**Набор тест-кейсов для тестирования Rest API приложения – библиотека книг.**

Для выполнения ручного тестирования использовался Postman.

Автотесты выполнены в среде Intellij Idea, на языке программирования Java c использованием Maven, Junit, Rest Assured

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Наименование тест-кейса** | **Необходимые действия** | **Ожидаемый результат** | **Наименование java класса** |
| **1** | Получение списка всех книг. | Отправить GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books> | В теле ответа получен Json со списком книг.  Получен Status Code 200. | getBooks.java |
| **2** | Получение книги по ее id. | Отправить GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books>  для получения списка всех существующих книг.  Отправить GET запрос на соответствующий endpoint:  [http://localhost:5000/api/books/<id](http://localhost:5000/api/books/%3cid)>  где <id> заменить на цифровое значение id книги, полученное на предыдущем шаге. | В теле ответа получен Json с книгой с заданным id.  Получен Status Code 200. | getBook.java |
| **3** | Тестирование запроса GET с обращением к несуществующему id. | Отправить GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books>  для получения списка всех существующих книг.  Отправить GET запрос на соответствующий endpoint:  [http://localhost:5000/api/books/<id](http://localhost:5000/api/books/%3cid)>  где <id> заменить на цифровое значение, отсутствующее в списке книг, полученных на предыдущем шаге. | Получен Status Code 404. | getBook.java |
| **4** | Добавление новой книги. | Подготовить данные для отправки в запросе, следующего формата:  {  "author": "Роберт Мартин",  "isElectronicBook": false,  "name": "Чистый код",  "year": 1998  }  Отправить POST запрос на соответствующий endpoint:  <http://localhost:5000/api/books>  Проверить, что отправленные данные успешно добавлены в список книг, отправив GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books> | В теле ответа получен код 201.  Добавляемая книга появилась в списке книг с новым id и все введенные данные соответствуют. | postBook.java |
| **5** | Добавление новой книги с заполнением только обязательного поля. | Подготовить данные для отправки в запросе, следующего формата:  {  "name": "Чистый код",  }  Отправить POST запрос на соответствующий endpoint:  <http://localhost:5000/api/books>  Проверить, что отправленные данные успешно добавлены в список книг, отправив GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books> | В теле ответа получен код 201.  Добавляемая книга появилась в списке книг с новым id, поле name соответствует введённому, остальные поля заполнены автоматически. | postBook.java |
| **6** | Проверка корректности типа вводимых полей при отправке POST запроса. | Подготовить данные для отправки в запросе, с неверным типом значения какого-либо поля. Например:  {  "author": "Роберт Мартин",  "isElectronicBook": "false",  "name": "Чистый код",  "year": "1998"  }  Отправить POST запрос на соответствующий endpoint:  <http://localhost:5000/api/books> | В теле ответа получен код 400. |  |
| **7** | Добавление новой книги с отсутствующим обязательным для заполнения полем. | Подготовить данные для отправки в запросе, следующего формата:  {  "author": "Роберт Мартин",  "isElectronicBook": "false",  "year": "1998"  }  Отправить POST запрос на соответствующий endpoint:  <http://localhost:5000/api/books> | В теле ответа получен код 400. | postBook.java |
| **8** | Тестирование запроса POST с подключением к ошибочному endpoint. | Подготовить данные для отправки в запросе, следующего формата:  {  "name": "Чистый код",  }  Отправить POST запрос на несоответствующий endpoint. Например:  <http://localhost:5000/api/books/qq> | В теле ответа получен код 405. | postBook.java |
