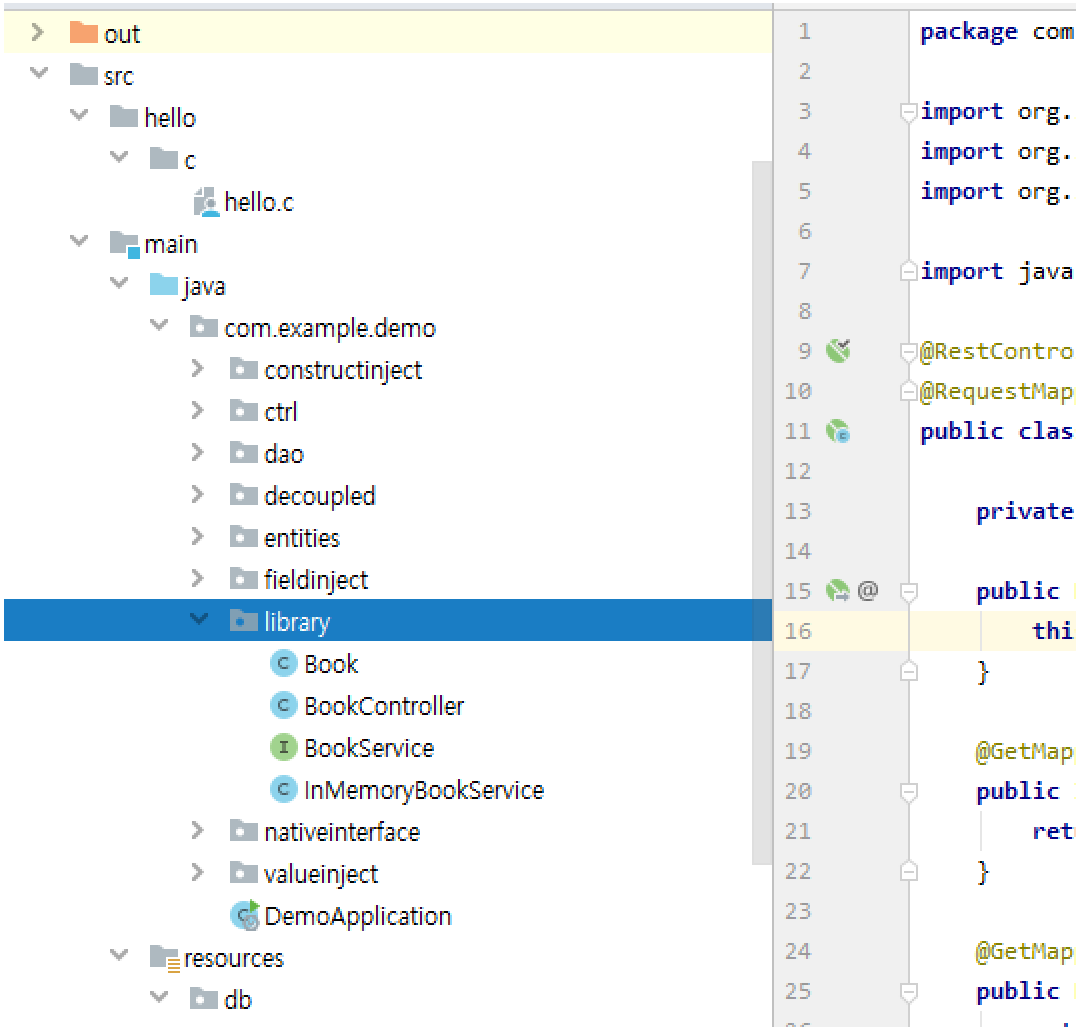
Spring In Memory Book

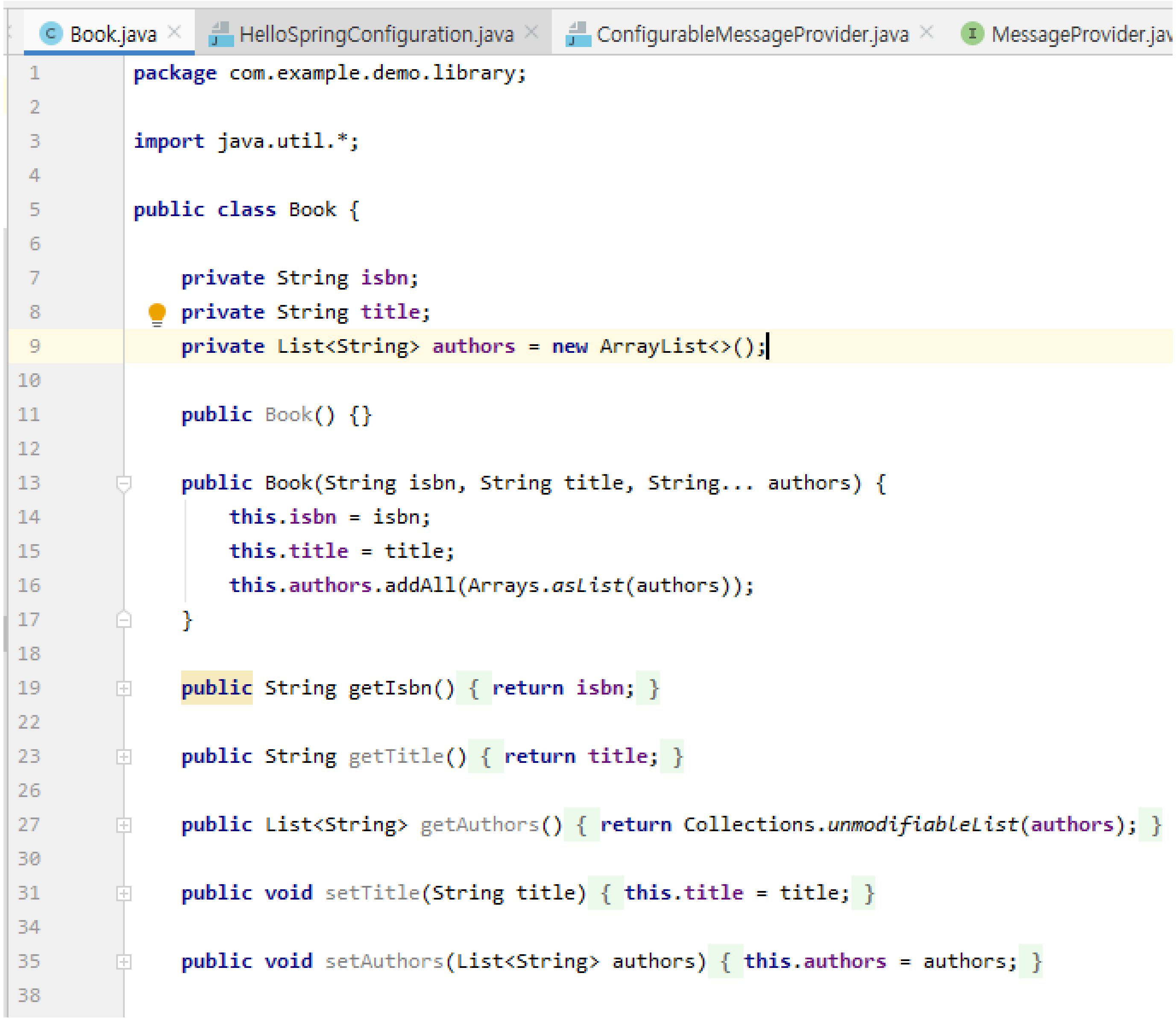
Innova Lee(이상훈)
gcccompil3r@gmail.com

IJ
IntelliJ IDEA
ULTIMATE 2019.2
JET
BRAINS

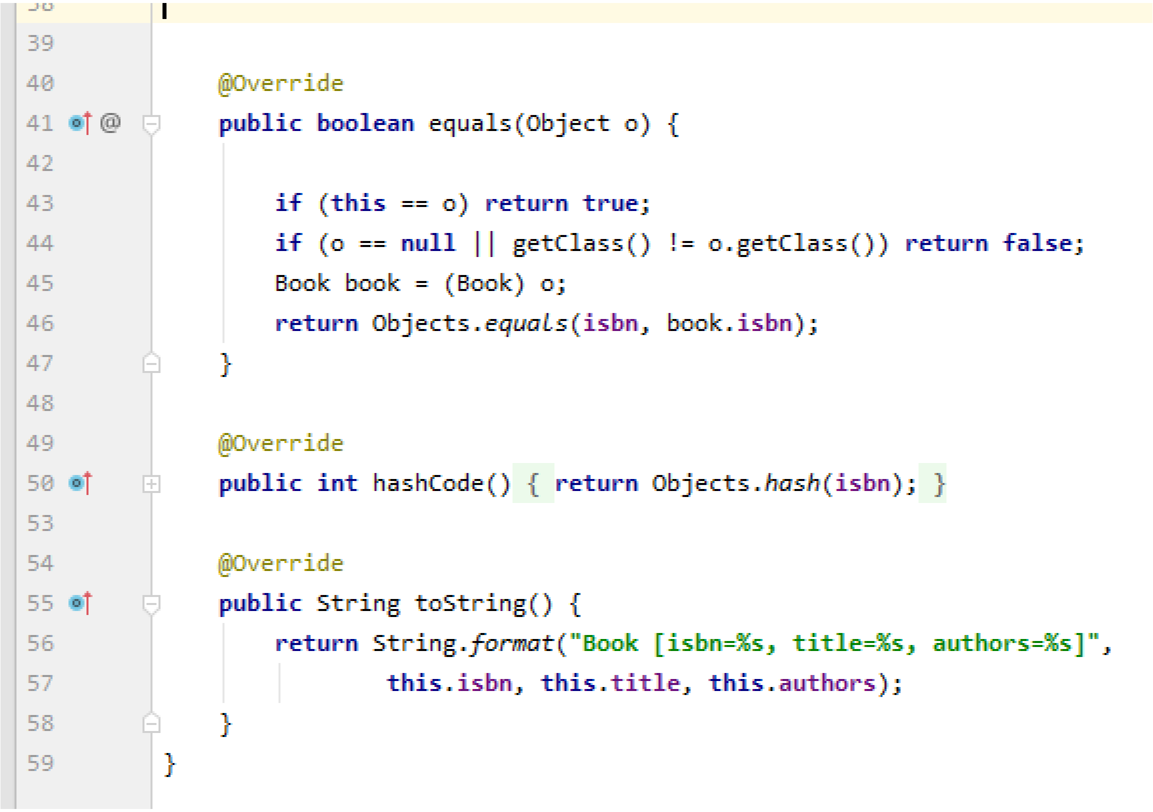


```
> out
v src
  v hello
