**LAPORAN PRAKTIKUM PEMROGRAMAN PLATFORM KHUSUS**

**MODUL 3**

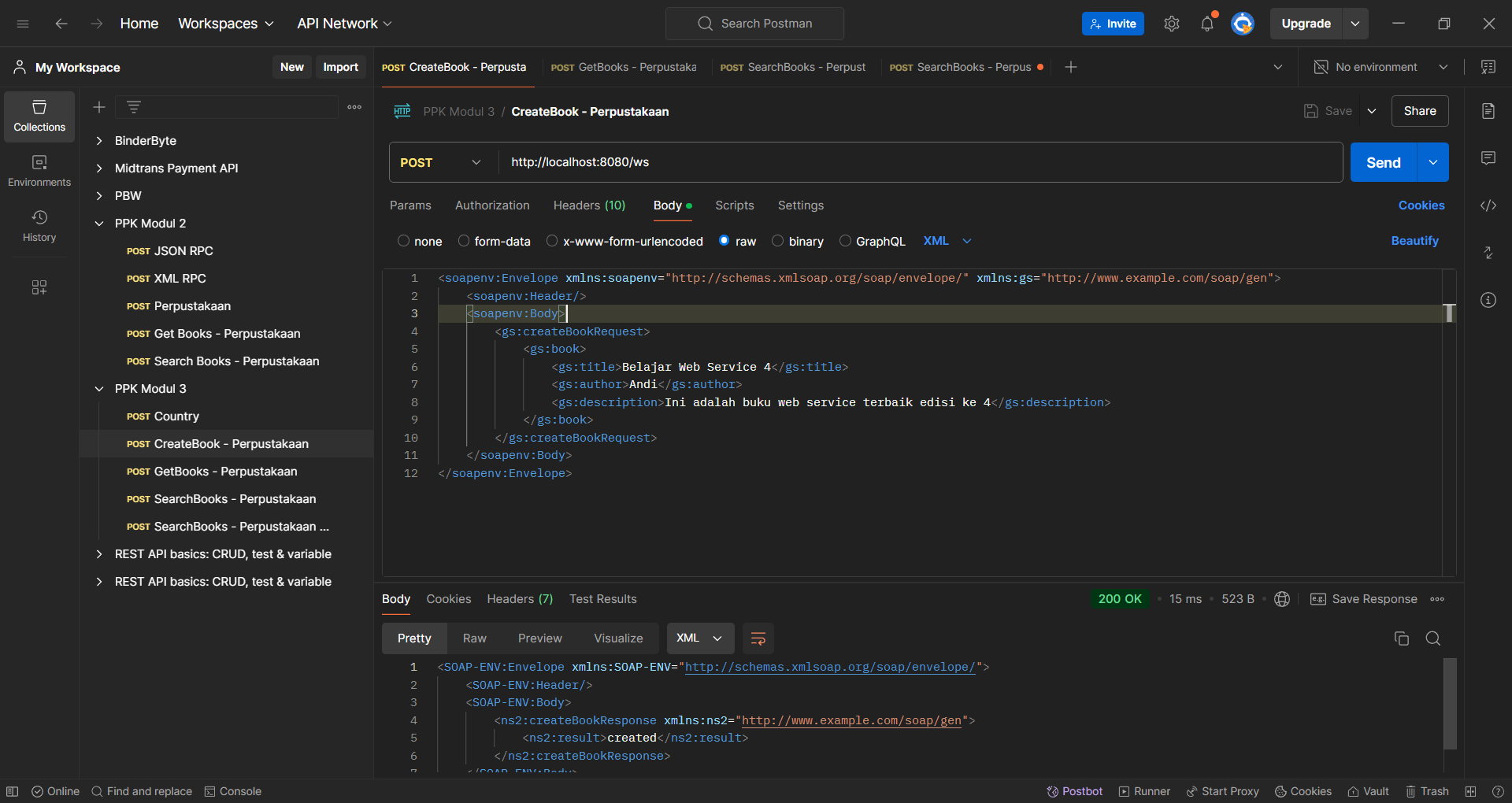
**SIMPLE OBJECT ACCESS PROTOCOL (SOAP)**

Ubahlah layanan perpustakaan pada modul praktikum 2 ke versi SOAP Web Service. Layanan yang dibuat adalah menambahkan koleksi buku dan mendapatkan semua koleksi buku. Lakukan pengujian untuk setiap layanan yang dibuat. Selanjutnya, buat laporan pekerjaan Anda dengan penjelasan lengkap dalam format pdf.

