app/services/readinglist_service.py: 71% 44 18 0

```python
from pathlib import Path
from app.models.request import Request
from app.schemas.readinglist import ReadingListCreate, ReadingListDetail, ReadingListSummary
from app.models.readinglist import ReadingList
from app.utils.data_manager import CSVRepository

class ReadingListService:
    def __init__(self):
        self.repo = CSVRepository()
        self.path = Path(__file__).resolve().parents[1] / "data" / "ReadingLists.csv"
        self.fields = ["ListID", "UserID", "Name", "ISBNs"]

    def __generate_next_id(self) -> int:
        rows = self.repo.read_all(self.path)
        if not rows:
            return 1
        ids = [int(r["ListID"]) for r in rows if r["ListID"].isdigit()]
        return max(ids, default=0) + 1

    def __already_added(self, list_id: int, isbn: str) -> bool:
        rows = self.repo.read_all(self.path)

        found = False
        for row in rows:
            if row["ListID"] == str(list_id):
                books = row["ISBNs"].split("|") if row["ISBNs"] else []
                if isbn in books:
                    found = True
                    break
        return found

    def __number_of_readinglist(self, user_id)-> int:
```

app/services/readinglist_service.py: 71%

```
        for row in rows:
            if str(user_id) == row["UserID"]:
                count += 1

        return count

    def get_all_readinglist(self, user_id: int):
        rows = self.repo.read_all(self.path)
        result = []
        for r in rows:
            if r["UserID"] == str(user_id):
                result.append(r)
        return result

    def create_list(self, data: ReadingListCreate, user_id: int) -> ReadingListDetail:
        next_id = self.__generate_next_id()
        num_readinglist = self.__number_of_readinglist(user_id=user_id)
        if num_readinglist > 10:
            raise ValueError("You can only have 10 reading lists")

        rows = self.get_all_readinglist(user_id=user_id)
        for r in rows:
            if r["Name"].lower() == data.name.lower():
                raise ValueError(f'A reading list named "{data.name}" already exists.')

        readinglist = ReadingList(list_id=next_id,
                                  user_id=user_id,
                                  name=data.name)

        self.repo.append_row(self.path, self.fields, readinglist.to_csv_dict())
        return ReadingListDetail(**readinglist.to_api_dict())

    def delete_list(self, list_id: int, user_id: int):

        rows = self.repo.read_all(self.path)
        updated_rows = [
            r for r in rows if not (int(r["ListID"]) == list_id and int(r["UserID"]) == user_id)
        ]
        if len(updated_rows) == original_count:
            return False

        for i, row in enumerate(updated_rows, start=1):
            row["ListID"] = str(i)
        self.repo.write_all(self.path, self.fields, updated_rows)
        return True
```

« prev ^ index » next

coverage.py v7.11.3, created at 2025-11-19 02:12 +0000