| **id** | **Название** | **Описание** | **Приоритет** | **Предусловия** | **Шаги** | **Ожидаемый результат** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Получение всего списка книг | При получении всего списка книг статус запроса должен быть 200, тело ответа должно соответствовать JSON схеме | Высокий | - | 1. Отправить GET запрос на http://localhost:5000/api/books | 1. Статус запроса 200, валидация JSON схемы |
| 2 | Получение книги по несуществующему id | При получении книги по несуществующему id статус запроса должен быть 404, в ответе должна быть описана ошибка | Низкий | - | 1. Отправить GET запрос на http://localhost:5000/api/books/{id}, где id - номер несуществующей книги | 1. Статус запроса 404, ошибка "Book with id {id} not found" |
| 3 | Получение книги | При получении книги статус запроса должен быть 200, тело ответа должно соответствовать JSON схеме | Высокий | - | 1. Отправить GET запрос на http://localhost:5000/api/books/{id}, где id - номер книги | 1. Статус запроса 200, валидация JSON схемы |
| 4 | Удаление книги по несуществующему id | При удалении книги по несуществующему id статус запроса должен быть 404, в ответе должна быть описана ошибка | Низкий | - | 1. Отправить DELETE запрос на http://localhost:5000/api/books/{id}, где id - номер несуществующей книги | 1. Статус запроса 404, ошибка "Book with id {id} not found" |
| 5 | Удаление существующей книги | При удалении существующей книги статус запроса должен быть 200, тело ответа должно содержать сообщение об успешном удалении книги | Высокий | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить DELETE запрос на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Статус запроса 200, тело ответа содержит "result":true |
| 6 | Получение всего списка книг после удаления книги | При получении всего списка книг после удаления книги тело ответа должно уменьшиться на 1 | Средний | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить GET запрос на http://localhost:5000/api/books  2. Отправить DELETE запрос на http://localhost:5000/api/books/{id}, где id - номер созданной книги 3. Отправить GET запрос на http://localhost:5000/api/books | 1. Определяется размер тела ответа  2. Книга удаляется 3. Размер тела ответа уменьшaется на 1 |
| 7 | Получение удаленной книги | При получении удаленной книги статус запроса должен быть 404, в ответе должна быть описана ошибка | Средний | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить DELETE запрос на http://localhost:5000/api/books/{id}, где id - номер созданной книги  2. Отправить GET запрос на http://localhost:5000/api/books/{id}, где id - номер удаленной книги | 1. Книга удаляется  2. Статус запроса 404, ошибка "Book with id {id} not found" |
| 8 | Создание новой книги после удаления | После удаления книги новые созданные книги должны продолжать брать id, начиная от последней книги. Удаленный из середины списка id должен оставаться незанятым | Средний | Создать несколько книг с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить DELETE запрос на http://localhost:5000/api/books/{id}, где id - номер созданной книги не последней в списке  2. Отправить POST запрос на http://localhost:5000/api/books | 1. Книга удаляется  2. Id созданной книги не совпадает с удаленным из середины списка id |
| 9 | Получение всего списка книг после создания книги | При получение всего списка книг после создания книги данные последней книги из тела ответа должны совпадать с данными созданной книги | Средний | - | 1. Отправить GET запрос на http://localhost:5000/api/books  2. Отправить POST запрос на http://localhost:5000/api/books  3. Отправить GET запрос на http://localhost:5000/api/books | 1. Определяется размер тела ответа  2. Книга создается 3. Данные последней книги из тела ответа (id, name, author, year, isElectronicBook) совпадают с данными созданной книги, размер тела ответа увеличивается на 1 |
| 10 | Получение созданной книги | При получении созданной книги данные из тела ответа должны совпадать с данными созданной книги | Средний | - | 1. Отправить POST запрос на http://localhost:5000/api/books  2. Отправить GET запрос на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Книга создается. Данные из тела ответа (id, name, author, year, isElectronicBook) совпадают с данными созданной книги |