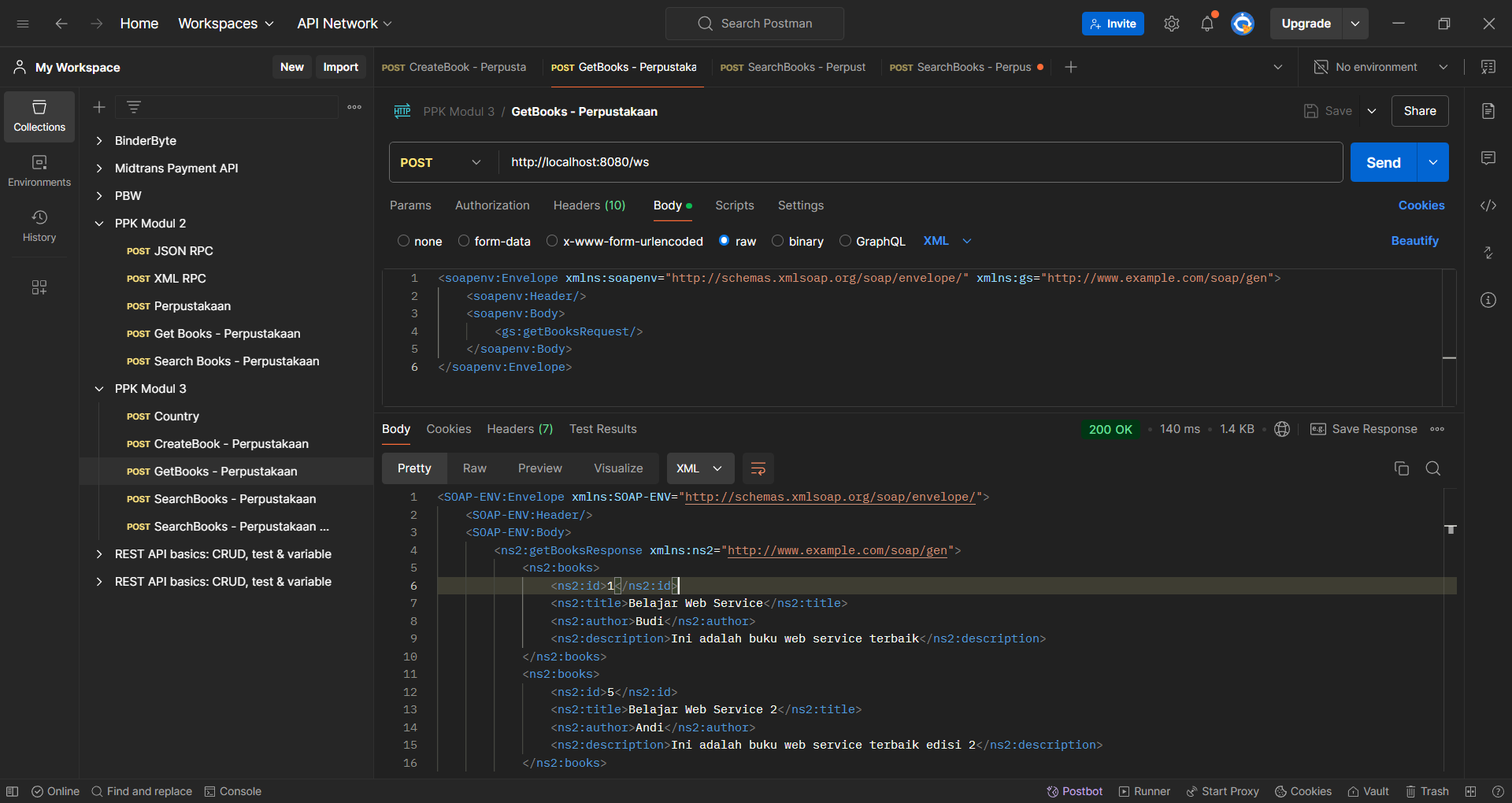
**Link Repository Git:** [**https://github.com/afrzl/Pemrograman-Platform-Khusus/tree/master/Pertemuan%203/perpustakaan**](https://github.com/afrzl/Pemrograman-Platform-Khusus/tree/master/Pertemuan%203/perpustakaan)

