// Book.java

@Table("book")

public class Book {

@Id

private Long id;

@Column("title")

private String title;

@Column("author")

private String author;

@Column("published\_date")

private LocalDate publishedDate;

// getters and setters

}

// BookRepository.java

public interface BookRepository extends ReactiveCrudRepository<Book, Long>, BookRepositoryCustom {

}

// BookRepositoryCustom.java

public interface BookRepositoryCustom {

Flux<Book> findByCriteria(BookSearchCriteria criteria);

}

// BookSearchCriteria.java

public class BookSearchCriteria {

private String title;

private String author;

private LocalDate publishedDate;

// getters and setters

// constructor, builder, or any other convenient method to create instances

}

// BookRepositoryCustomImpl.java

public class BookRepositoryCustomImpl implements BookRepositoryCustom {

private final R2dbcEntityOperations entityOperations;

public BookRepositoryCustomImpl(R2dbcEntityOperations entityOperations) {

this.entityOperations = entityOperations;

}

@Override

public Flux<Book> findByCriteria(BookSearchCriteria criteria) {

Criteria criteriaObject = Criteria.empty();

if (criteria.getTitle() != null) {

criteriaObject = criteriaObject.and("title").is(criteria.getTitle());

}

if (criteria.getAuthor() != null) {

criteriaObject = criteriaObject.and("author").is(criteria.getAuthor());

}

if (criteria.getPublishedDate() != null) {

criteriaObject = criteriaObject.and("published\_date").is(criteria.getPublishedDate());

}

return entityOperations.select(Book.class)

.matching(Query.query(criteriaObject))

.all();

}

}

// BookService.java or BookController.java

@Service

public class BookService {

private final BookRepository bookRepository;

public BookService(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public Flux<Book> searchBooks(BookSearchCriteria criteria) {

return bookRepository.findByCriteria(criteria);

}

}