| 11 | Создание книги | При создании книги данные из тела ответа должны совпадать с данными созданной книги, статус запрос должен быть 201 | Высокий | Отправить GET запрос на http://localhost:5000/api/books и узнать последнее id | 1. Создать книгу с помощью запроса POST на http://localhost:5000/api/books c заполненными полями id, name, author, year, isElectronicBook | 1. Данные из тела ответа (name, author, year, isElectronicBook) совпадают с данными созданной книги, id увеличился на 1, статус запроса 201, валидация JSON схемы |
| 12 | Создание книги с единственным заполненным полем name | При создании книги данные из тела ответа должны совпадать с данными созданной книги, статус запроса должен быть 201 | Высокий | Отправить GET запрос на http://localhost:5000/api/books и узнать последнее id | 1. Создать книгу с помощью запроса POST на http://localhost:5000/api/books c заполненным валидным полем name | 1. Данные из тела ответа: name совпадает с данными созданной книги, id увеличился на 1, author, year, isElectronicBook имеют значения по-умолчанию ("", 0, false, соответственно); статус запроса 201, валидация JSON схемы |
| 13 | Создание книги с невалидным полем name | При создании книги с невалидным полем name статус запроса должен быть 400, тело ответа должно содержать сообщение об ошибке | Низкий | - | 1. Создать книгу с помощью запроса POST на http://localhost:5000/api/books c заполненными полем name: "", 1, true, null | 1. Статус запроса 400, ошибка "Name must be String type (Unicode)" |
| 14 | Создание книги с невалидным полем author | При создании книги с невалидным полем author статус запроса должен быть 400, тело ответа должно содержать сообщение об ошибке | Низкий | - | 1. Создать книгу с помощью запроса POST на http://localhost:5000/api/books c валидным полем name и полем author: 1, true, null | 1. Статус запроса 400, ошибка "Author must be String type (Unicode)" |
| 15 | Создание книги с невалидным полем year | При создании книги с невалидным полем year статус запроса должен быть 400, тело ответа должно содержать сообщение об ошибке | Низкий | - | 1. Создать книгу с помощью запроса POST на http://localhost:5000/api/books c валидными полями name и author, с полем year: "year", true, null | 1. Статус запроса 400, ошибка "Year must be Int type" |
| 16 | Создание книги с невалидным полем isElectronicBookErrorMessage | При создании книги с невалидным полем isElectronicBookErrorMessage статус запроса должен быть 400, тело ответа должно содержать сообщение об ошибке | Низкий | - | 1. Создать книгу с помощью запроса POST на http://localhost:5000/api/books c валидными полями name, author, year и с полем isElectronicBookErrorMessage: "isElectronicBookErrorMessage", 1, null | 1. Статус запроса 400, ошибка "IsElectronicBook must be Bool type" |
| 17 | Создание книги с отсутствующем полем name | При создании книги с отсутствующим полем name статус запроса должен быть 400, тело ответа должно содержать сообщение об ошибке | Средний | - | 1. Создать книгу с помощью запроса POST на http://localhost:5000/api/books c валидным полем author | 1. Статус запроса 400, ошибка "Name is required" |
| 18 | Создание книги с отсутствующем телом запроса | При создании книги с отсутствующим телом запроса статус запроса должен быть 400, тело ответа должно содержать сообщение об ошибке | Низкий | - | 1. Создать книгу с помощью запроса POST на http://localhost:5000/api/books c пустым телом запроса | 1. Статус запроса 400, ошибка "Name is required" |
| 19 | Редактирование книги по несуществующему id | При редактировании книги по несуществующему id статус запроса должен быть 404, в ответе должна быть описана ошибка | Низкий | - | 1. Отправить PUT запрос на http://localhost:5000/api/books/{id}, где id - номер несуществующей книги | 1. Статус запроса 404, ошибка "Book with id {id} not found" |
| 20 | Редактирование книги с отсутствующем полем author | При редактировании книги с отсутствующем полем author статус запроса должен быть 400, в ответе должна быть описана ошибка | Средний | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить PUT запрос c валидным полем name на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Статус запроса 400, ошибка "Author is required" |
| 21 | Редактирование книги с отсутствующем полем name | При редактировании книги с отсутствующем полем name статус запроса должен быть 400, в ответе должна быть описана ошибка | Средний | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить PUT запрос c валидным полем author на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Статус запроса 400, ошибка "Name is required" |
| 22 | Редактирование книги с отсутствующем полем year | При редактировании книги с отсутствующем полем year статус запроса должен быть 400, в ответе должна быть описана ошибка | Средний | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить PUT запрос c валидными полями name, author на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Статус запроса 400, ошибка "Year is required" |