**POSTMAN**

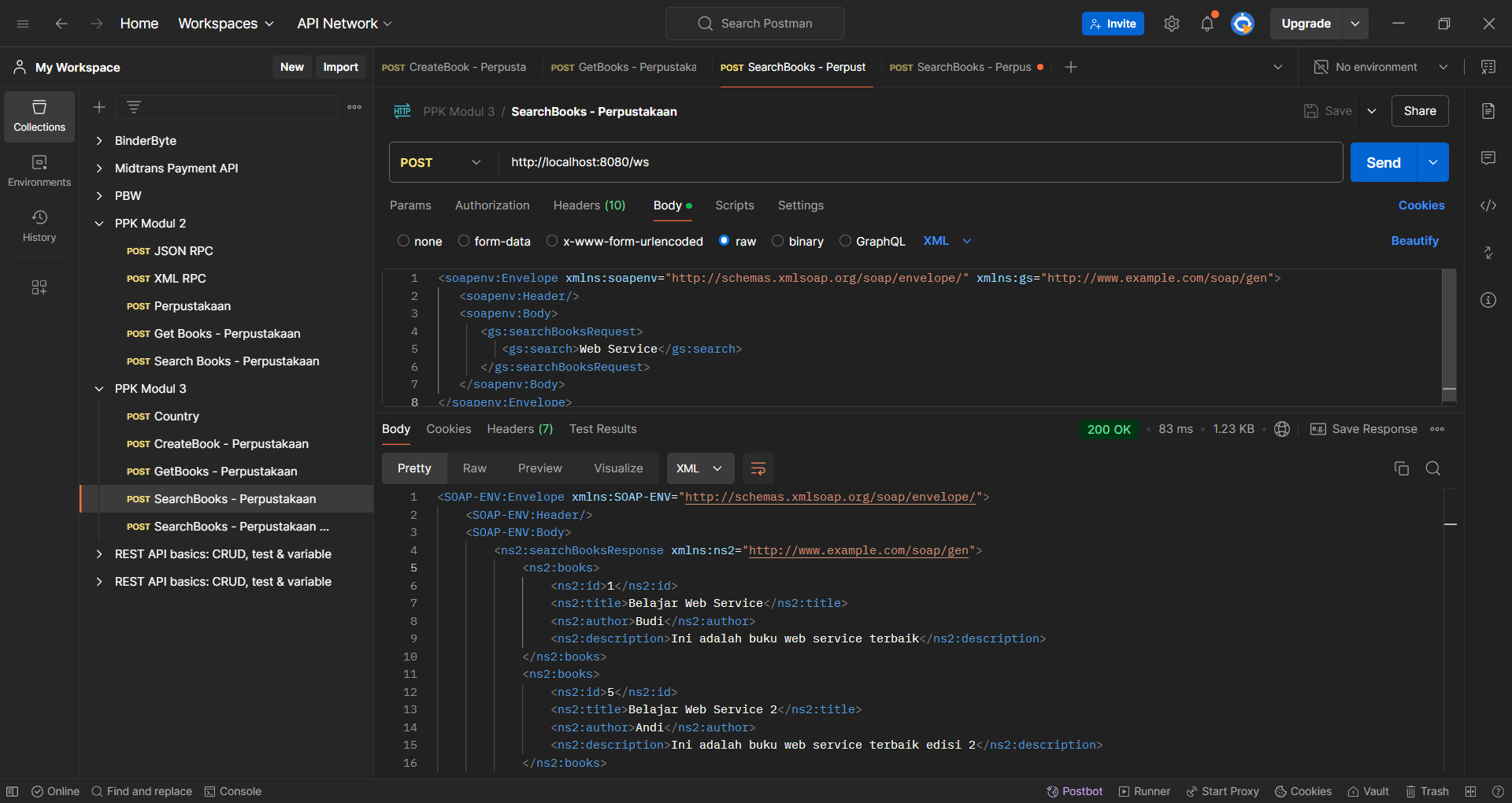
1. **CreateBook**

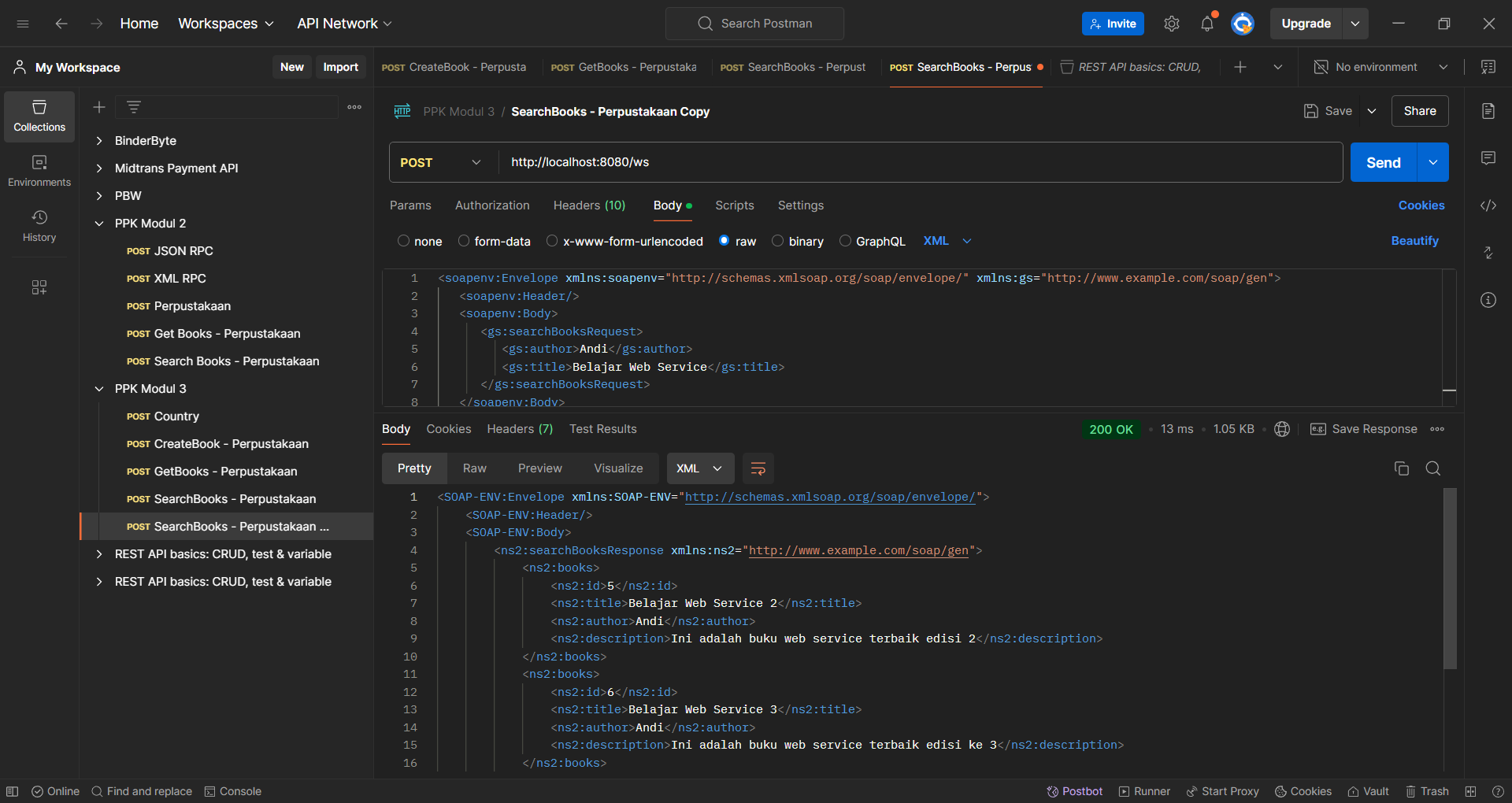
****

1. **GetBooks**

****

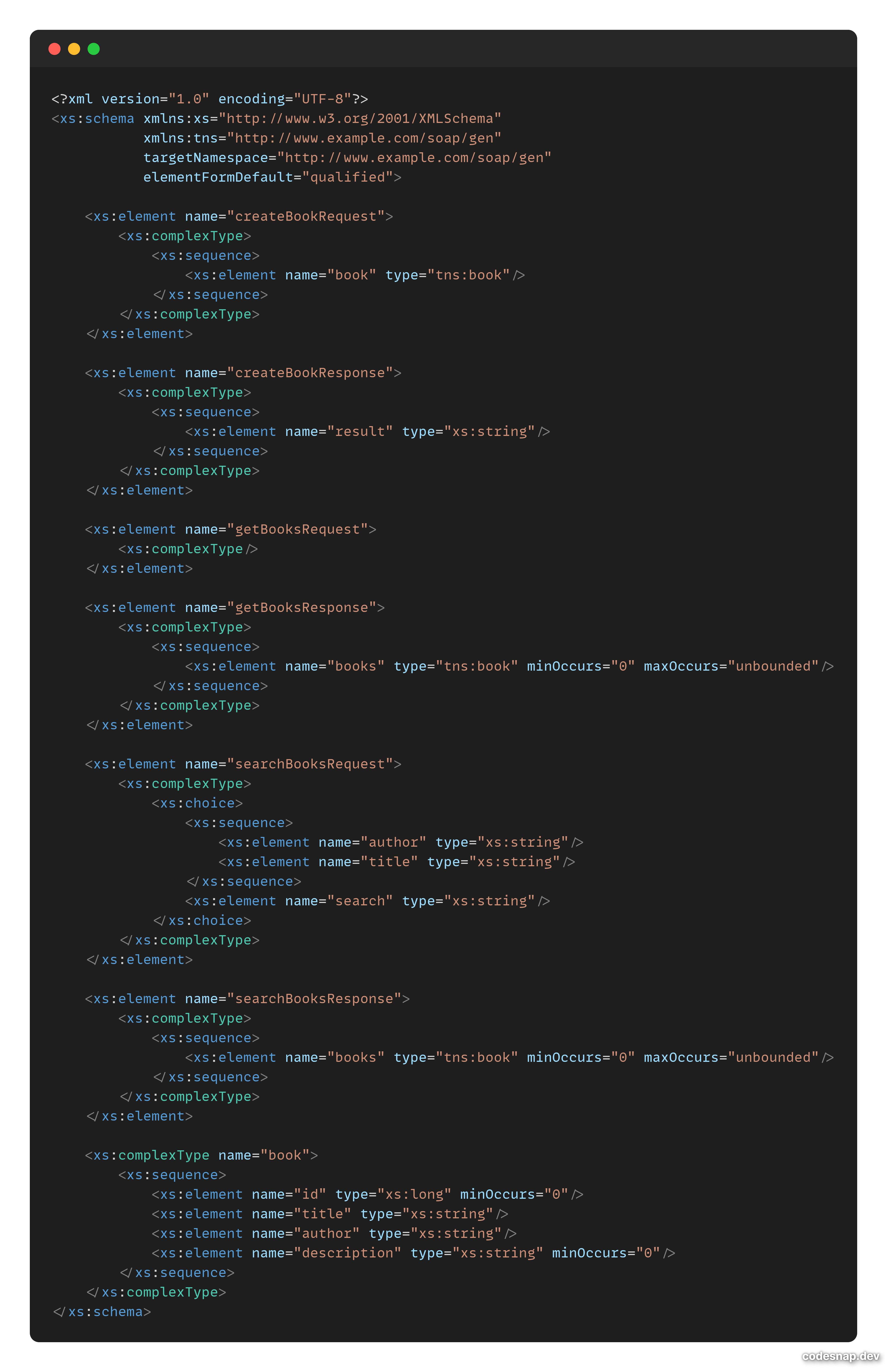
1. **SearchBooks**

****

****

**SOURCE CODE**

1. **books.xsd**

****

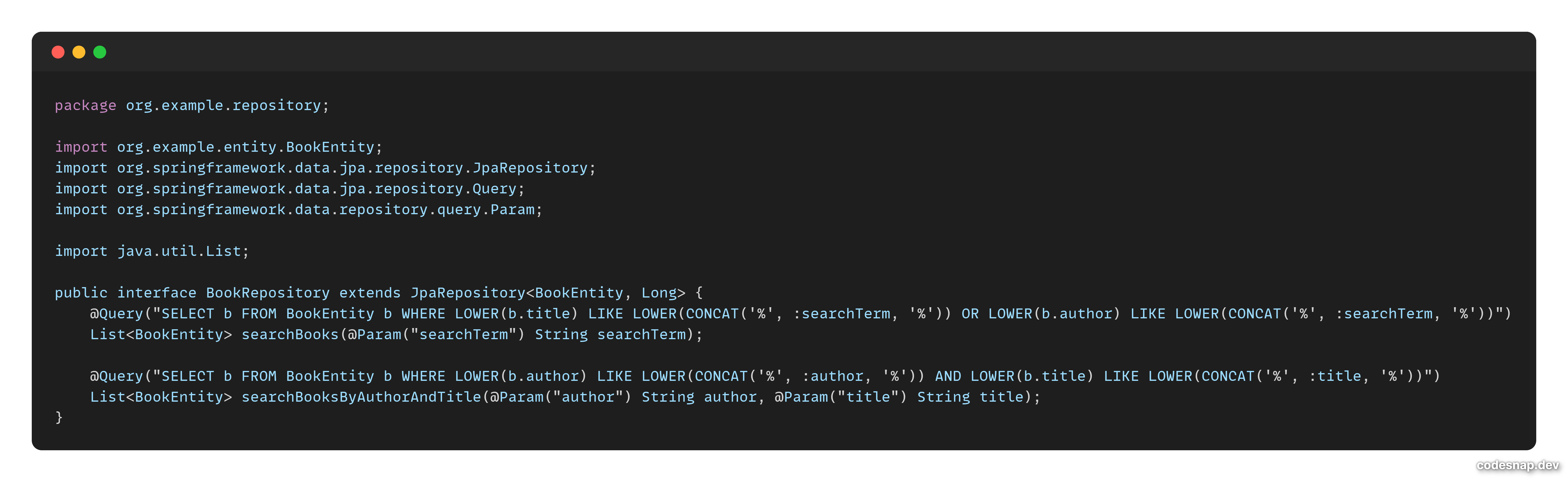
File XSD ini mendefinisikan struktur untuk layanan web SOAP terkait manajemen buku. Skema mencakup operasi untuk membuat, mengambil, dan mencari buku, serta mendefinisikan struktur data buku. Menggunakan namespace khusus, skema ini menyediakan blueprint yang jelas untuk implementasi dan penggunaan layanan web buku, memastikan konsistensi dalam pertukaran pesan SOAP.

1. **BookEntity (entity)**

****

Kelas BookEntity ini adalah entitas JPA yang merepresentasikan tabel 'books' dalam database. Memiliki empat atribut: id (kunci utama yang di-generate secara otomatis), title, author, dan description. Anotasi @Entity menandai kelas sebagai entitas JPA, sementara @Table menentukan nama tabel. Kelas ini menyediakan getter dan setter untuk semua atribut, memungkinkan akses dan modifikasi data buku dalam aplikasi.

