

Coverage for **app/routers/book\_router.py**: 56%

34 statements   19 run   15 missing   0 excluded

« prev   ^ index   » next   coverage.py v7.11.3, created at 2025-11-19 02:12 +0000

```
1 from fastapi import APIRouter, HTTPException
2 from app.schemas.book import BookCreate, BookUpdate
3 from app.services.book_service import BookService
4
5 router = APIRouter(prefix="/books", tags=["Books"])
6 service = BookService()
7
8 @router.post("/")
9 def create_book(book: BookCreate):
10     try:
11         return service.create_book(book)
12     except ValueError as e:
13         raise HTTPException(status_code=400, detail=str(e))
14
15 @router.get("/")
16 def get_all_books():
17     """Retrieve a list of books."""
18     return service.get_all_books()
19
20 @router.get("/{isbn}")
21 def get_book(isbn: str):
22     """Retrieve a specific book by ISBN."""
23     book = service.get_book(isbn)
24     if not book:
25         raise HTTPException(status_code=404, detail="Book not found")
26     return book
27
28 @router.put("/{isbn}")
29 def update_book(isbn: str, book: BookUpdate):
30     """Update an existing book's details."""
```

```
31     try:
32         updated_book = service.update_book(isbn, book)
33         if not updated_book:
34             raise HTTPException(status_code=404, detail="Book not found")
35         return updated_book
36     except ValueError as e:
37         raise HTTPException(status_code=400, detail=str(e))
38
39 @router.delete("/{isbn}")
40 def delete_book(isbn: str):
41     if not service.delete_book(isbn):
42         raise HTTPException(status_code=404, detail="Book not found")
43     return {"message": "Book deleted successfully"}
```

« prev ^ index » next coverage.py v7.11.3, created at 2025-11-19 02:12 +0000