

1. Отправка POST запроса к эндпоинту /auth. Токен получен.

Описание:

Запрос на создание токена для доступа к операциям put и delete /booking.

Шаги:

Отправляем запрос к api с эндпоинтом /auth, в body указать данные пользователя, логин/пароль (admin/password123).

```
{
  "username" : "admin",
  "password" : "password123"
}
```

Ожидаемый результат:

Получен ответ с кодом 200. Ответ содержит token.

2. Отправка POST запроса к эндпоинту /auth. Некорректные учетные данные.

Описание:

Запрос на создание токена для доступа к операциям put и delete /booking.

Шаги:

Отправляем запрос к api с эндпоинтом /auth, в body вместо логина/пароля ничего не указывать, оставить тело запроса пустым.

```
{ }
```

Ожидаемый результат:

*В документации не указано поведение системы в случае выполнения запроса без тела.

Получен ответ с кодом 200. Ответ содержит Bad credentials.

3. Отправка POST запроса к эндпоинту /booking. Код 200. Получен id бронирования.

Описание:

Запрос на создание нового бронирования.

Шаги:

Отправляем запрос к api с эндпоинтом /booking, в body указываем следующие параметры: имя, фамилию, стоимость, есть ли депозит, даты бронирования, дополнительные нужды.

```
{
  "firstname" : "Jim",
  "lastname" : "Brown",
  "totalprice" : 111,
  "depositpaid" : true,
  "bookingdates" : {
    "checkin" : "2018-01-01",
    "checkout" : "2019-01-01"
  },
}
```

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

Ожидаемый результат:

Получен ответ с кодом 200. Ответ содержит информацию о бронировании включая созданный идентификатор бронирования (bookinkid).

4. Отправка POST запроса к эндпоинту /booking. Код 500. Ошибка обработки запроса.

Описание:

Запрос на создание нового бронирования.

Шаги:

Отправляем запрос к api с эндпоинтом /booking, в body указываем следующие параметры: имя, фамилию, стоимость, есть ли депозит, дополнительные нужды. Из body -> bookingdates убрать параметры даты заселения и дата выезда.

```
{
    "firstname" : "Jim",
    "lastname" : "Brown",
    "totalprice" : 111,
    "depositpaid" : true,
    "bookingdates" : {

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

Ожидаемый результат:

*В документации не указано поведение системы в случае выполнения запроса без каких-либо параметров - требует уточнения.

Получен ответ с кодом 500.

5. Отправка POST запроса к эндпоинту /booking. Код 400. Некорректный запрос.

Описание:

Запрос на создание нового бронирования.

Шаги:

Отправляем запрос к api с эндпоинтом /booking, в body указываем следующие параметры: имя, фамилию, стоимость, есть ли депозит, дата, заселения и дата выезда, дополнительные нужды.

Из body -> bookingdates убрать параметр дата заселения, оставив пустые кавычки.