1. **BookMapper (mapper)**

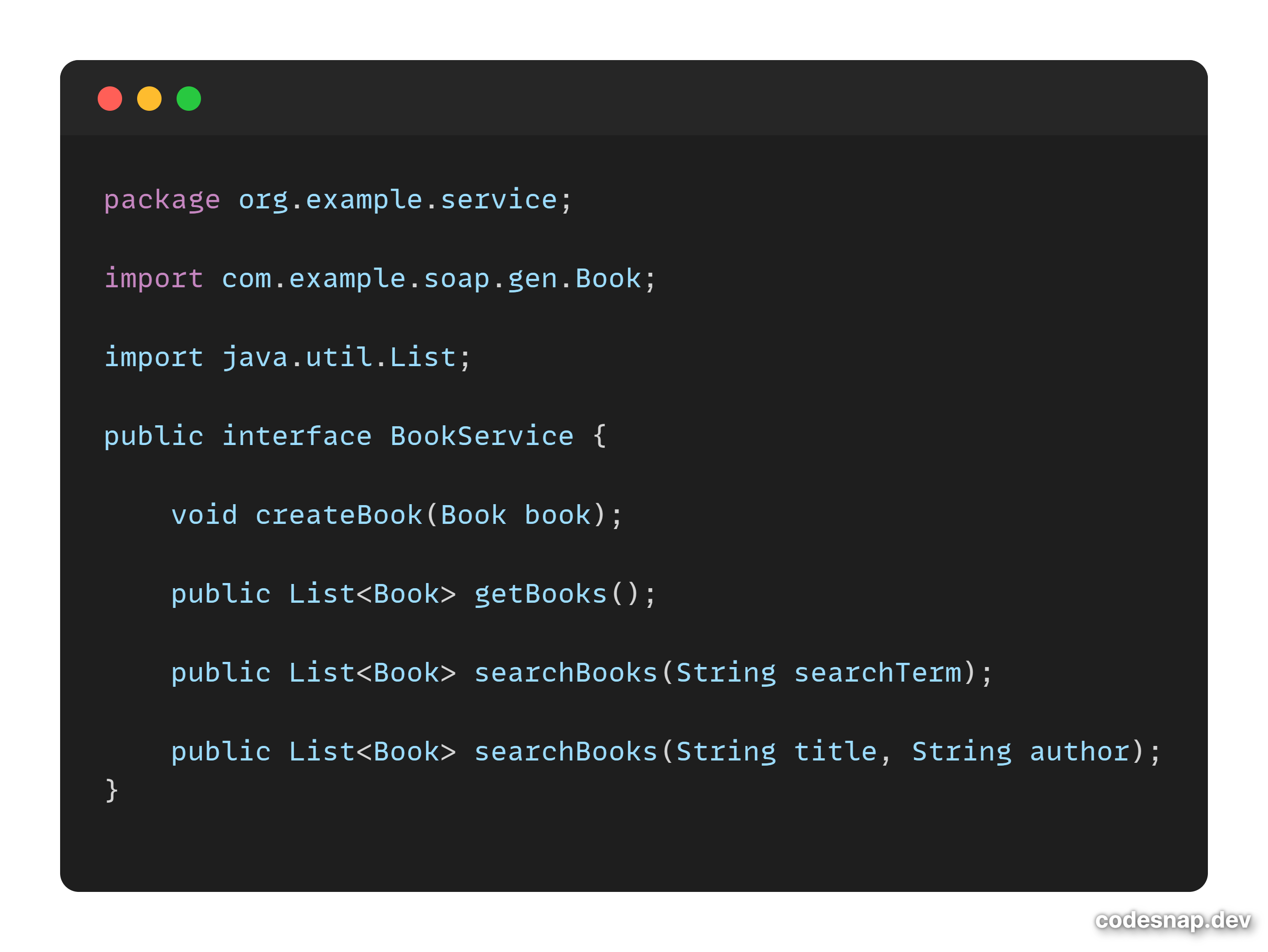


Interface BookRepository ini memperluas JpaRepository untuk entitas BookEntity, menyediakan operasi CRUD standar dan fungsionalitas pencarian khusus:

