return bookId;
* }
* public void setBookId(long bookId) {
* this.bookId = bookId;
* }
* public String getTitle() {
* return title;
* }
* public void setTitle(String title) {
* this.title = title;
* }
* public double getPrice() {
* return price;
* }
* public void setPrice(double price) {
* this.price = price;
* }
* public Integer getVolume() {
* return volume;
* }
* public void setVolume(Integer volume) {
* this.volume = volume;
* }
* public LocalDate getPublishDate() {
* return publishDate;
* }
* public void setPublishDate(LocalDate publishDate) {
* this.publishDate = publishDate;
* }
* @Override
* public String toString() {
* return "Book [bookId=" + bookId + ", title=" + title + ", price=" + price + ", volume=" + volume
* + ", publishDate=" + publishDate + "]";
* }
* }

# Subject.java

* package com.cognizant.fsd.restfulapiassignment1.model;
* import java.io.Serializable;
* import java.util.HashSet;
* import java.util.Set;
* public class Subject implements Serializable {
* private static final long serialVersionUID = 1L;
* private long subjectId;
* private String subTitle;
* private int durationInHours;
* private Set<Book> references=new HashSet<Book>();
* public long getSubjectId() {
* return subjectId;
* }
* public void setSubjectId(long subjectId) {
* this.subjectId = subjectId;
* }
* public String getSubTitle() {
* return subTitle;
* }
* public void setSubTitle(String subTitle) {
* this.subTitle = subTitle;
* }
* public int getDurationInHours() {
* return durationInHours;
* }
* public void setDurationInHours(int durationInHours) {
* this.durationInHours = durationInHours;
* }
* public Set<Book> getReferences() {
* return references;
* }
* public void setReferences(Set<Book> references) {
* this.references = references;
* }
* @Override
* public String toString() {
* return "Subject [subjectId=" + subjectId + ", subTitle=" + subTitle + ", durationInHours=" + durationInHours
* + ", references=" + references + "]";
* }
* }

# BookStoreService.java

* /\*\*
* \*
* \*/
* package com.cognizant.fsd.restfulapiassignment1.service;
* import java.time.LocalDate;
* import java.util.ArrayList;
* import java.util.Arrays;
* import java.util.List;
* import org.springframework.stereotype.Component;
* import com.cognizant.fsd.restfulapiassignment1.model.Book;
* @Component
* public class BookStoreService implements IBookStoreService {
* private static List<Book> books = new ArrayList<Book>();
* static {
* //Initialize Data
* Book book1 = new Book(1, "Basic Java", 300, 3, LocalDate.of(2016, 2, 12));
* Book book2 = new Book(2, "Advance Java", 400, 4, LocalDate.of(2018, 8, 10));
* Book book3 = new Book(3, "Spring Boot", 500, 5, LocalDate.of(2017, 5, 11));
* Book book4 = new Book(4, "Jenkins", 600, 6, LocalDate.of(2015, 6, 11));
* books.add(book1);
* books.add(book2);
* books.add(book3);
* books.add(book4);
* }
* @Override
* public void addBook(Book book) {
* System.out.println(">> Inside Service >>");
* books.add(book);
* }
* @Override
* public void editBook(Book book,long bookId) {
* Book record = getBookById(bookId);
* books.remove(record);
* book.setBookId(bookId);
* books.add(book);
* }
* @Override
* public boolean deleteBook(long bookId) {
* Book record = getBookById(bookId);
* books.remove(record);
* return Boolean.TRUE;
* }
* @Override
* public Book getBookById(long bookId) {
* return books.stream().filter(b ->b.getBookId() == bookId).findFirst().get();
* }
* @Override
* public List<Book> getAllBooks() {
* return books;
* }
* }

# IBookStoreService.java

* package com.cognizant.fsd.restfulapiassignment1.service;
* import java.util.List;
* import com.cognizant.fsd.restfulapiassignment1.model.Book;
* public interface IBookStoreService {
* public void addBook(Book book);
* public void editBook(Book book,long bookId);
* public boolean deleteBook(long bookId);
* public Book getBookById(long bookId);
* public List<Book> getAllBooks();
* }

