**Кейс:**

Крупная компания планирует создать Систему для бронирования отелей. Система позволяет бронировать номера в отелях по всем странам. В системе можно бронировать номера в отелях, с возможностью предоплаты и без.

**Список функциональных требований:**  
1. Регистрация пользователей и создание профилей как для пользователей, так и для владельцев отелей;

2. Поиск отелей по странам и городам;

3. Просмотр информации об отеле, номерах и ценах для бронирования;

4. Бронирование номеров с возможностью предоплаты или без;

5. Роли в системе: пользователь, админ отеля;

6. Возможность отказаться от предоплаты и оплатить номер при заселении;

7. Оплата комиссии системе за бронирование;

8. Получение электронного подтверждения бронирования;

9. Интеграция с системами платежных сервисов;

10. Отчетность для администраторов, описывающая количество забронированных номеров, частоту бронирований и т.д.

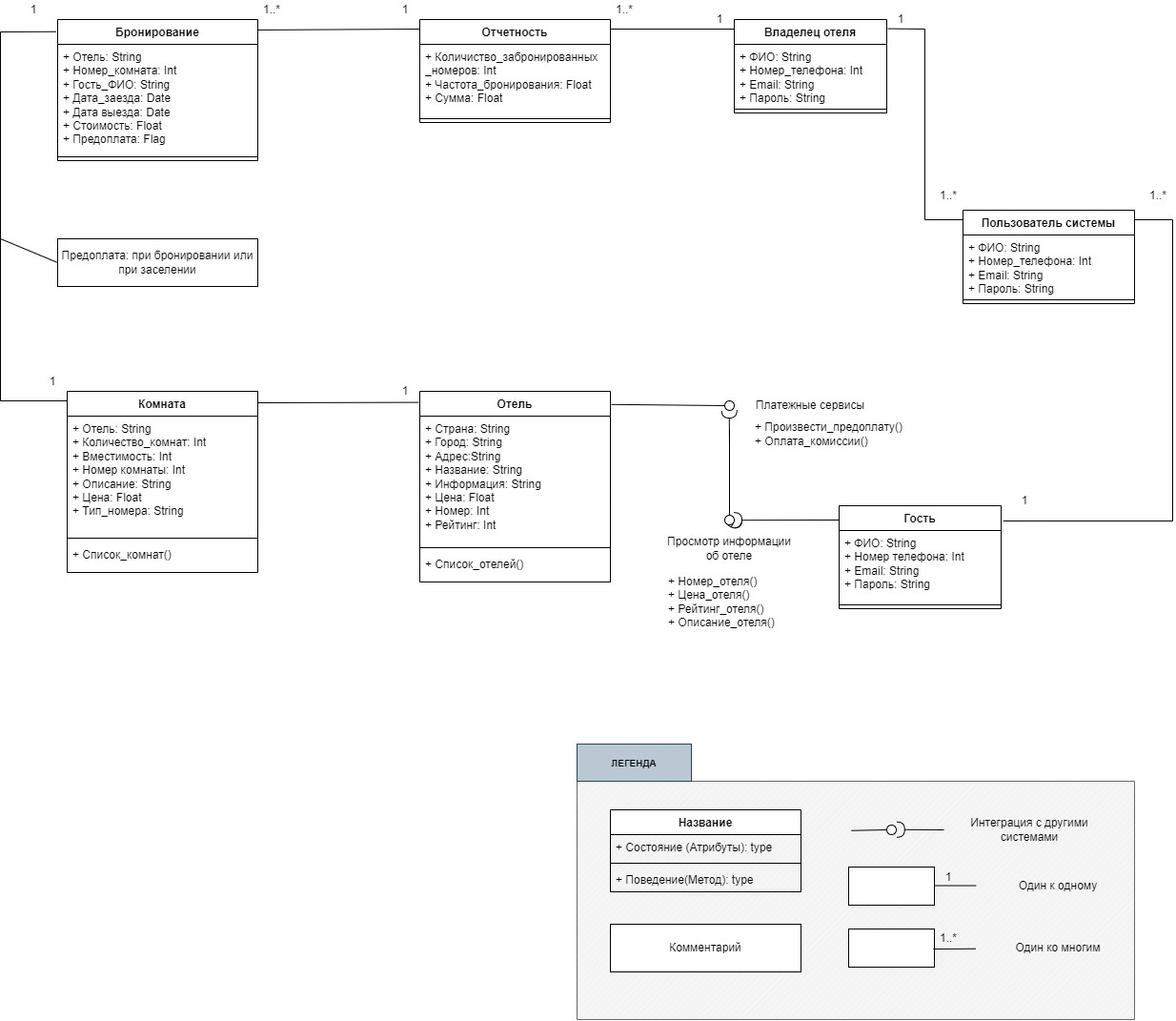
**Задание:**

1. Описать систему с помощью UML-диаграмм как: Диаграмма классов, диаграмма последовательностей, ER-диаграмма.
2. Описать API (в REST) получения всех отелей, номеров в отелях, и бронирования. Должно быть включены: примеры запроса/ответа сервера, входные/выходные данные и тд.