| **9** | Обновление информации о книге по ее id. | Подготовить данные для отправки в запросе, следующего формата:  {  "author": "Роберт Мартин",  "isElectronicBook": false,  "name": "Чистый код",  "year": 1998  }  Отправить GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books>  для получения списка всех существующих книг.  Отправить PUT запрос на соответствующий endpoint:  [http://localhost:5000/api/books/<id](http://localhost:5000/api/books/%3cid)>  где <id> заменить на цифровое значение id книги, полученное на предыдущем шаге.  Проверить, что отправленные данные успешно обновлены, отправив GET запрос на endpoint:  [http://localhost:5000/api/books/<id](http://localhost:5000/api/books/%3cid)>  с id соответствующем id из предыдущего шага. | В теле ответа получен код 200.  Обновленная информация о книге соответствует отправляемым данным. | putBook.java |
| **10** | Проверка корректности типа вводимых полей при отправке PUT запроса. | Подготовить данные для отправки в запросе, с неверным типом значения какого-либо поля. Например:  {  "author": "Роберт Мартин",  "isElectronicBook": "false",  "name": "Чистый код",  "year": "1998"  }  Отправить GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books>  для получения списка всех существующих книг.  Отправить PUT запрос на соответствующий endpoint:  [http://localhost:5000/api/books/<id](http://localhost:5000/api/books/%3cid)>  где <id> заменить на цифровое значение id книги, полученное на предыдущем шаге. | В теле ответа получен код 400. |  |
| **11** | Обновление информации о книге по ее id без заполнения обязательных полей. | Подготовить данные для отправки в запросе, следующего формата:  {  "author": "Роберт Мартин",  "isElectronicBook": false,  "name": "Чистый код",  "year": 1998  }  Отправить GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books>  для получения списка всех существующих книг.  Отправить PUT запрос на соответствующий endpoint:  [http://localhost:5000/api/books/<id](http://localhost:5000/api/books/%3cid)>  где <id> заменить на цифровое значение id книги, полученное на предыдущем шаге. | В теле ответа получен код 400. | putBook.java |
| **12** | Тестирование запроса PUT с обращением к несуществующему id. | Подготовить данные для отправки в запросе, следующего формата:  {  "author": "Роберт Мартин",  "isElectronicBook": false,  "name": "Чистый код",  "year": 1998  }  Отправить GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books>  для получения списка всех существующих книг.  Отправить PUT запрос на соответствующий endpoint:  [http://localhost:5000/api/books/<id](http://localhost:5000/api/books/%3cid)>  где <id> заменить на цифровое значение, отсутствующее в списке книг, полученных на предыдущем шаге. | В теле ответа получен код 404. | puttBook.java |
| **13** | Тестирование запроса PUT с подключением к ошибочному endpoint. | Подготовить данные для отправки в запросе, следующего формата:  {  "author": "Роберт Мартин",  "isElectronicBook": false,  "name": "Чистый код",  "year": 1998  }  Отправить PUT запрос на несоответствующий endpoint. Например:  <http://localhost:5000/api/books/qq> | В теле ответа получен код 405. | putBook.java |
| **14** | Удаление книги по ее id. | Отправить GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books>  для получения списка всех существующих книг.  Отправить DELETE запрос на соответствующий endpoint:  [http://localhost:5000/api/books/<id](http://localhost:5000/api/books/%3cid)>  где <id> заменить на цифровое значение id книги, полученное на предыдущем шаге.  Проверить, что отправленные данные успешно удалены, отправив GET запрос на endpoint:  [http://localhost:5000/api/books/<id](http://localhost:5000/api/books/%3cid)>  с id соответствующем id из предыдущего шага. | В теле ответа получен код 200.  При вызове удаленной книги в теле ответа получен код 404. | deleteBook.java |
| **15** | Удаление книги с несуществующим id. | Отправить GET запрос на соответствующий endpoint:  <http://localhost:5000/api/books>  для получения списка всех существующих книг.  Отправить DELETE запрос на соответствующий endpoint:  [http://localhost:5000/api/books/<id](http://localhost:5000/api/books/%3cid)>  где <id> заменить на цифровое значение, отсутствующее в списке книг, полученных на предыдущем шаге. | В теле ответа получен код 404. | deleteBook.java |
| **16** | Тестирование запроса DELETE с подключением к ошибочному endpoint. | Отправить DELETE запрос на несоответствующий endpoint. Например:  <http://localhost:5000/api/books/qq> | В теле ответа получен код 405. | deleteBook.java |