

Коллекция Postman: <https://.postman.co/workspace/My-Workspace~dffd081c-b865-4f6f-8501-59ce344aa29b/collection/41481443-37ed2a58-76bc-497c-8731-98ec6dfb894c?action=share&creator=41481443>

Тестовый стенд: книжный магазин  
Отчет по тестированию API Booking  
Дата тестирования: 7.02.2025  
Версия API: v2  
Количество запросов: 7

## 1. Обзор коллекции

Коллекция охватывает полный CRUD-цикл работы с бронированиями: -  
Создание токена аутентификации  
- Создание/чтение/обновление/удаление бронирований  
- Визуализация данных ответов

## 2. Детальный анализ эндпоинтов

### 2.1 Аутентификация

POST /auth

- Назначение: Получение токена для авторизованных операций - Тело запроса: username/password  
- Особенности: Используется в других запросах через Cookie

### 2.2 Операции с бронированиями

Метод	Endpoint	Назначение	Особенности
-------	----------	------------	-------------

GET	/booking	Получение списка ID бронирований	Визуализация "No Bookings Available"
-----	----------	----------------------------------	--------------------------------------

GET	/booking/{id}	Получение деталей бронирования	Красивая табличная визуализация данных
-----	---------------	--------------------------------	--

POST	/booking	Создание бронирования	Пример тела: данные клиента, даты, дополнительные услуги
------	----------	-----------------------	--

PUT	/booking/{id}	Полное обновление бронирования	Требует токена в Cookie
-----	---------------	--------------------------------	-------------------------

PATCH	/booking/{id}	Частичное обновление	Изменение только имени/фамилии
-------	---------------	----------------------	--------------------------------

DELETE	/booking/{id}	Удаление бронирования	Требует авторизации
--------	---------------	-----------------------	---------------------

## 3. Примеры запросов

Создание бронирования (POST): json

```
{  
  "firstname": "Amy",  
  "lastname": "Green",  
  "totalprice": 122, "depositpaid": true, "bookingdates": {  
    "checkin": "2025-02-23",  
    "checkout": "2025-10-23" },  
}
```

```
"additionalneeds": "Breakfast" }
```

Частичное обновление (PATCH): json

```
{  
  "firstname": "Alise", "lastname": "Green"  
}
```

#### 4. Визуализация данных

Коллекция содержит скрипты для отображения данных в табличном формате:

- Стилизованные таблицы CSS
- Условное форматирование (например, для depositpaid)
- Ховер-эффекты для строк

#### 5. Рекомендации по тестированию

##### 1. Позитивные тесты:

- Все CRUD-операции с валидными данными
- Проверка граничных значений дат
- Проверка разных additionalneeds

##### 2. Негативные тесты:

- Попытки операций без токена
- Невалидные форматы дат
- Отрицательные значения price
- Попытки изменить несуществующее бронирование

##### 3. Дополнительные проверки:

- Валидация структуры ответов
- Проверка времени ответа
- Тестирование параллельных запросов

#### 6. Статистика коллекции

- GET: 2 запроса
- POST: 2 запроса
- PUT: 1 запрос
- PATCH: 1 запрос
- DELETE: 1 запрос