**Описать систему с помощью UML-диаграмм как: Диаграмма классов, диаграмма последовательностей, ER-диаграмма.**

**Диаграмма классов.**

**Система бронирования отелей.**

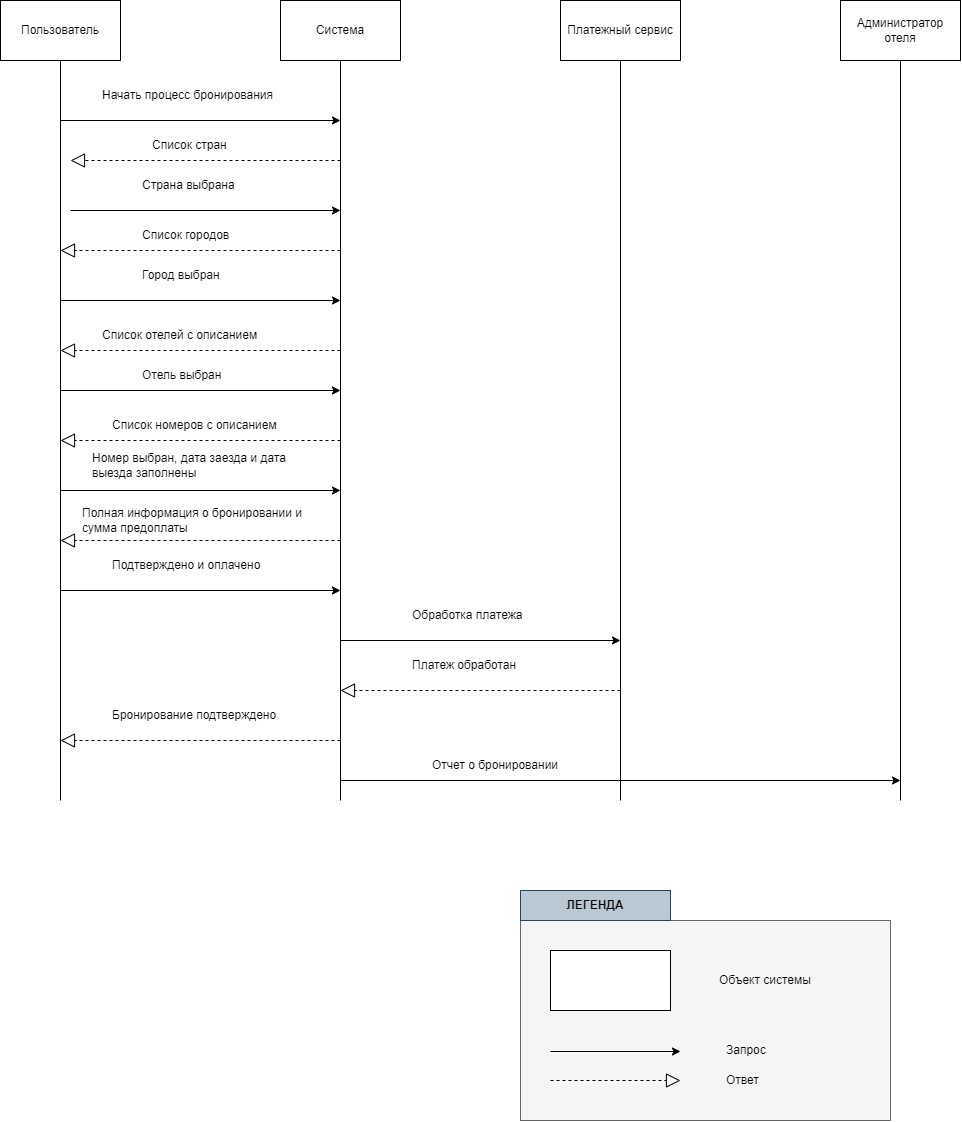


**Диаграмма последовательностей.**

**Система бронирования отелей.**

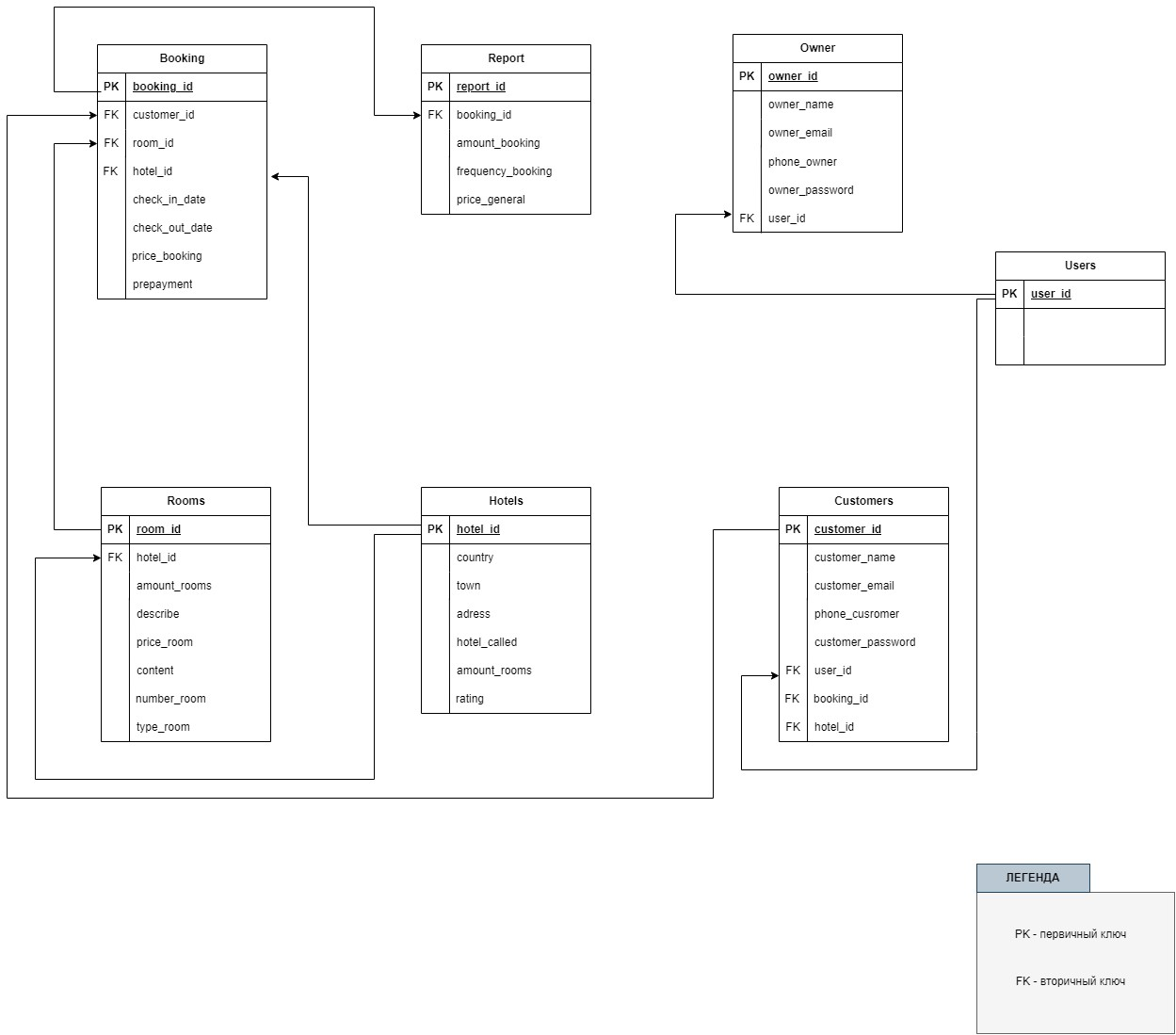
**Сценарий предоплаты.**

**Преусловие – пользователь уже зарегистрирован в системе.**

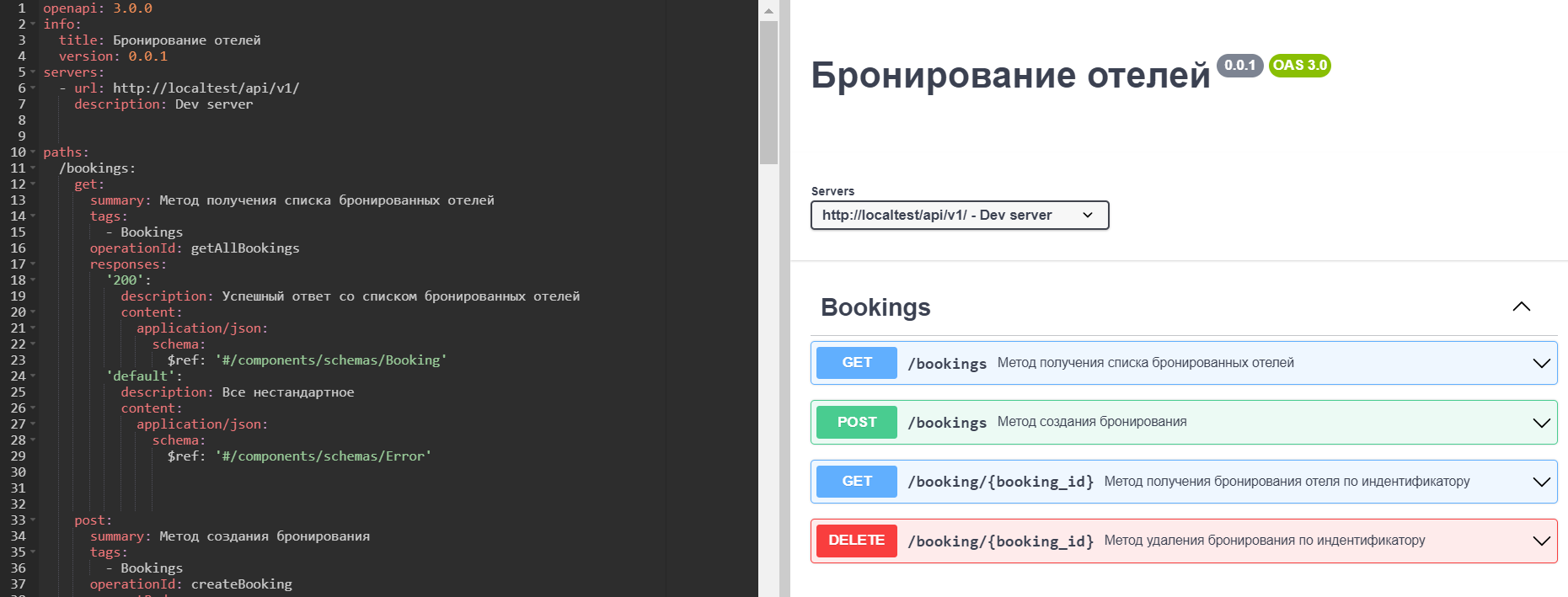


**ER-диаграмма.**

**Система бронирования отелей.**



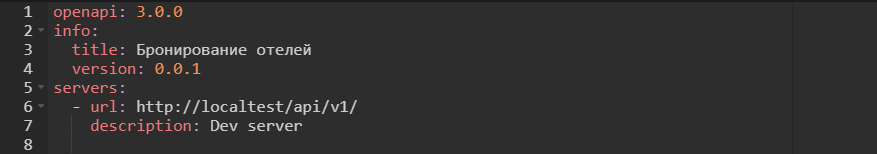