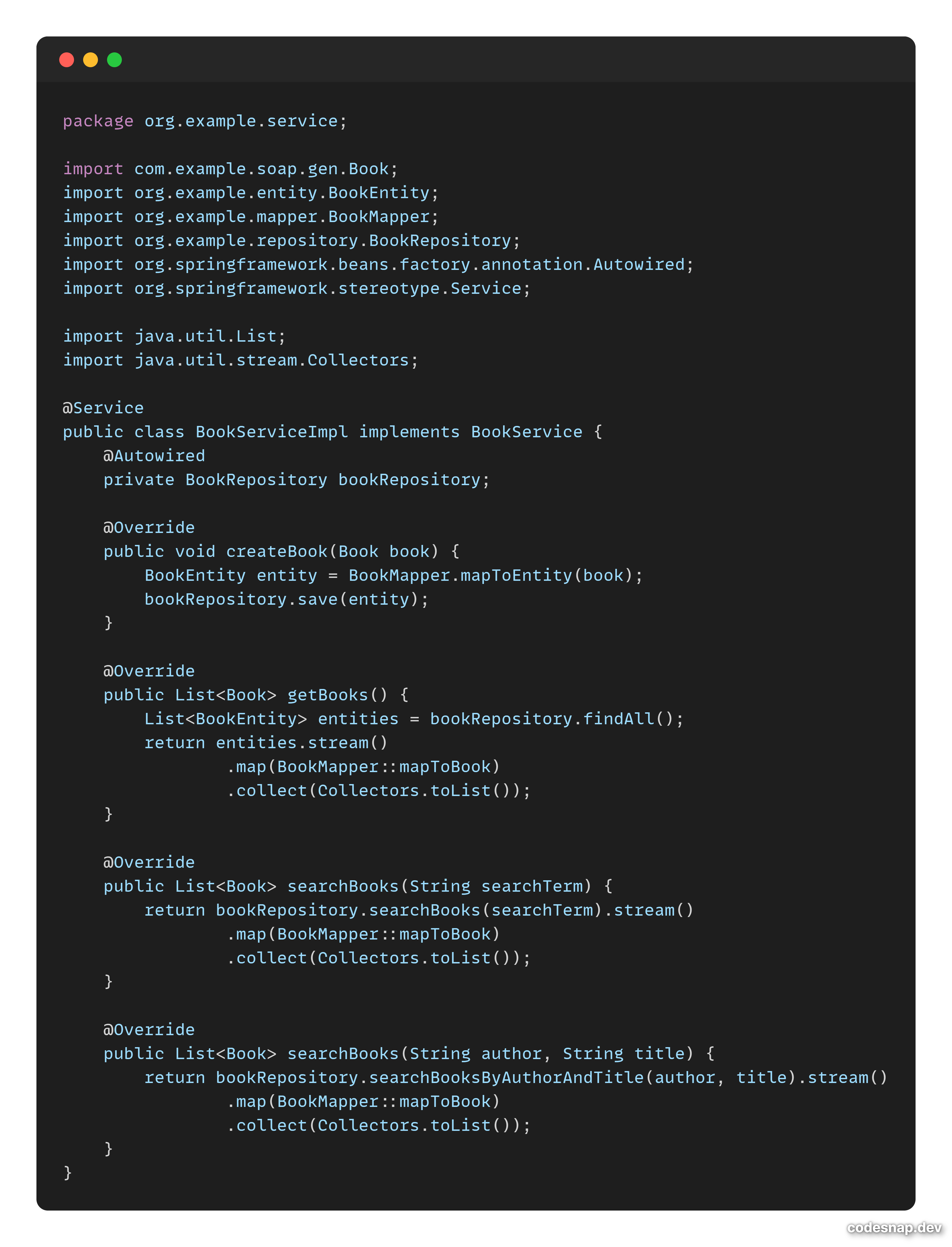
1. Mewarisi metode CRUD dasar dari JpaRepository untuk BookEntity.
2. searchBooks: Mencari buku berdasarkan kata kunci dalam judul atau penulis, menggunakan query JPQL kustom.
3. searchBooksByAuthorAndTitle: Mencari buku berdasarkan penulis dan judul secara bersamaan.

Kedua metode pencarian menggunakan anotasi @Query dengan JPQL untuk mendefinisikan logika pencarian yang fleksibel, memungkinkan pencarian case-insensitive dengan operator LIKE.

1. **BookService (service)**

****

1. **BookServiceImpl (service)**

****

Kelas BookServiceImpl ini mengimplementasikan BookService, menyediakan logika bisnis untuk operasi buku:

1. Menggunakan BookRepository untuk akses data.
2. createBook: Menyimpan buku baru ke database.
3. getBooks: Mengambil semua buku dari database.
4. searchBooks(String): Mencari buku berdasarkan kata kunci.
5. searchBooks(String, String): Mencari buku berdasarkan penulis dan judul.

Semua metode menggunakan BookMapper untuk konversi antara entitas database (BookEntity) dan objek SOAP (Book). Kelas ini menggunakan Stream API untuk transformasi data.