# application.properties

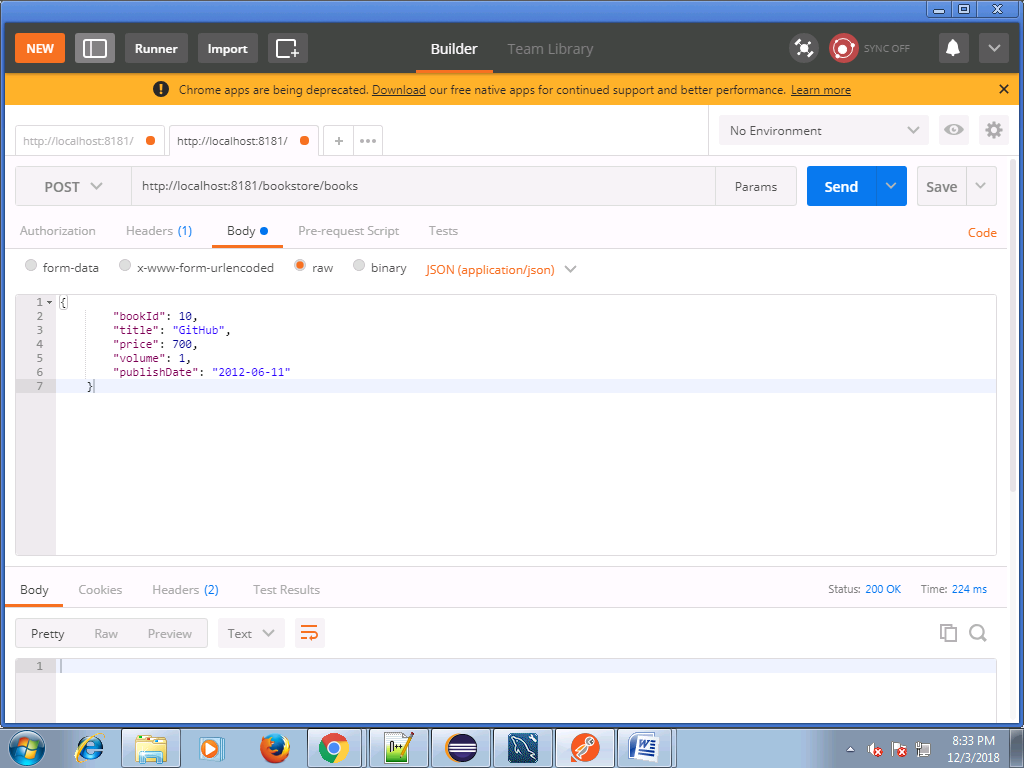
* server.port = 8181

# pom.xml

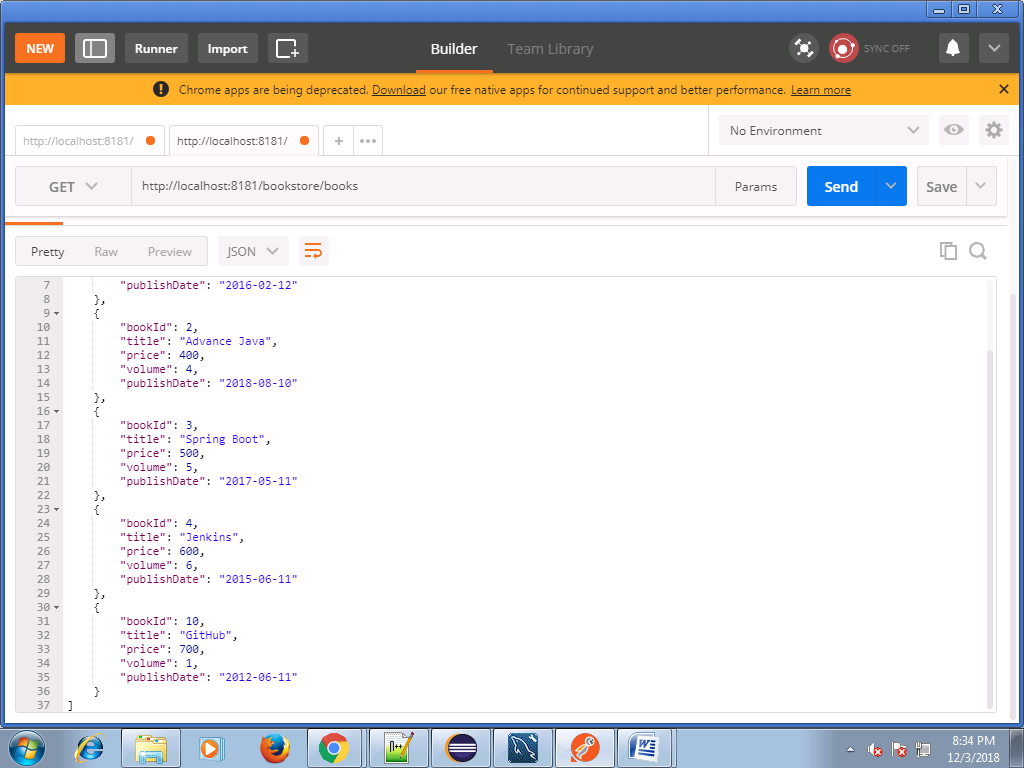
* <?xml version=*"1.0"* encoding=*"UTF-8"*?>
* <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*
* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>
* <modelVersion>4.0.0</modelVersion>
* <groupId>com.cognizant.fsd.restfulapiassignment1</groupId>
* <artifactId>11\_RestfulAPI\_Assignmet1</artifactId>
* <version>0.0.1-SNAPSHOT</version>
* <packaging>jar</packaging>
* <name>11\_RestfulAPI\_Assignmet1</name>
* <description>Demo project for Spring Boot</description>
* <parent>
* <groupId>org.springframework.boot</groupId>
* <artifactId>spring-boot-starter-parent</artifactId>
* <version>2.1.0.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-jpa</artifactId>
* </dependency> -->
* <!-- <dependency>
* <groupId>org.springframework.boot</groupId>
* <artifactId>spring-boot-starter-hateoas</artifactId>
* </dependency> -->
* <dependency>
* <groupId>org.springframework.boot</groupId>
* <artifactId>spring-boot-starter-jersey</artifactId>
* </dependency>
* <!-- <dependency>
* <groupId>org.springframework.boot</groupId>
* <artifactId>spring-boot-starter-security</artifactId>
* </dependency> -->
* <dependency>
* <groupId>org.springframework.boot</groupId>
* <artifactId>spring-boot-starter-web</artifactId>
* </dependency>
* <!-- <dependency>
* <groupId>org.springframework.boot</groupId>
* <artifactId>spring-boot-devtools</artifactId>
* <scope>runtime</scope>
* </dependency> -->
* <!-- <dependency>
* <groupId>mysql</groupId>
* <artifactId>mysql-connector-java</artifactId>
* <scope>runtime</scope>
* </dependency> -->
* <dependency>
* <groupId>org.springframework.boot</groupId>
* <artifactId>spring-boot-starter-test</artifactId>
* <scope>test</scope>
* </dependency>
* <!-- <dependency>
* <groupId>org.springframework.security</groupId>
* <artifactId>spring-security-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>