**Описать API (в REST) получения всех отелей, номеров в отелях, и бронирования. Должно быть включены: примеры запроса/ответа сервера, входные/выходные данные и тд.**



Детально:

Заполнение основной информации о спецификации:   
- название

- версия  
- сервера, на которые будут отправляться запросы, если нажать Try it out



Методы:

- указан путь

- указаны методы GET, POST, DELETE

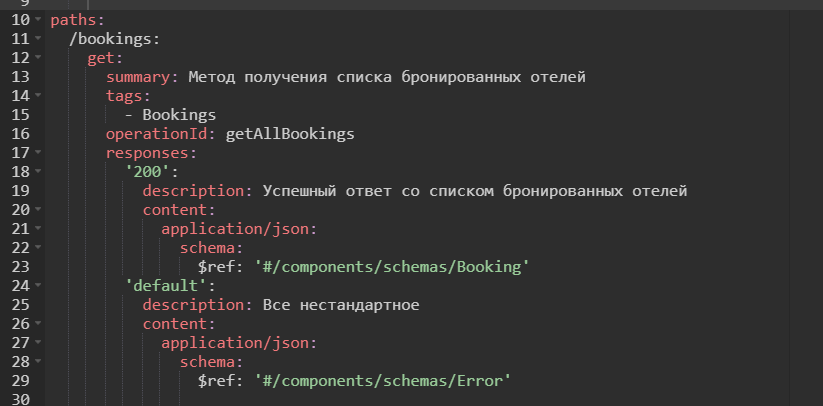
- описание каждого метода  
- его теги (массив)  
- responses (коды ответов: «202» и «default»)

- requestBody (указан тип json)

- parameters (указываем, где параметр передается)

