

№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	1	Получение списка всех книг		1. Отправить Get запрос http://localhost:5000/api/books	Запрос успешно отправлен
				2. Проверить код состояния	HTTP Status: 200 OK
				3. Проверить тело ответа от сервера.	Тело ответа в формате JSON будет иметь следующий вид: { "books": [{ "author": "Роберт Мартин", "id": 1, "isElectronicBook": false, "name": "Икстый код", "year": 1998 }, { "author": "Томас Кормен, Чарльз Лейкерсон", "id": 2, "isElectronicBook": true, "name": "Алгоритмы: построение и анализ", "year": 1989 }] }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	2	Получить книгу по id Необходимо добавить книгу. 1. Отправить Post запрос http://localhost:5000/api/books В формате JSON передать в body запрос: { "name": "Java", "author": "Katy Era", "isElectronicBook": true, "year": 2019 } 2. Полученный в теле ответа id книги, использовать в Get запросе для получения книги.		1. Отправить Get запрос http://localhost:5000/api/books/id	Запрос успешно отправлен
				2. Проверить код состояния	HTTP Status: 200 OK
				3. Проверить тело ответа от сервера	Тело ответа в формате JSON будет иметь следующий вид: { "book": { "author": "Katy Era", "id": (указанный id), "isElectronicBook": true, "name": "Java", "year": 2019 } }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	3	Получить книгу по несуществующему id		1. Отправить Get запрос http://localhost:5000/api/books/9999999	Запрос успешно отправлен
				2. Проверить код состояния	HTTP Status: 404 NOT FOUND
				3. Проверить тело ответа от сервера	Тело ответа в формате JSON, будет иметь следующий вид: { "error": "Book with id 9999999not found" }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	4	Добавить новую книгу	Параметры для метода Post передаются в теле запроса в json объекте. Тело запроса: { "name": "Selenium Python", "author": "Томас Кормен, Чарльз Лейкерсон", "isElectronicBook": true, "year": 2022 }	1. Отправить Post запрос http://localhost:5000/api/books Заполнить body тестовыми данными.	Запрос успешно отправлен
				2. Проверить код состояния	Status: 201 CREATED
				3. Проверить тело ответа от сервера	Тело ответа в формате JSON возвращается от сервера и будет иметь следующий вид: { "book": { "author": "Томас Кормен, Чарльз Лейкерсон", "id": 3, "isElectronicBook": true, "name": "Selenium Python", "year": 2022 } }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	5	Добавить новую книгу указав только название книги	Параметры для метода Post передаются в теле запроса в json объекте. Тело запроса { "name": "Selenium Java" }	1. Отправить Post запрос http://localhost:5000/api/books Заполнить body тестовыми данными	Запрос успешно отправлен
				2. Проверить код состояния	Status: 201 CREATED
				3. Проверить тело ответа от сервера	Поле "author", "year" и "isElectronicBook" заполнились значениями по умолчанию Тело ответа в формате JSON будет иметь следующий вид: { "book": { "author": "", "id": 9, "isElectronicBook": false, "name": "Selenium Java", "year": 0 } }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	6	Добавить новую книгу с большим названием	Параметры для метода Post передаются в теле запроса в json объекте. Тело запроса { "name": "Рассвет полнокни, или Совершенство славы, торжества и мудрости порфиноносных, браноносных и мирных теневя России с последованием дидактических, зротических и других равного рода в стихах и прозе опыта Семена Боброва", "author": "Бобров Семён Сергеевич", "isElectronicBook": true, "year": 1763 }	1. Отправить Post запрос http://localhost:5000/api/books	Запрос успешно отправлен
				2. Проверить код состояния	Status: 201 CREATED