**Screenshots**

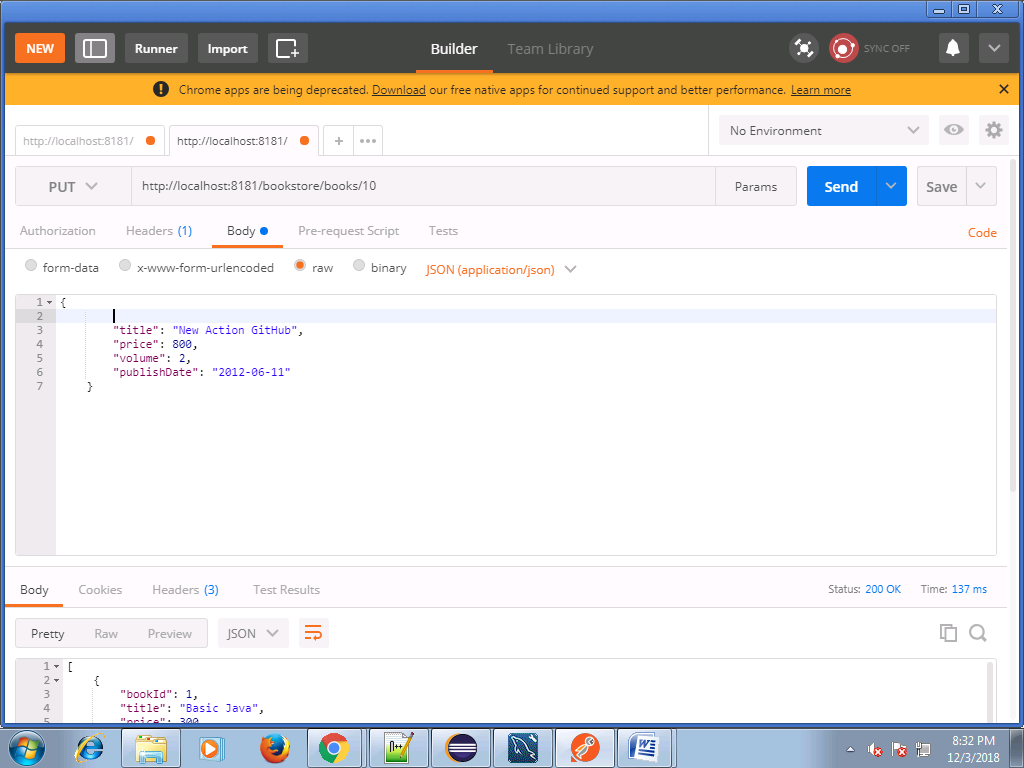
**a) POST Book(creates new book):**

****

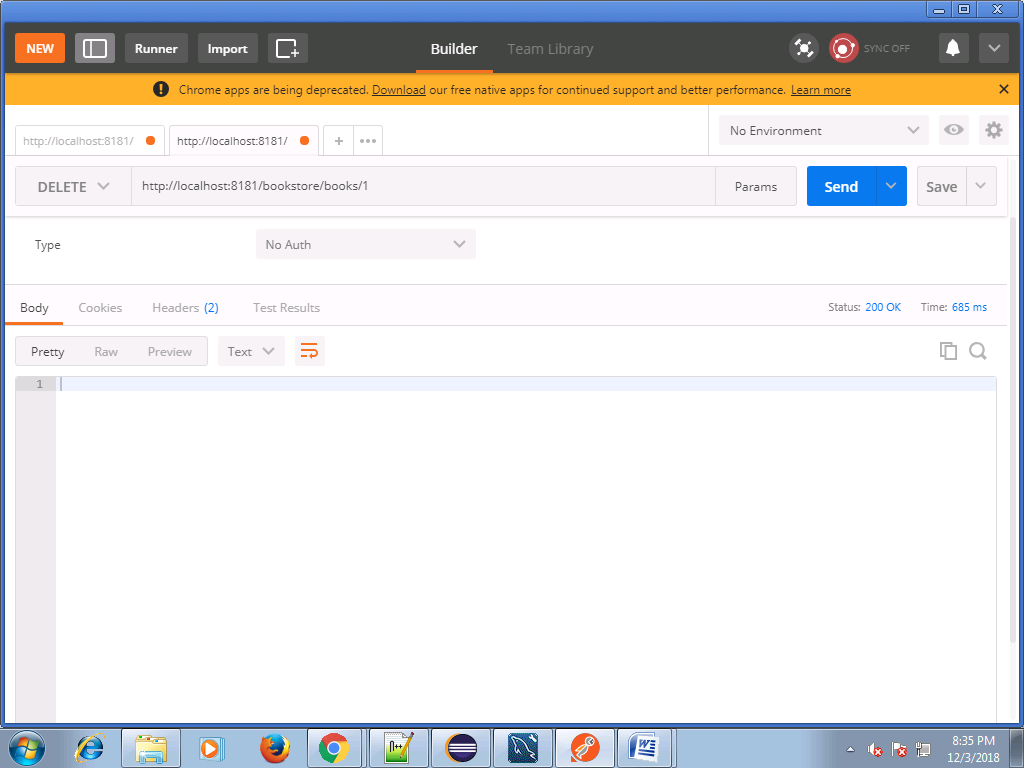