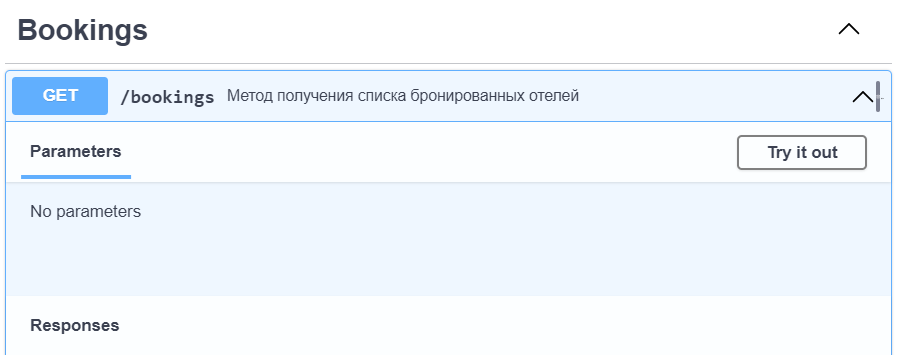
Метод GET

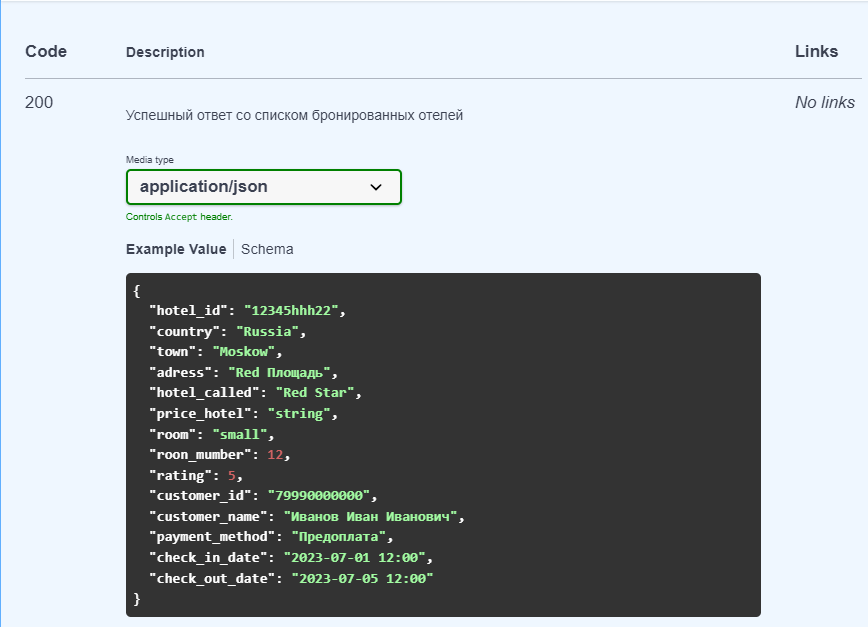
- Получение списка всех доступных бронированных отелей



Ответ

Метод не принимает никаких параметров, но возвращает в ответе код 200 с успешным ответом со списком бронированных отелей. И структура ответа.

