

Coverage for app/models/readinglist.py: 96%



25 statements 24 run 1 missing 0 excluded

« prev ^ index » next coverage.py v7.11.3, created at 2025-11-21 23:38 -0800

```
1 from datetime import datetime
2 from typing import List, Dict
3 from app.schemas.book import BookItem
4
5 class ReadingList:
6     def __init__(self, list_id: int, user_id: int, name: str, books: List[BookItem] = None, is_public: bool = False)
7         self.list_id = list_id
8         self.user_id = user_id
9         self.name = name
10        self.books = books or []
11        self.is_public = is_public
12
13    def to_csv_dict(self) -> dict:
14
15        return {
16            "ListID": self.list_id,
17            "UserID": self.user_id,
18            "Name": self.name,
19            "ISBNs": "|".join(self.books) if self.books else "",
20            "IsPublic": "true" if self.is_public else "false",
21        }
22
23
24    @classmethod
25    def from_dict(cls, row: Dict[str, str]) -> "ReadingList":
26        return cls(
27            list_id = int(row["ListID"]),
28            user_id = int(row["UserID"]),
29            name = row["Name"],
30            books = row.get("ISBNs", "").split("|") if row.get("ISBNs") else [],
31            is_public=row.get("IsPublic", "false") == "true"
32        )
33
34    def to_api_dict(self, book_info) -> dict:
35        return {
36            "list_id": self.list_id,
37            "user_id": self.user_id,
38            "name": self.name,
39            "books": book_info,
40            "is_public": self.is_public,
41        }
42
43    def rename(self, new_name: str):
44        self.name = new_name
45
46    def add_book(self, isbn: str):
47        self.books.append(isbn)
48
49    def remove_book(self, isbn: str):
50        self.books = [b for b in self.books if b != isbn]
51
52    def total_books(self) -> int:
53        return len(self.books)
```

« prev ^ index » next coverage.py v7.11.3, created at 2025-11-21 23:38 -0800