```
{
    "firstname" : "Jim",
    "lastname" : "Brown",
```

```
"totalprice" : 111,  
"depositpaid" : true,  
"bookingdates" : {  
    "checkin" : "",  
    "checkout" : "2019-01-01"  
}  
"additionalneeds" : "Breakfast"  
}
```

Ожидаемый результат:

*В документации не указано поведение системы в случае выполнения запроса без каких-либо параметров - требует уточнения.

Получен ответ с кодом 400 и текстом Bad Request.

6. Отправка GET запроса к эндпоинту /booking. Получены идентификаторы всех бронирований в системе.

Описание:

Получение уникальных идентификаторов всех существующих в системе бронирований.

Шаги:

Отправляем запрос к api с эндпоинтом /booking. В данном случае запрос не содержит параметров.

Ожидаемый результат:

Ответ с кодом 200. Получен список идентификаторов всех бронирований.

7. Отправка GET запроса к эндпоинту /booking с параметрами. Фильтр по Имени.

Описание:

Получение уникальных идентификаторов всех существующих в системе бронирований на конкретное имя клиента.

Шаги:

Отправляем запрос к api с эндпоинтом /booking. В параметрах запроса необходимо создать два параметра:

"firstname": "Jim"

"lastname": "Brown"

*параметры опциональны, возможно получить данные только по имени или только по фамилии.

Ожидаемый результат:

Ответ с кодом 200. Получен список идентификаторов всех бронирований на имя Jim Brown.

8. Отправка GET запроса к эндпоинту /booking с параметрами. Фильтр по дате.

Описание:

Получение уникальных идентификаторов всех существующих в системе бронирований на конкретные даты.

Шаги:

Отправляем запрос к api с эндпоинтом /booking. В параметрах запроса необходимо создать два параметра:

"checkin": "2014-03-13"

"checkout": "2014-05-21"

*параметры опциональны, возможно получить данные только по дате заселения или по дате выселения.

Ожидаемый результат:

Ответ с кодом 200. Получен список идентификаторов всех бронирований на заданный период.

9. Отправка GET запроса к эндпоинту /booking/:id. Код 200. Получение информации по id.

Описание:

Получение информации о бронировании по id бронирования.

Шаги:

Отправляем запрос к api с эндпоинтом /booking/:id. В параметрах запроса необходимо указать:

"id": "1"

Ожидаемый результат:

Ответ с кодом 200. Получена информация о бронировании.

10. Отправка GET запроса к эндпоинту /booking/:id. Код 404. Бронирование не найдено.

Описание:

Получение информации о бронировании по id бронирования.

Шаги:

Отправляем запрос к api с эндпоинтом /booking/:id. В параметрах запроса необходимо указать:

"id": "27050"

Ожидаемый результат:

Ответ с кодом 404. Not found. Бронирование не найдено.

11. Отправка PUT запроса к эндпоинту /booking/:id. Обновление текущего бронирования.

Описание:

Обновление информации текущего бронирования.

Шаги:

Отправляем запрос к api с эндпоинтом /booking/:id. В параметрах запроса необходимо создать параметр:

"id": "1".

В body указать:

```
{
  "firstname" : "James",
  "lastname" : "Brown",
  "totalprice" : 111,
  "depositpaid" : true,
  "bookingdates" : {
```

```
        "checkin" : "2018-01-01",
        "checkout" : "2019-01-01"
    },
    "additionalneeds" : "Breakfast"
}
```

Ожидаемый результат:

Ответ с кодом 200. Информация о текущем бронировании обновлена.

12. Отправка PATCH запроса к эндпоинту /booking/:id. Обновление текущего бронирования с частичной нагрузкой.

Описание:

Обновление информации о текущем бронировании с частичной нагрузкой.

Шаги:

Отправляем запрос к api с эндпоинтом /booking/:id. В параметрах запроса необходимо указать:

"id": "1"

В body указать:

```
{
    "firstname" : "James",
    "lastname" : "Brown"
}
```

Ожидаемый результат:

Ответ с кодом 200. Обновлена информация о текущем бронировании с частичной нагрузкой.

13. Отправка DELETE запроса к эндпоинту /booking/:id. Удалить бронирование.

Описание:

Удаление информации о бронировании.

Шаги:

Отправляем запрос к api с эндпоинтом /booking/:id. В параметрах запроса необходимо создать параметр:

"id": "1"

Ожидаемый результат:

Ответ с кодом 200. Информация о бронировании удалена.

14. Отправка GET запроса к эндпоинту /ping.

Описание:

Проверка работоспособности API.

Шаги:

Отправляем запрос к api с эндпоинтом /ping. В параметрах запроса ничего не нужно указывать.

Ожидаемый результат:

Ответ с кодом 200, в случае если API запущен и работает. В случае если нет - ответ с кодом ошибки (*в документации не указано какой код ошибки должны получить в таком случае).