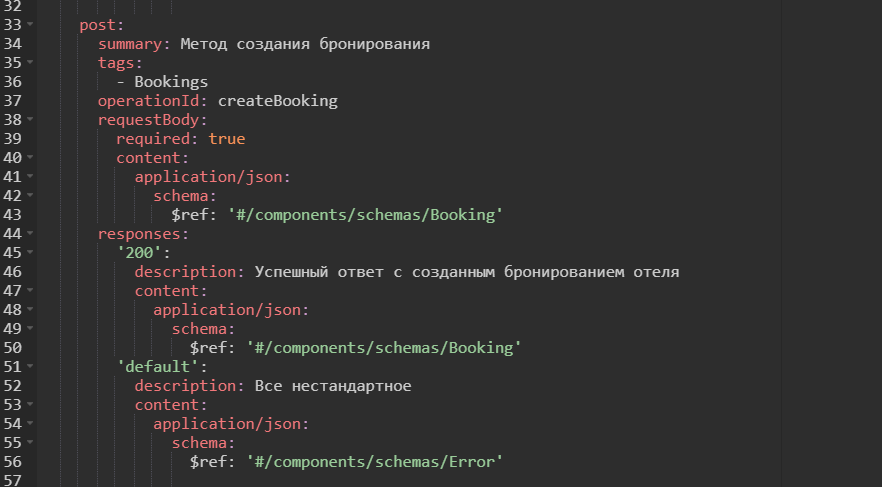




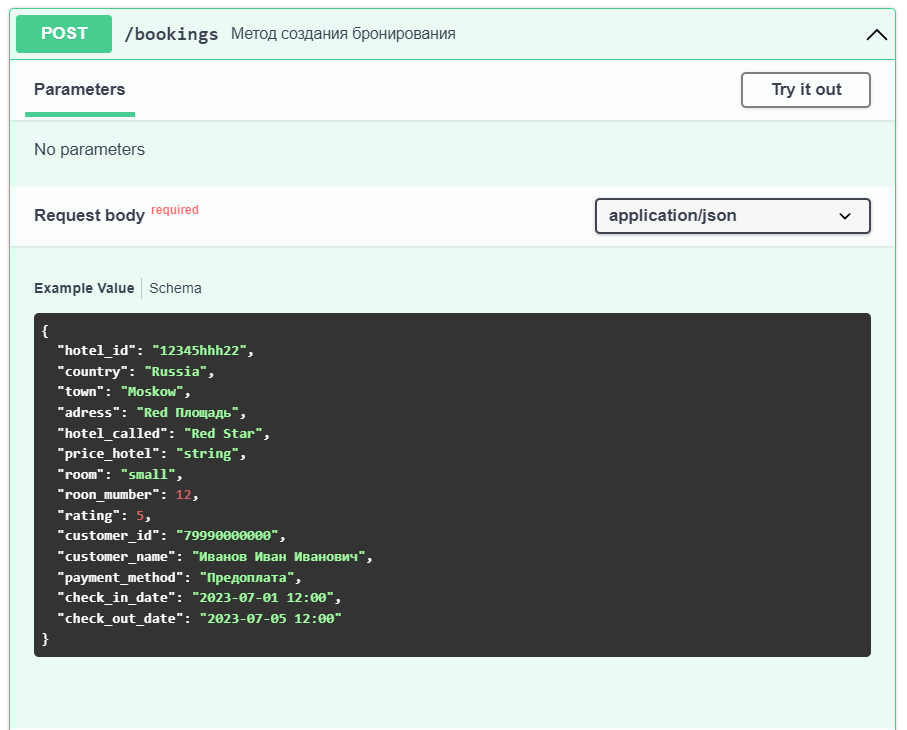


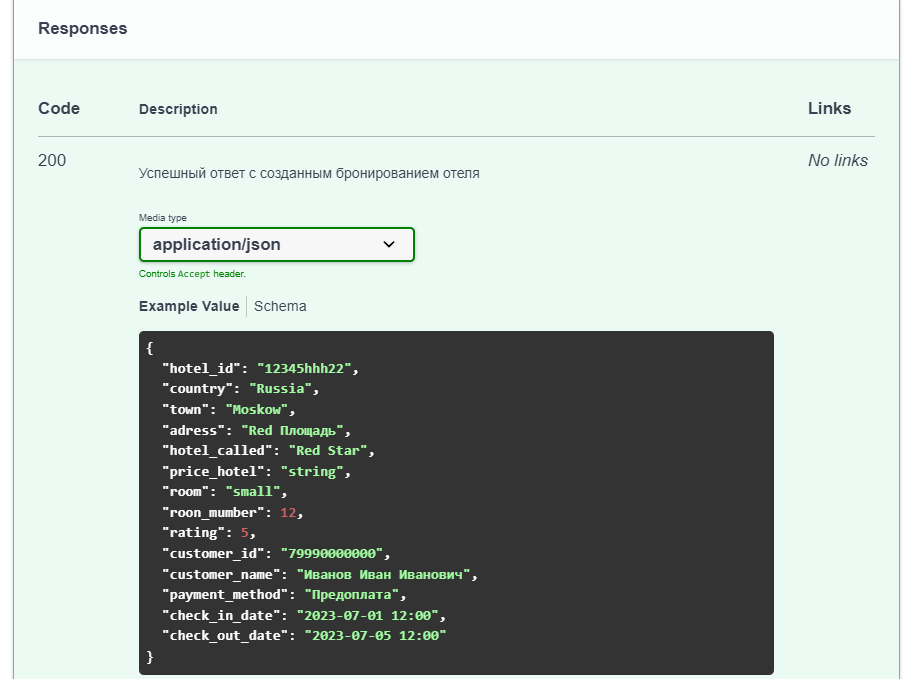
Метод POST

Метод создания бронирования.



