

Program no 4:

Create a Spring Boot Application using Maven Plugin

- Write a sample REST Controller API using Spring Annotations
- Product details using GET, POST, PUT and DELETE methods.
- Using Postman invoke the REST Controller to demonstrate end to end working

**Book.java**

**package** com.example.restful\_demo;

**import** jakarta.annotation.Nonnull;

```
public class Book {  
    private Long id;  
  
    public Long getId() {  
        return id;  
    }  
    public void setId(Long id) {  
        this.id = id;  
    }  
    public String getTitle() {  
        return title;  
    }  
    public void setTitle(String title) {  
        this.title = title;  
    }  
    public String getAuthor() {  
        return author;  
    }  
    public void setAuthor(String author) {  
        this.author = author;  
    }  
    public int getPublicationYear() {  
        return publicationYear;  
    }  
    public void setPublicationYear(int publicationYear) {  
        this.publicationYear = publicationYear;  
    }  
    private String title;  
    private String author;  
    private int publicationYear;  
}
```

## BookController.java

```
package com.example.restful_demo;
//imports

@RestController
@RequestMapping("/api/books")
public class BookController {
    private final List<Book> books = new ArrayList<Book>();

    @GetMapping
    public List<Book> getAllBooks() {
        return books;
    }

    @GetMapping("/{id}")
    public Book getBookById(@PathVariable Long id) {
        return findBookById(id);
    }

    @PostMapping
    public Book addBook(@RequestBody Book book) {
        book.setId(generateBookId());
        books.add(book);
        return book;
    }

    @PutMapping("/{id}")
    public Book updateBook(@PathVariable Long id, @RequestBody Book updatedBook) {
        Book existingBook = findBookById(id);

        if (existingBook != null) {
            existingBook.setTitle(updatedBook.getTitle());
            existingBook.setAuthor(updatedBook.getAuthor());
            existingBook.setPublicationYear(updatedBook.getPublicationYear());
        }
        return existingBook;
    }

    @DeleteMapping("/{id}")
    public void deleteBook(@PathVariable Long id) {
        Book bookToRemove = findBookById(id);
        if (bookToRemove != null) {
            books.remove(bookToRemove);
        }
    }
}
```

```

// Helper method to find a book by ID
private Book findBookById(Long id) {
    for(Person p:al) {
        if(p.getName().equalsIgnoreCase(name)) {
            return p;
        }else {
            return null;
        }
    }
}
}
}

```

### **main class:**

```

package com.example.restful_demo;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class RestfulDemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(RestfulDemoApplication.class, args);
    }
}

```