    v c
        hello.c
  v main
    v java
      v com.example.demo
        > constructinject
        > ctrl
        > dao
        > decoupled
        > entities
        > fieldinject
        v library
            Book
            BookController
            BookService
            InMemoryBookService
        > nativeinterface
        > valueinject
          DemoApplication
    v resources
      v db
```

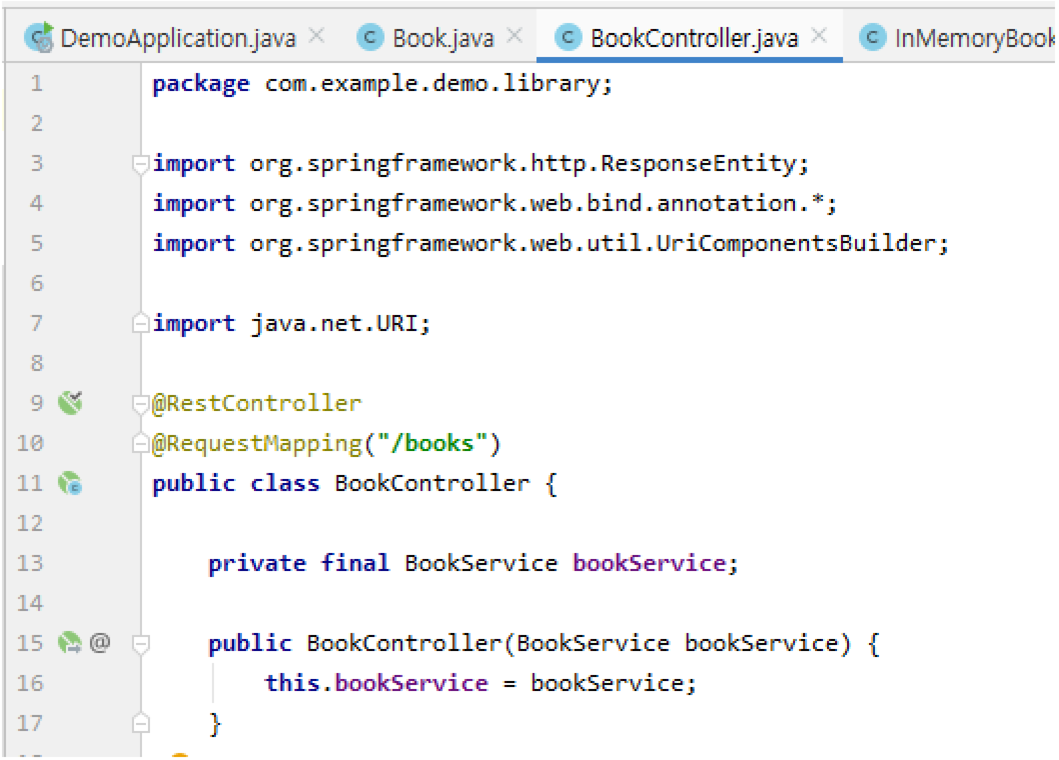
```
package com

import org.
import org.
import org.