Ответ:



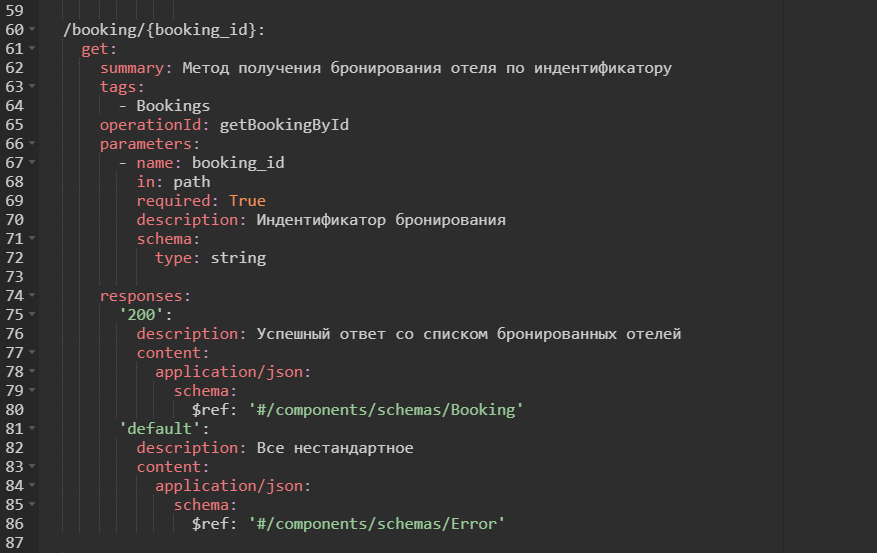




Метод GET

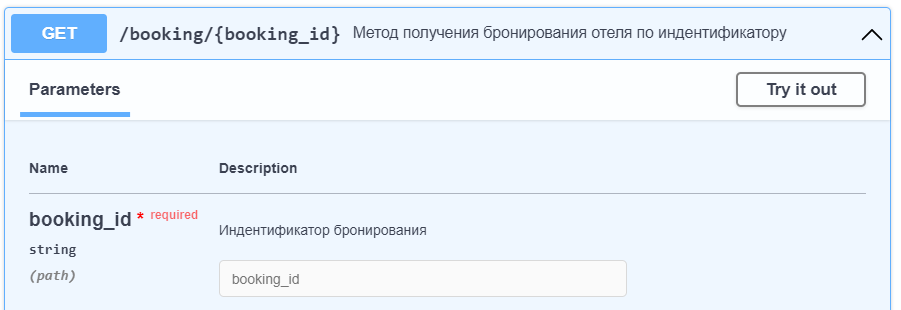
Получение одного бронирования отеля по идентификатору.