**get Request:**

****

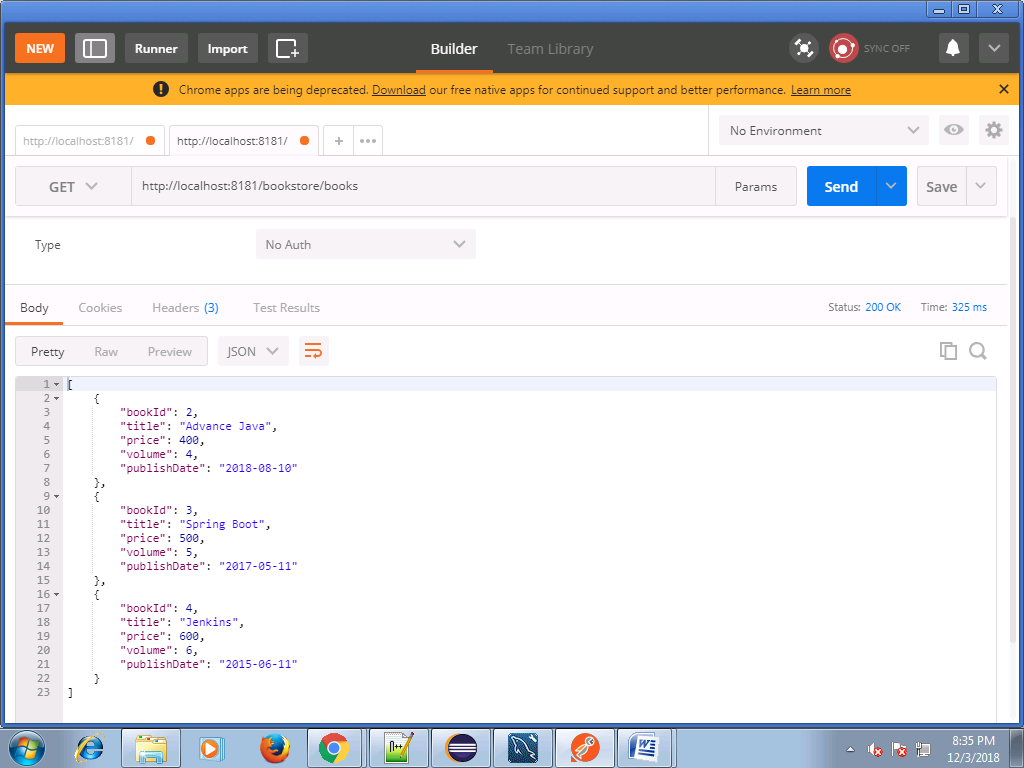
**b) PUT book (modify an existing book entry)**