import java

@RestContro
@RequestMap
public clas

    private

    public
        thi
    }

    @GetMap
    public
        ret
    }

    @GetMap
    public
```


Book.java
HelloSpringConfiguration.java
ConfigurableMessageProvider.java
MessageProvider.jav

```java
package com.example.demo.library;

import java.util.*;

public class Book {

    private String isbn;
    private String title;
    private List<String> authors = new ArrayList<>();

    public Book() {}

    public Book(String isbn, String title, String... authors) {
        this.isbn = isbn;
        this.title = title;
        this.authors.addAll(Arrays.asList(authors));
    }

    public String getIsbn() { return isbn; }

    public String getTitle() { return title; }

    public List<String> getAuthors() { return Collections.unmodifiableList(authors); }

    public void setTitle(String title) { this.title = title; }

    public void setAuthors(List<String> authors) { this.authors = authors; }

```

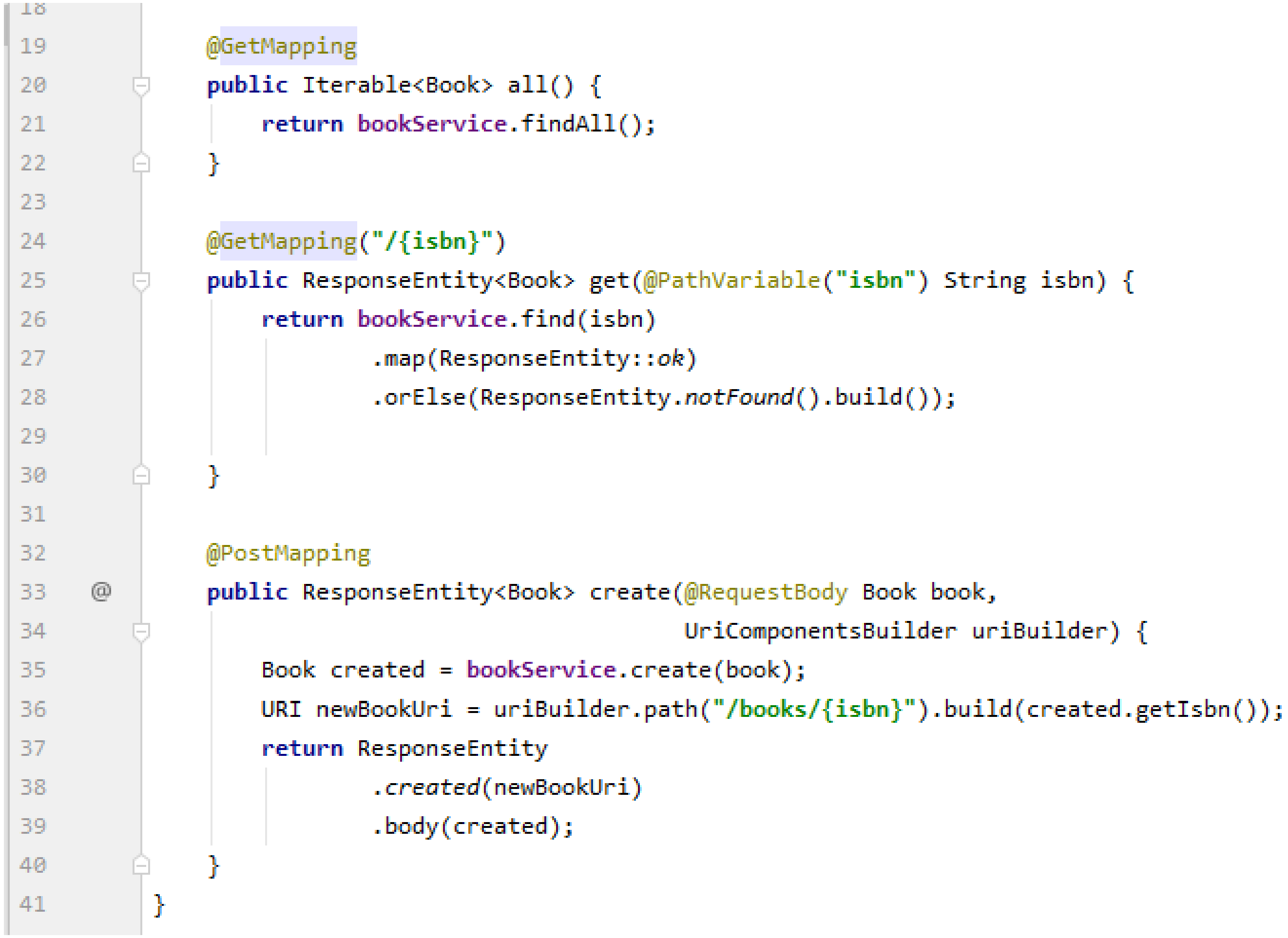


```java

    @Override
    public boolean equals(Object o) {