| 23 | Редактирование книги с отсутствующем полем isElectronicBook | При редактировании книги с отсутствующем полем isElectronicBook статус запроса должен быть 400, в ответе должна быть описана ошибка | Средний | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить PUT запрос c валидными полями name, author, year на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Статус запроса 400, ошибка "IsElectronicBook is required" |
| 24 | Редактирование книги | При редактировании книги данные из тела ответа должны совпадать с данными отредактированной книги, статус запрос должен быть 200 | Высокий | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить PUT запрос c валидными полями name, author, year, isElectronicBook на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Данные из тела ответа (name, author, year, isElectronicBook) совпадают с данными отредактированной книги, статус запроса 200, валидация JSON схемы |
| 25 | Редактирование книги с невалидным полем name | При редактировании книги с невалидным полем name статус запрос должен быть 400, тело ответа должно содержать сообщение об ошибке | Низкий | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить PUT запрос c невалидным полем name ("", 1, true, null) на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Статус запроса 400, ошибка "Name must be String type (Unicode)" |
| 26 | Редактирование книги с невалидным полем author | При редактировании книги с невалидным полем author статус запрос должен быть 400, тело ответа должно содержать сообщение об ошибке | Низкий | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить PUT запрос c валидным полем name и невалидным полем author (1, true, null) на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Статус запроса 400, ошибка "Author must be String type (Unicode)" |
| 27 | Редактирование книги с невалидным полем year | При редактировании книги с невалидным полем year статус запрос должен быть 400, тело ответа должно содержать сообщение об ошибке | Низкий | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить PUT запрос c валидными полями name, author и невалидным полем year ("year", true, null) на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Статус запроса 400, ошибка "Year must be Int type" |
| 28 | Редактирование книги с невалидным полем isElectronicBook | При редактировании книги с невалидным полем isElectronicBook статус запрос должен быть 400, тело ответа должно содержать сообщение об ошибке | Низкий | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1. Отправить PUT запрос c валидными полями name, author, year и невалидным полем isElectronicBook ("isElectronicBook", 1, null) на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Статус запроса 400, ошибка "IsElectronicBook must be Bool type" |
| 29 | Редактирование книги с отсутствующем телом запроса | При редактировании книги с отсутствующем телом запроса статус запрос должен быть 400, тело ответа должно содержать сообщение об ошибке | Низкий | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1.Отправить PUT запрос c пустым телом {} на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Статус запроса 400, ошибка "Not found request Json body" |
| 30 | Получение всего списка книг после редактирования книги | При получение всего списка книг после редактирования книги данные последней книги из тела ответа должны совпадать с данными отредактированной книги | Средний | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1.Отправить PUT запрос на http://localhost:5000/api/books/{id}, где id - номер созданной книги  2. Отправить GET запрос на http://localhost:5000/api/books | 1. Книга отредактирована  2. Данные последней книги из тела ответа (id, name, author, year, isElectronicBook) совпадают с данными отредактированной книги |
| 31 | Получение отредактированной книги | При получении отредактированной книги данные из тела ответа должны совпадать с данными отредактированной книги | Средний | Создать книгу с помощью запроса POST на http://localhost:5000/api/books | 1.Отправить PUT запрос на http://localhost:5000/api/books/{id}, где id - номер созданной книги  2. Отправить GET запрос на http://localhost:5000/api/books/{id}, где id - номер созданной книги | 1. Книга отредактирована  2. Данные из тела ответа (id, name, author, year, isElectronicBook) совпадают с данными отредактированной книги |