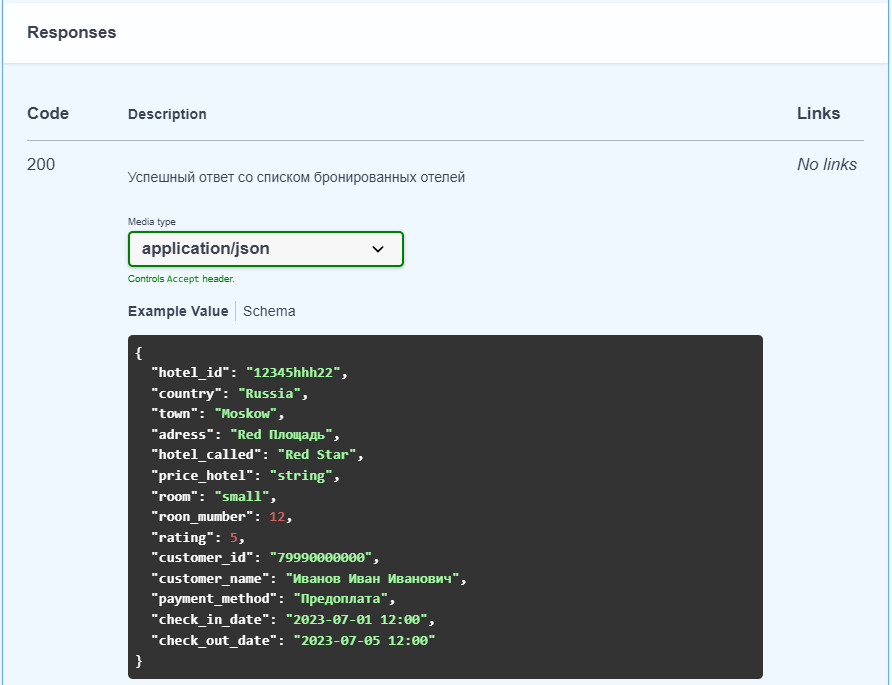
Здесь появился parameters – booking\_id

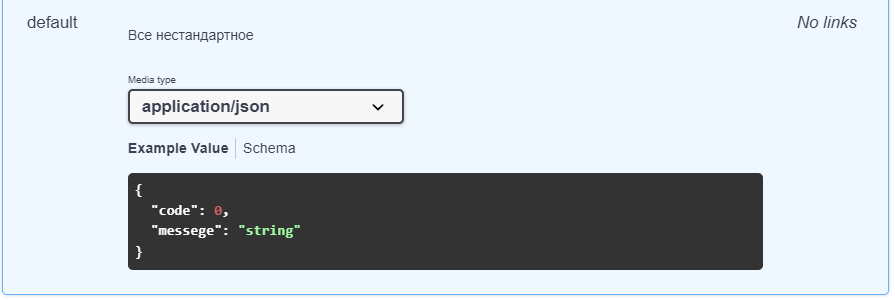


Ответ:

Блок Parameters заполнен.