        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Book book = (Book) o;
        return Objects.equals(isbn, book.isbn);
    }

    @Override
    public int hashCode() { return Objects.hash(isbn); }

    @Override
    public String toString() {
        return String.format("Book [isbn=%s, title=%s, authors=%s]",
                this.isbn, this.title, this.authors);
    }
}
```

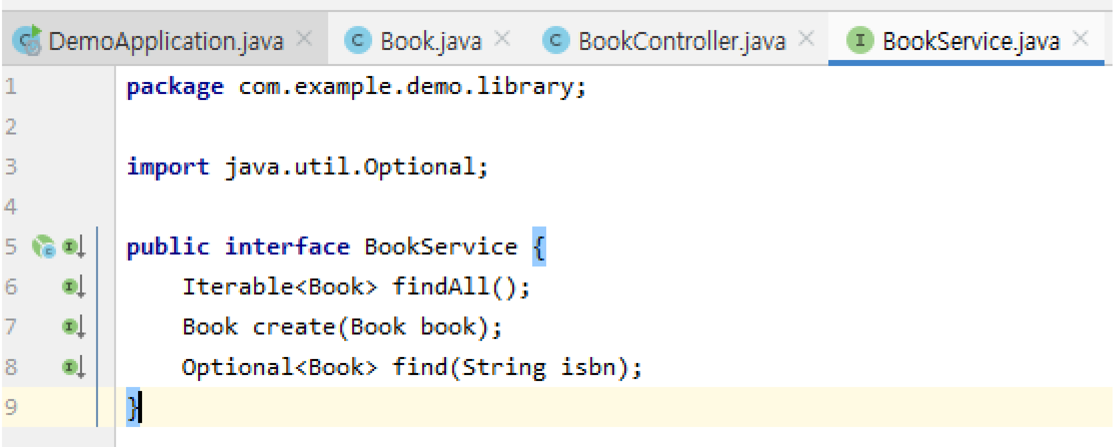


```java
package com.example.demo.library;

import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.util.UriComponentsBuilder;

import java.net.URI;

@RestController
@RequestMapping("/books")
public class BookController {

    private final BookService bookService;

    public BookController(BookService bookService) {
        this.bookService = bookService;
    }
```

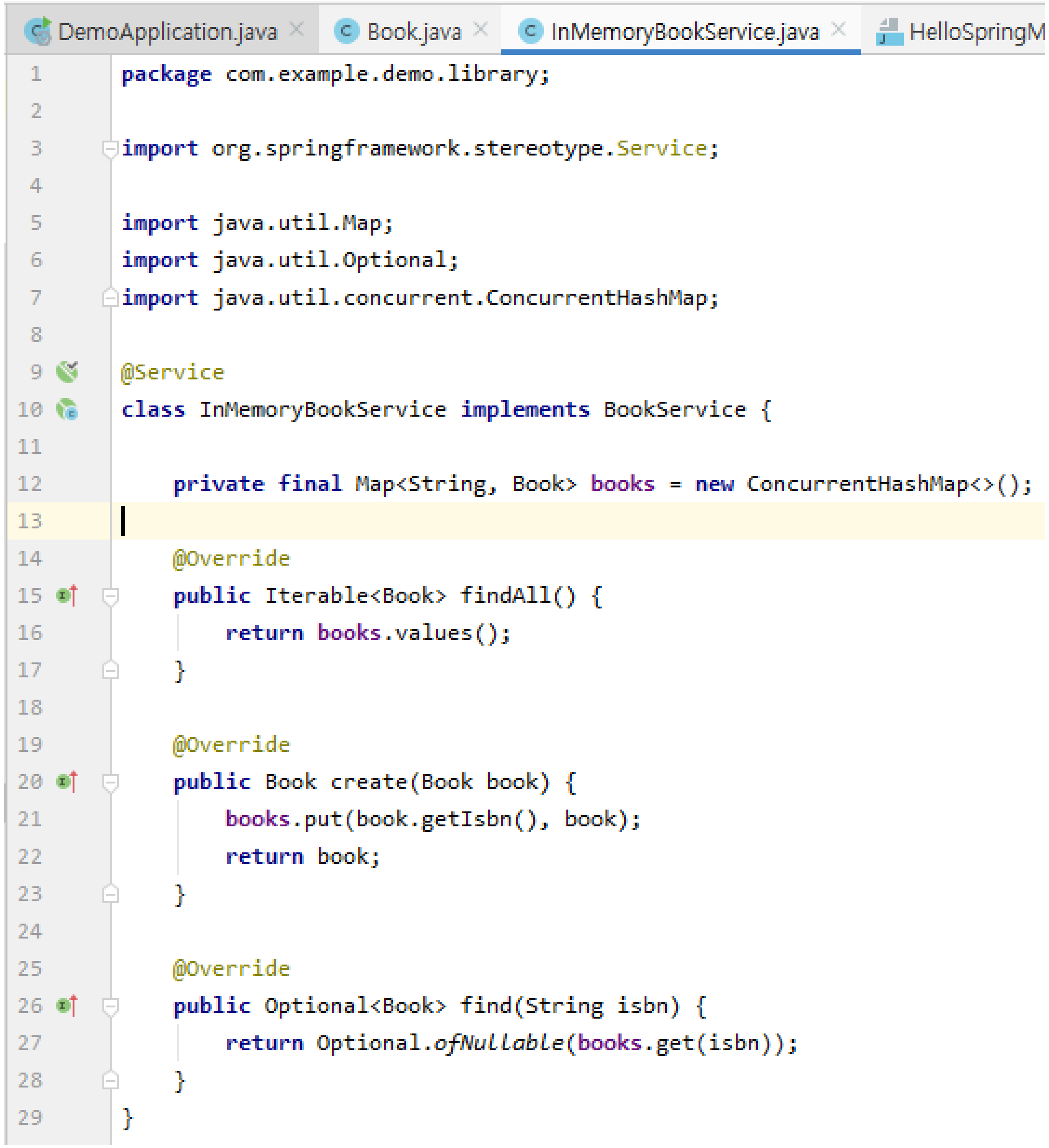
]



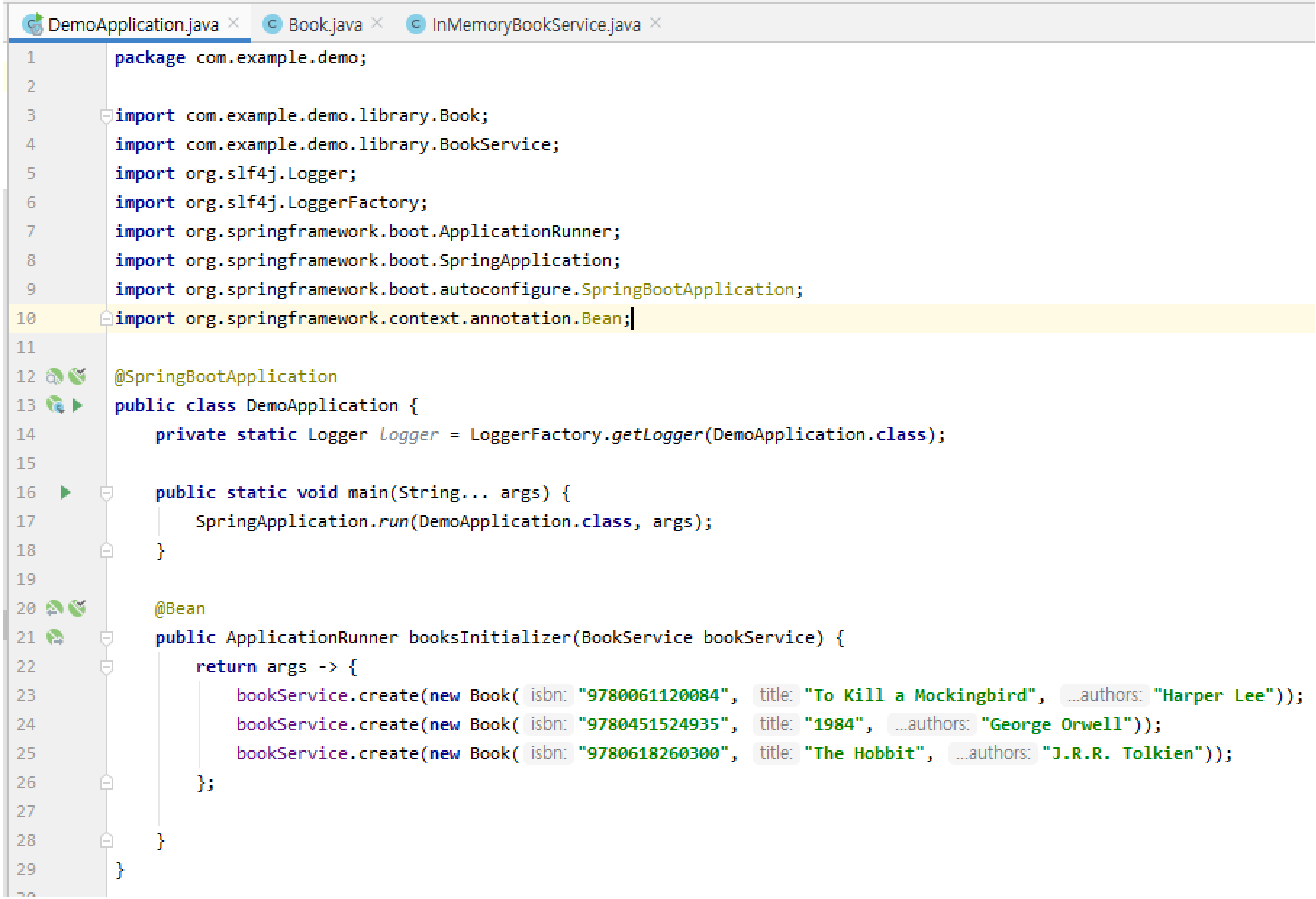
```java

    @GetMapping
    public Iterable<Book> all() {
        return bookService.findAll();
    }

    @GetMapping("/{isbn}")
    public ResponseEntity<Book> get(@PathVariable("isbn") String isbn) {
        return bookService.find(isbn)
