**ЦРТ**

**ТЕСТ-КЕЙСЫ ДЛЯ REST API «БИБЛИОТЕКА КНИГ»**

Дата создания документа: 14.02.2023

Документ создал: Казанский Кирилл Александрович

Содержание

[**1.** **Назначение документа** 3](#_Toc127384849)

[**2.** **Сценарии тестирования** 3](#_Toc127384850)

[**2.1.** **Тестирование работоспособности** 3](#_Toc127384851)

[**2.1.1.** **Получение всех записей.** 3](#_Toc127384852)

[**2.1.2.** **Получение конкретной записи.** 3](#_Toc127384853)

[**2.1.3.** **Обновление имеющейся записи.** 3](#_Toc127384854)

[**2.1.4.** **Добавление новой записи.** 3](#_Toc127384855)

[**2.1.5.** **Удаление записи.** 4](#_Toc127384856)

[**2.2.** **Тестирование соответствия данных** 5](#_Toc127384857)

[**2.2.1.** **Обновление имеющейся записи.** 5](#_Toc127384858)

[**2.2.2.** **Добавление новой записи.** 5](#_Toc127384859)

[**2.2.3.** **Удаление записи.** 5](#_Toc127384860)

**Назначение документа**

Данный документ предназначен для текстового описания логики, структуры и ожидаемых результатов автоматизированных тест-кейсов для api «Библиотека книг».

1. **Сценарии тестирования**

Все сценарии, описанные ниже производятся на тестовом сервере и не должны иметь доступ к публичным данным.

* 1. **Тестирование работоспособности**

Примечание: данные сценарии используются для проверки способности api обрабатывать запросы.

* + 1. **Получение всех записей.**

Действия:

1. Используя endpoint GET /api/books производится get запрос;
2. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса.
3. Тест завершен успешно.
   * 1. **Получение конкретной записи.**

Действия:

1. Используя endpoint GET /api/books производится get запрос;
2. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса.
3. Если: записи отсутствуют
   1. Тест провален с отметкой об отсутствии записей.
4. Используя endpoint GET /api/books/<book\_id> и id первой записи производится get запрос;
5. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса
6. Тест завершен успешно.
   * 1. **Обновление имеющейся записи.**

Действия:

1. Используя endpoint GET /api/books производится get запрос;
2. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса.
3. Если: записи отсутствуют
   1. Тест провален с отметкой об отсутствии записей.
4. Используя endpoint PUT /api/books/<book\_id> и id первой записи производится put запрос с измененными данными;
5. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса
6. Тест завершен успешно.
   * 1. **Добавление новой записи.**

Действия:

1. Создается новая запись;
2. Используя endpoint POST /api/books и данные новой записи производится post запрос;
3. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса
4. Тест завершен успешно.
   * 1. **Удаление записи.**

Действия:

1. Используя endpoint GET /api/books производится get запрос;
2. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса.
3. Если: записи отсутствуют
   1. Тест провален с отметкой об отсутствии записей.
4. Используя endpoint DELETE /api/books/<book\_id> и id первой записи производится delete запрос;
5. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса
6. Тест завершен успешно.
   1. **Тестирование соответствия данных**

Примечание: данные сценарии используются для проверки корректности обработки запросов api.

* + 1. **Обновление имеющейся записи.**

Действия:

1. Используя endpoint GET /api/books производится get запрос;
2. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса.
3. Если: записи отсутствуют
   1. Тест провален с отметкой об отсутствии записей.
4. Используя endpoint PUT /api/books/<book\_id> и id первой записи производится put запрос с измененными данными;
5. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса
6. Если: обновлённая запись не соответствует измененным данным
   1. Тест провален с отображением не соответствия
7. Тест завершен успешно.
   * 1. **Добавление новой записи.**

Действия:

1. Создается новая запись;
2. Используя endpoint POST /api/books и данные новой записи производится post запрос;
3. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса
4. Если: добавленная запись не соответствует данным новой записи
   1. Тест провален с отображением не соответствия
5. Тест завершен успешно.
   * 1. **Удаление записи.**

Действия:

1. Используя endpoint GET /api/books производится get запрос;
2. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса.
3. Если: записи отсутствуют
   1. Тест провален с отметкой об отсутствии записей.
4. Используя endpoint DELETE /api/books/<book\_id> и id первой записи производится delete запрос;
5. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса
6. Используя endpoint GET /api/books производится get запрос;
7. Если: код статуса запроса не равен ОК(200)
   1. Тест провален с отображением кода статуса.
8. Если: количество записей не изменилось
   1. Тест провален с пометкой о том, что запись не удалилась.
9. Тест завершен успешно.