				3. Проверить тело ответа от сервера	Название отображается также как не запрос. Тело ответа в формате JSON будет иметь следующий вид: { "book": { "author": "Бобров Семён Сергеевич", "id": 8, "isElectronicBook": true, "name": "Рассвет полнокни, или Совершенство славы, торжества и мудрости порфиноносных, браноносных и мирных теневя России с последованием дидактических, зротических и других равного рода в стихах и прозе опыта Семена Боброва", "year": 1763 } }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	7	Добавить новую книгу с спецсимволами в названии	Параметры для метода Post передаются в теле запроса в json объекте. Тело запроса { "name": "« Энциклопедия символов »" }	1. Отправить Post запрос http://localhost:5000/api/books	Запрос успешно отправлен
				2. Проверить код состояния	Status: 201 CREATED
				3. Проверить тело ответа от сервера	Тело ответа в формате JSON будет иметь следующий вид: { "book": { "author": "", "id": 10, "isElectronicBook": false, "name": "« Энциклопедия символов »", "year": 0 } }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	8	Добавить новую книгу с названием из цифр	Параметры для метода Post передаются в теле запроса в json объекте. Тело запроса { "name": "1945" }	1. Отправить Post запрос http://localhost:5000/api/books	Запрос успешно отправлен
				2. Проверить код состояния	Status: 201 CREATED
				3. Проверить тело ответа	Тело ответа в формате JSON будет иметь следующий вид: { "book": { "author": "", "id": 12, "isElectronicBook": false, "name": "1945", "year": 0 } }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	9	Добавить новую книгу указав 2024 год	Параметры для метода Post передаются в теле запроса в json объекте. Тело запроса { "name": "Год 2024 еще не наступит", "year": 2024 }	1. Отправить Post запрос http://localhost:5000/api/books	Запрос успешно отправлен
				2. Проверить код состояния	HTTP Status: 400 BAD REQUEST
				3. Проверить в теле ответа	Тело ответа в формате JSON будет иметь следующий вид: { "error": "Invalid date format" }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	10	Добавить новую книгу с несуществующим годом	Параметры для метода Post передаются в теле запроса в json объекте. Тело запроса { "name": "1945", "yeav": "-1" }	1. Отправить Post запрос http://localhost:5000/api/books	Запрос успешно отправлен
				2. Проверить код состояния	HTTP Status: 400 BAD REQUEST
				3. Проверить в теле ответа	Тело ответа в формате JSON будет иметь следующий вид: { "error": "Invalid date format" }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	11	Добавить новую книгу заполнить поле name проблемами	Параметры для метода Post передаются в теле запроса в json объекте. Тело запроса { "name": "", "author": "Томас Кормен", "isElectronicBook": true, "year": 2022 }	1. Отправить Post запрос http://localhost:5000/api/books	Запрос успешно отправлен
				2. Проверить код состояния	HTTP Status: 400 BAD REQUEST
				3. Проверить тело ответа от сервера	Тело ответа в формате JSON будет иметь следующий вид: { "error": "Required name field is not filled " }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат
	12	Добавить новую книгу оставить поле name пустым	Параметры для метода Post передаются в теле запроса в json объекте. Тело запроса { "name": "", "author": "Кирилл Мнотроунов", "isElectronicBook": true, "year": 2012 }	1. Отправить Post запрос http://localhost:5000/api/books	Запрос успешно отправлен
				2. Проверить код состояния	HTTP Status: 400 BAD REQUEST
				3. Проверить тело ответа от сервера	Тело ответа в формате JSON будет иметь следующий вид: { "error": "Required name field is not filled " }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат

	13	Обновить информацию о книге	Необходимо добавить книгу. 1. Отправить Post запрос http://localhost:5000/api/books В формате JSON передать в body запрос: { "name": "Острые козырьки", "author": "ТОМАС ШЕНБИ", "isElectronicBook": false, "year": 2017 } 2.Полученный в теле ответа id книги, использовать в Put запросе для обновления информации о книге.	Параметры для метода Put передается в теле запроса в json объекте. Тело запроса { "name": "Острые козырьки", "author": "Томас Шенби", "isElectronicBook": true, "year": 2016 }	1.Отправить PUT запрос http://localhost:5000/api/books/id 2.Проверить код состояния 3. Проверить в теле ответа	Запрос успешно отправлен HTTP Status: 200 OK Тело ответа в формате JSON будет иметь следующий вид { "book": { "author": "Томас Шенби", "id": (указанный id), "isElectronicBook": true, "name": "Острые козырьки", "year": 2016 } }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат	
	14	Обновить информацию о книге указав несуществующий id	Параметры для метода Post передается в теле запроса в json объекте. Тело запроса { "name": "Король лев", "author": "Тимон и пумба", "isElectronicBook": true, "year": 1995 }	1.Отправить PUT запрос http://localhost:5000/api/books/9999999 2.Проверить код состояния 3. Проверить в теле ответа	Запрос успешно отправлен HTTP Status: 404 NOT FOUND Тело ответа в формате JSON будет иметь следующий вид { "error": "Book with id 9999999 not found" }	
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат	
	15	Обновить информацию о книге оставив поле name пустым	Необходимо добавить книгу. 1. Отправить Post запрос http://localhost:5000/api/books В формате JSON передать в body запрос: { "name": "Сказка про Шрека", "author": "Осен", "isElectronicBook": false, "year": 2000 } 2.Полученный в теле ответа id книги, использовать в Put запросе для обновления информации о книге.	Параметры для метода Post передается в теле запроса в json объекте. Тело запроса { "name": "", "author": "Тимон и пумба", "isElectronicBook": true, "year": 1978 }	1.Отправить Post запрос http://localhost:5000/api/books/id 2.Проверить код состояния 3. Проверить в теле ответа	Запрос успешно отправлен HTTP Status: 400 BAD REQUEST Тело ответа в формате JSON будет иметь следующий вид { "error": "Required name field is not filled " }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат	
	16	Обновить информацию о книге заполнив поле name пробелами	Необходимо добавить книгу. 1. Отправить Post запрос http://localhost:5000/api/books В формате JSON передать в body запрос: { "name": "Король лев", "author": "Тимон", "isElectronicBook": false, "year": 1996 } 2.Полученный в теле ответа id книги, использовать в Put запросе для обновления информации о книге.	Параметры для метода Post передается в теле запроса в json объекте. Тело запроса { "name": " ", "author": "Тимон и пумба", "isElectronicBook": true, "year": 1986 }	1.Отправить PUT запрос http://localhost:5000/api/books/id 2.Проверить код состояния 3. Проверить в теле ответа	Запрос успешно отправлен HTTP Status: 400 BAD REQUEST Тело ответа в формате JSON будет иметь следующий вид { "error": "Required name field is not filled " }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат	
	17	Удалить книгу	Необходимо добавить книгу. 1. Отправить Post запрос http://localhost:5000/api/books В формате JSON передать в body запрос: { "name": "Сумерки", "author": "Сэмюэл К", "isElectronicBook": true, "year": 2012 } 2.Полученный в теле ответа id книги, использовать в Delete запросе для удаления книги.	Параметры для метода Delete передается в теле запроса в json объекте. Тело запроса { "name": " ", "author": "Тимон и пумба", "isElectronicBook": true, "year": 1986 }	1.Отправить Delete запрос http://localhost:5000/api/books/id 2.Проверить код состояния 3. Проверить в теле ответа	Запрос успешно отправлен HTTP Status: 200 OK Тело ответа в формате JSON возвращается от сервера и будет иметь следующий вид { "result": true }
№	Название	Предусловия	Тестовые данные	Шаги	Ожидаемый результат	
	18	Удалить книгу с не верным id		1.Отправить Delete запрос http://localhost:5000/api/books/199 2.Проверить код состояния 3. Проверить в теле ответа	Запрос успешно отправлен HTTP Status: 404 NOT FOUND Тело ответа в формате JSON будет иметь следующий вид { "error": "Book with id 199 not found" }	