

Coverage for **app/models/readinglist.py**: 87%



39 statements

34 run

5 missing

0 excluded

« prev ^ index » next coverage.py v7.11.3, created at 2025-11-20 19:17 -0800

```
1 from datetime import datetime
2 from typing import List, Dict
3 from app.schemas.book import BookItem
4 from app.utils.data_manager import CSVRepository
5
6 class ReadingList:
7     def __init__(self, list_id: int, user_id: int, name: str, books: List[BookItem] = None, is_public: bool = False):
8         self.list_id = list_id
9         self.user_id = user_id
10        self.name = name
11        self.books = books or []
12        self.is_public = is_public
13
14    def to_csv_dict(self) -> dict:
15
16        return {
17            "ListID": self.list_id,
18            "UserID": self.user_id,
19            "Name": self.name,
20            "ISBNs": "|".join(self.books) if self.books else "",
21            "IsPublic": "true" if self.is_public else "false",
22        }
23
24
25    @classmethod
26    def from_dict(cls, row: Dict[str, str]) -> "ReadingList":
27        return cls(
28            list_id = int(row["ListID"]),
29            user_id = int(row["UserID"]),
30            name = row["Name"],
31            books = row.get("ISBNs", "").split("|") if row.get("ISBNs") else [],
32            is_public=row.get("IsPublic", "false") == "true"
33        )
34
```

```

35     def to_api_dict(self) -> dict:
36         book_info_list = self.__get_book_info()
37         return {
38             "list_id": self.list_id,
39             "user_id": self.user_id,
40             "name": self.name,
41             "books": book_info_list,
42             "is_public": self.is_public,
43         }
44
45     def __get_book_info(self):
46         books_path = "app/data/Books.csv"
47         repo = CSVRepository()
48         all_books = repo.read_all(books_path)
49         book_lookup = {row["ISBN"]: row for row in all_books}
50
51         returned_books = []
52         for isbn in self.books:
53             book_data = book_lookup.get(isbn)
54             if book_data:
55                 returned_books.append({
56                     "isbn": isbn,
57                     "book_title": book_data.get("Book-Title", "Unknown Title"),
58                     "author": book_data.get("Book-Author", "Unknown Author")
59                 })
60             else:
61                 returned_books.append({
62                     "isbn": isbn,
63                     "book_title": "Unknown Title",
64                     "author": "Unknown Author"
65                 })
66         return returned_books
67
68     def rename(self, new_name: str):
69         self.name = new_name
70
71     def add_book(self, isbn: str):
72         self.books.append(isbn)
73
74     def remove_book(self, isbn: str):
75         self.books = [b for b in self.books if b != isbn]

```

76

77

78

79

```
def total_books(self) -> int:
```

```
    return len(self.books)
```

« prev ^ index » next coverage.py v7.11.3, created at 2025-11-20 19:17 -0800