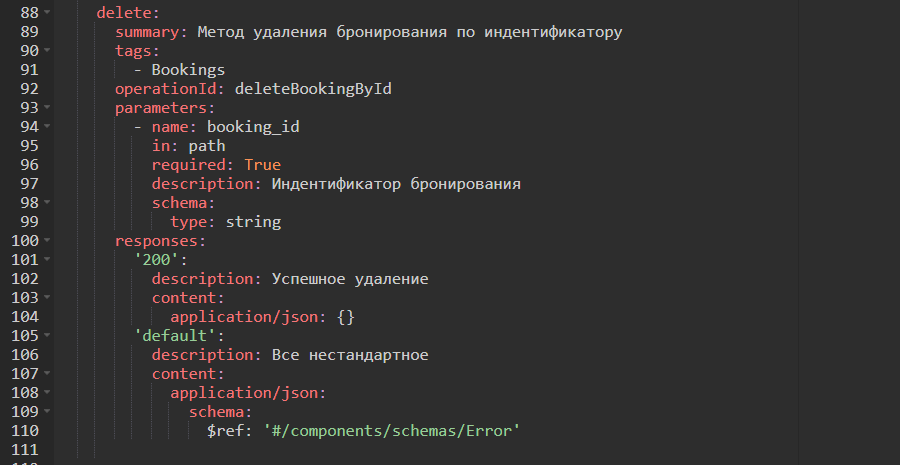






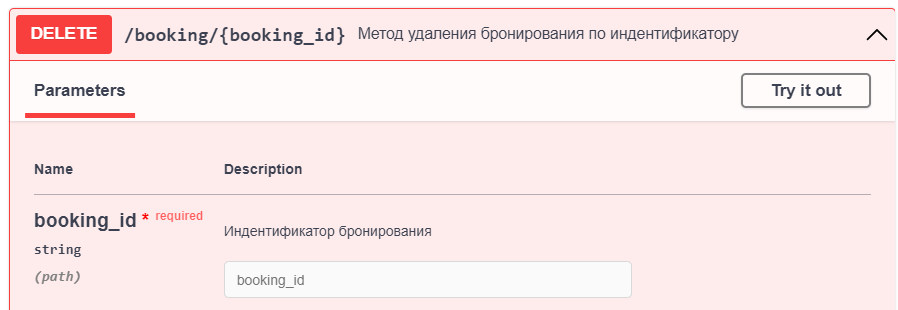
Метод DELETE

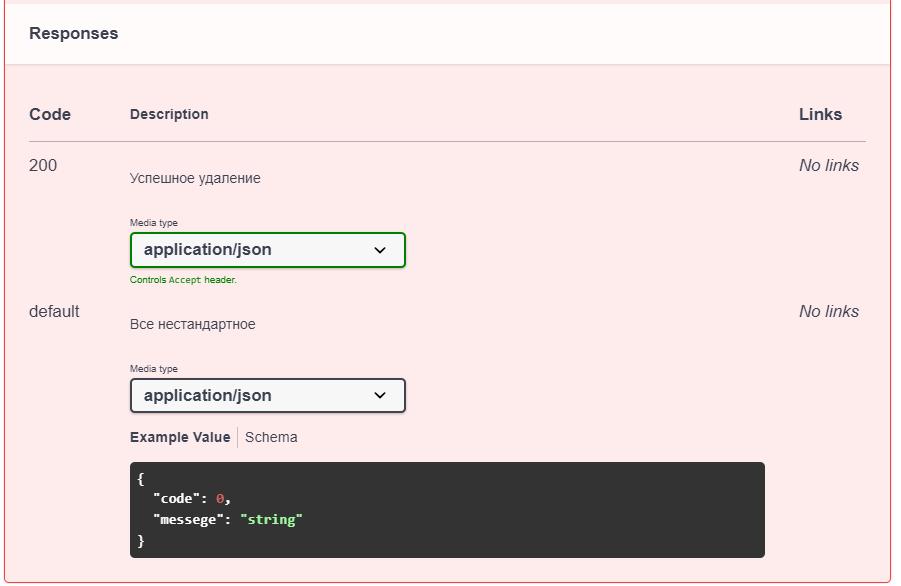
Ничего не возвращает.



Ответ:

Принимает идентификатор booking\_id, возвращает пустое 200 или ошибку.



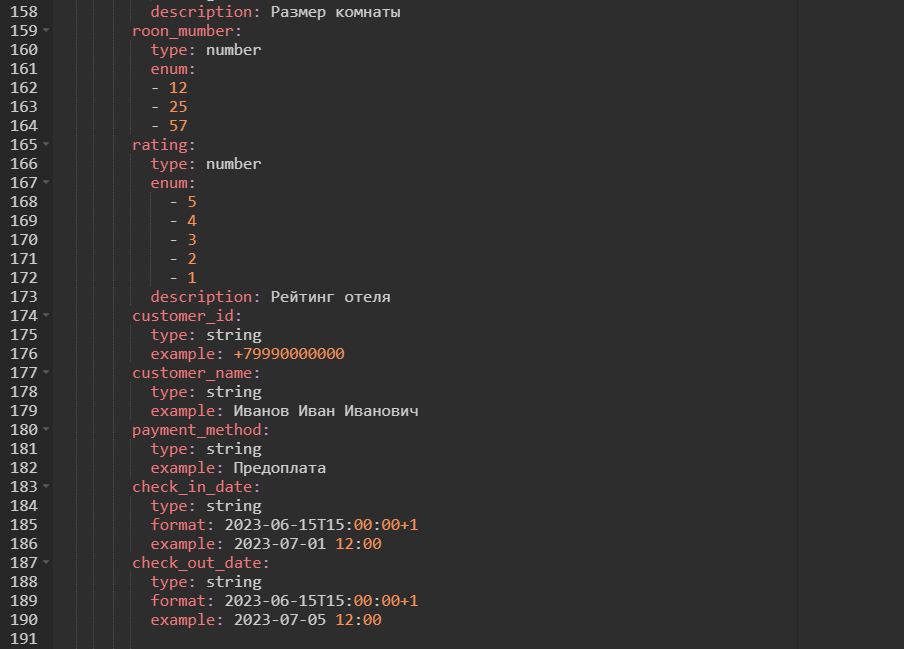


Блок components

Здесь все модели данных.

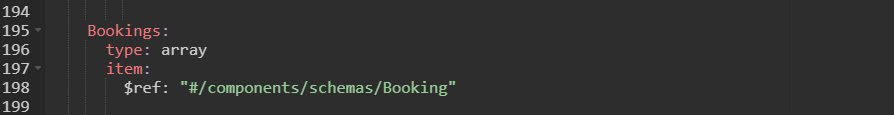
Обязательные атрибуты и их описание.





Массив бронированных отелей.

Ссылается на объект бронирования.



Объект ошибка.

Если с сервером что-то пойдет не так. Туда будет попадать все, что не 200.



Блок-схемы