1. **BookEndpoint**

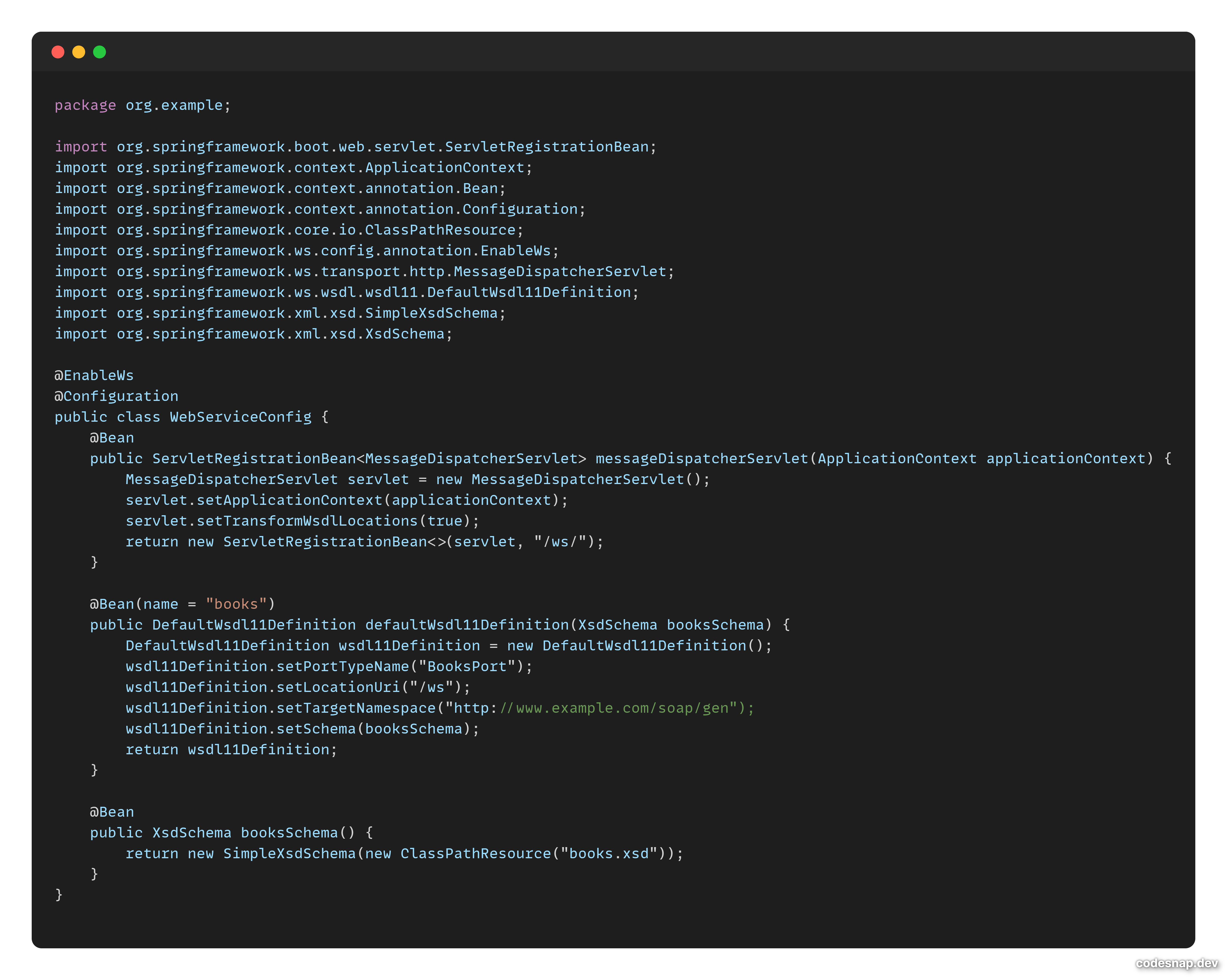
****

Kelas BookEndpoint ini adalah endpoint SOAP yang menangani permintaan terkait buku:

1. Dianotasi dengan @Endpoint untuk dikenali sebagai endpoint SOAP.
2. Menggunakan BookService untuk operasi bisnis.
3. Memiliki tiga metode yang masing-masing menangani operasi berbeda:
   * createBook: Membuat buku baru.
   * getBooks: Mengambil semua buku.
   * searchBooks: Mencari buku berdasarkan kriteria.
4. Setiap metode dianotasi dengan @PayloadRoot untuk menentukan namespace dan elemen lokal yang ditangani.
5. Menggunakan @RequestPayload untuk menerima data permintaan dan @ResponsePayload untuk mengirim respons.
6. Metode searchBooks menangani dua jenis pencarian: dengan kata kunci tunggal atau dengan penulis dan judul.

Kelas ini berfungsi sebagai penghubung antara permintaan SOAP dan logika bisnis, mengonversi data antara format SOAP dan objek Java.

1. **WebServiceConfig**

****

Kelas WebServiceConfig ini mengkonfigurasi layanan web SOAP:

1. Dianotasi dengan @EnableWs dan @Configuration untuk mengaktifkan dukungan Spring WS.
2. Mendefinisikan MessageDispatcherServlet untuk menangani permintaan SOAP.
3. Mengkonfigurasi WSDL dengan DefaultWsdl11Definition, termasuk nama port, URI lokasi, dan namespace target.
4. Menyediakan skema XSD untuk layanan web melalui metode booksSchema().

Konfigurasi ini memungkinkan Spring Boot untuk mengekspos endpoint SOAP dan menghasilkan WSDL berdasarkan skema XSD yang didefinisikan, memudahkan integrasi dengan klien SOAP.