****

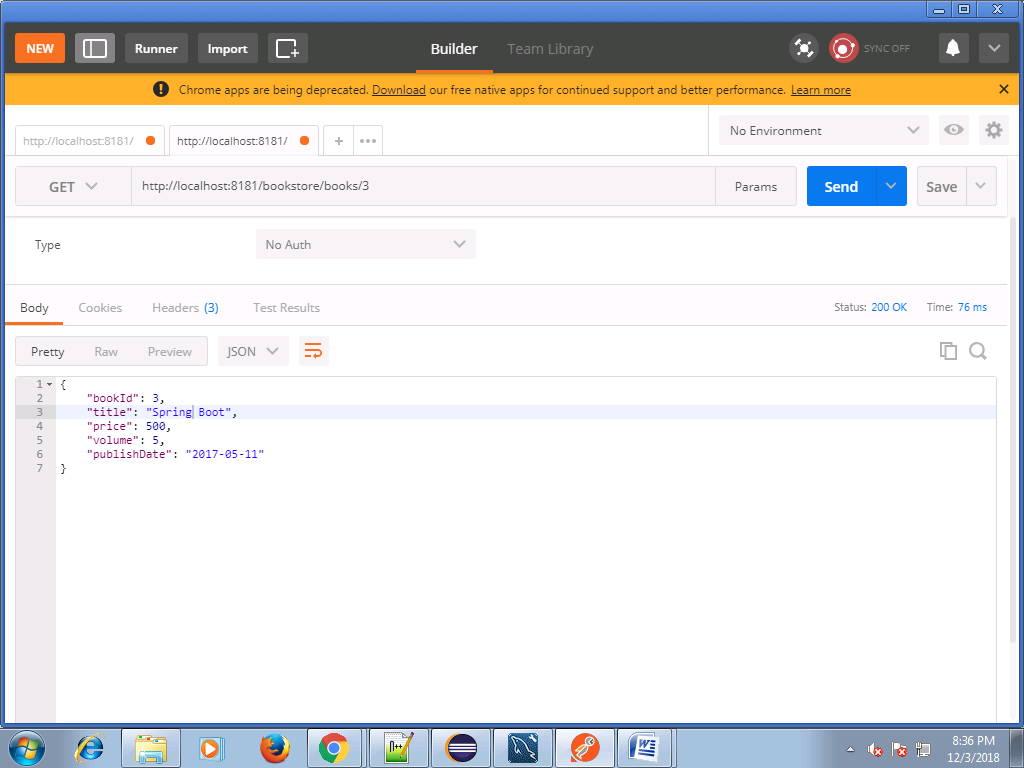
**c) DELETE Book**

****

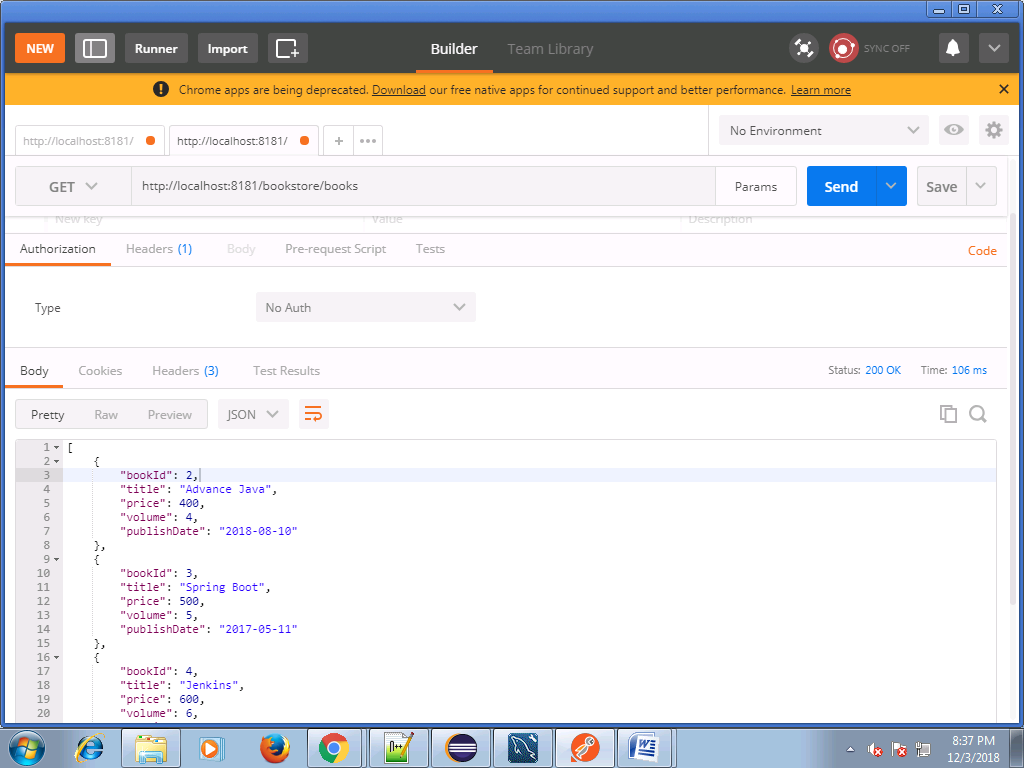
**get All Books**

****

**d) GET Book by id**

****

**e) GET all books**

****