                .map(ResponseEntity::ok)
                .orElse(ResponseEntity.notFound().build());

    }

    @PostMapping
    public ResponseEntity<Book> create(@RequestBody Book book,
                                       UriComponentsBuilder uriBuilder) {
        Book created = bookService.create(book);
        URI newBookUri = uriBuilder.path("/books/{isbn}").build(created.getIsbn());
        return ResponseEntity
                .created(newBookUri)
                .body(created);
    }
}
```



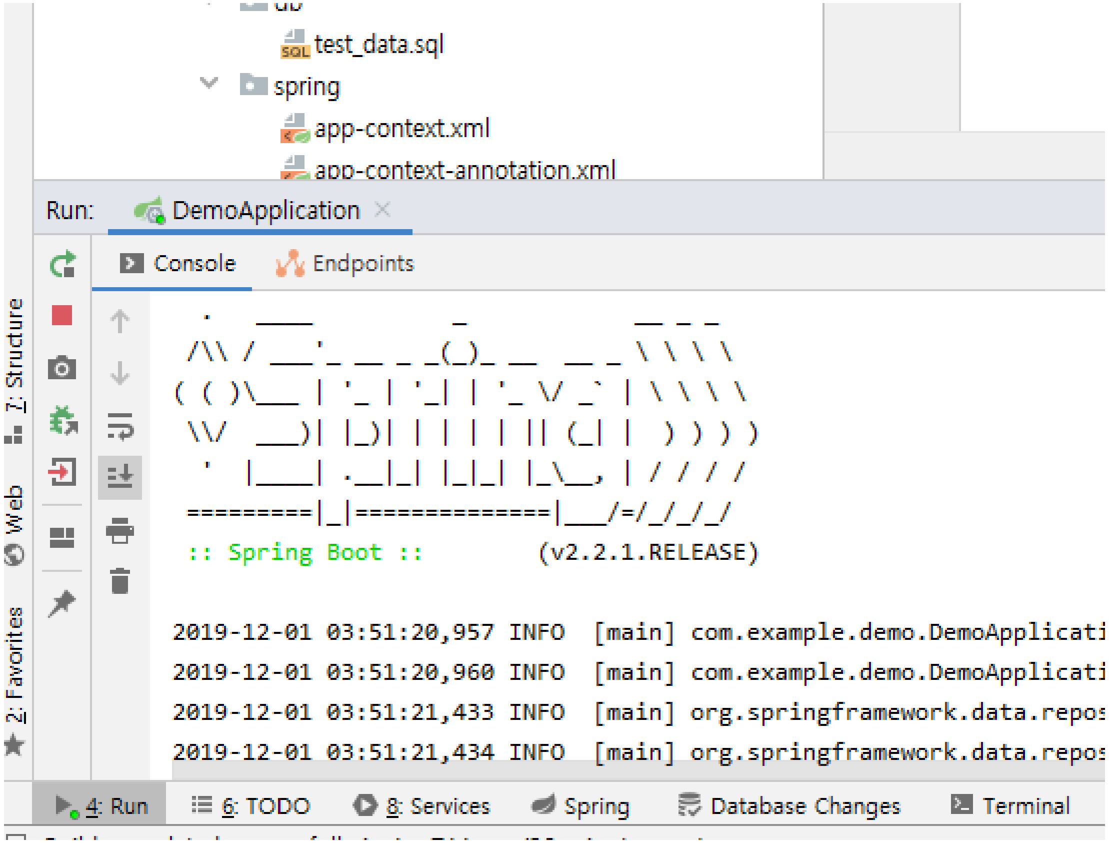
```java
package com.example.demo.library;

import java.util.Optional;

public interface BookService {
    Iterable<Book> findAll();
    Book create(Book book);
    Optional<Book> find(String isbn);
}
```



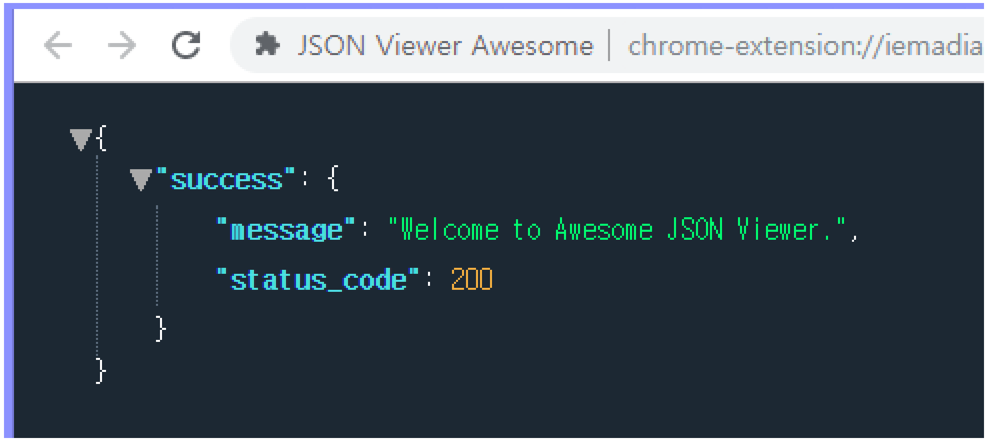
```java
package com.example.demo.library;

import org.springframework.stereotype.Service;

import java.util.Map;
import java.util.Optional;
import java.util.concurrent.ConcurrentHashMap;

@Service
class InMemoryBookService implements BookService {

    private final Map<String, Book> books = new ConcurrentHashMap<>();

    @Override
    public Iterable<Book> findAll() {
        return books.values();
    }

    @Override
    public Book create(Book book) {
        books.put(book.getIsbn(), book);
        return book;
    }

    @Override
    public Optional<Book> find(String isbn) {
        return Optional.ofNullable(books.get(isbn));
    }
}
```



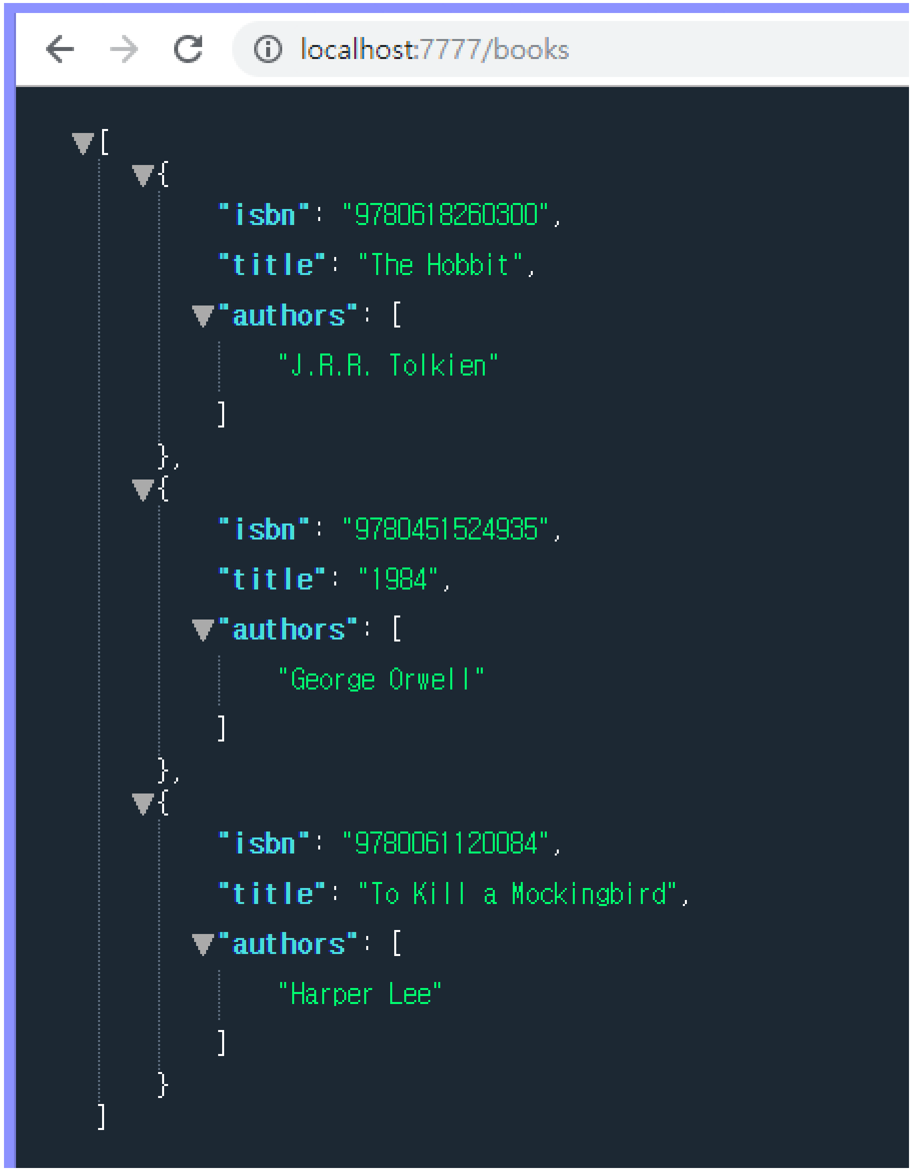
```java
package com.example.demo;

import com.example.demo.library.Book;
import com.example.demo.library.BookService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.ApplicationRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;

@SpringBootApplication
public class DemoApplication {
    private static Logger logger = LoggerFactory.getLogger(DemoApplication.class);

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

    @Bean
    public ApplicationRunner booksInitializer(BookService bookService) {
        return args -> {
            bookService.create(new Book( isbn: "9780061120084",  title: "To Kill a Mockingbird",  ...authors: "Harper Lee"));
            bookService.create(new Book( isbn: "9780451524935",  title: "1984",  ...authors: "George Orwell"));
            bookService.create(new Book( isbn: "9780618260300",  title: "The Hobbit",  ...authors: "J.R.R. Tolkien"));
        };

    }
}
```



```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.1.RELEASE)

2019-12-01 03:51:20,957 INFO  [main] com.example.demo.DemoApplicati
2019-12-01 03:51:20,960 INFO  [main] com.example.demo.DemoApplicati
2019-12-01 03:51:21,433 INFO  [main] org.springframework.data.repos
2019-12-01 03:51:21,434 INFO  [main] org.springframework.data.repos
```



```
{
  "success": {
    "message": "Welcome to Awesome JSON Viewer.",
    "status_code": 200
  }
}
```



localhost:7777/books

```
[
    {
        "isbn": "9780618260300",
        "title": "The Hobbit",
        "authors": [
            "J.R.R. Tolkien"
        ]
    },
    {
        "isbn": "9780451524935",
        "title": "1984",
        "authors": [
            "George Orwell"
        ]
    },
    {
        "isbn": "9780061120084",
        "title": "To Kill a Mockingbird",
        "authors": [
            "Harper Lee"
        ]
    }
]
```