**Тест-кейсы**

**Исполнитель:** Шабашкин (Сергеев на hh.ru) Дмитрий Сергеевич   
**Дата создания:** 23.06.2025  
**Проект:** Тестовое задание / ЯСП

**Примечание**: Баг репорты хранятся в отдельном документе.

**ID: 01**  
**Название теста:** Авторизация с валидным логином и паролем  
**Тип сценария:** Позитивный  
**Предусловия:**  
1) API доступен  
2) Установлен Content-type: application/json  
3) Метод Post / Auth доступен без предварительной авторизации  
**Эндпоинт:** Post / Auth  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:**  
{  
"username": "admin",  
"password": "password123"  
}  
**Шаги выполнения:**  
1) Выбрать метод POST, указать URL <https://restful-booker.herokuapp.com/auth>  
2) Установить заголовок Content-Type: application/json  
3 )В теле запроса ввести корректные данные (username, password)  
4) Сохранить token в переменную коллекции  
5) Отправить запрос  
6) Проверить статус ответа и наличие поля token

**Ожидаемый результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит поле token

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит поле token

**Статус:** Pass

**Комментарии:** -

**ID: 02**  
**Название теста:** Авторизация с неверным паролем  
**Тип сценария:** Негативный  
**Предусловия:**  
1) API доступен  
2) Установлен Content-type: application/json  
3) Метод Post / Auth доступен без предварительной авторизации  
**Эндпоинт:** Post / Auth  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:**  
{  
"username": "admin",  
"password": "password12345"  
}  
**Шаги выполнения:**  
1) Выбрать метод POST, указать URL <https://restful-booker.herokuapp.com/auth>  
2) Установить заголовок Content-Type: application/json  
3) В теле запроса ввести данные с валидным username и неверным паролем  
4) Отправить запрос  
5) Проверить статус ответа

**Ожидаемый результат:**  
1) Статус ответа: 401 Unauthorized  
2) Тело ответа не содержит поле token

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит поле "reason": "Bad credentials"

**Статус:** Fail

**Комментарии:** - API некорректно обрабатывает запрос, ожидалос 401 Unauthorized – Баг (Баги описаны в отдельном документе xlxx)

**ID: 03**  
**Название теста:** Авторизация с неверным именем пользователя  
**Тип сценария:** Негативный  
**Предусловия:**  
1) API доступен  
2) Установлен Content-type: application/json  
3) Метод Post / Auth доступен без предварительной авторизации  
**Эндпоинт:** Post / Auth  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:**  
{  
"username": "invalidUser",  
"password": "password123"  
}  
**Шаги выполнения:**  
1) Выбрать метод POST, указать URL <https://restful-booker.herokuapp.com/auth>  
2) Установить заголовок Content-Type: application/json  
3) В теле запроса ввести неверное имя пользователя и валидный пароль  
4) Отправить запрос  
5) Проверить статус ответа

**Ожидаемый результат:**  
1) Статус ответа: 401 Unauthorized  
2) Тело ответа не содержит поле token и возвращает сообщение об ошибке

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит: "reason": "Bad credentials"

**Статус:** Fail

**Комментарии:** - API возвращает 200 OK при неправильном username - Баг

**ID: 04**  
**Название теста:** Создание бронирования с валидными данными  
**Тип сценария:** Позитивный  
**Предусловия:**  
1) API доступен  
2) Установлен Content-type: application/json  
3) Метод Post /booking доступен без авторизации  
**Эндпоинт:** Post /booking  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:**  
{  
"firstname": "Jim",  
"lastname": "Brown",  
"totalprice": 111,  
"depositpaid": true,  
"bookingdates": {  
"checkin": "2018-01-01",  
"checkout": "2019-01-01"  
},  
"additionalneeds": "Breakfast"  
}  
**Шаги выполнения:**  
1) Выбрать метод POST, указать URL <https://restful-booker.herokuapp.com/booking>  
2) Установить заголовок Content-Type: application/json  
3) В теле запроса ввести валидные данные бронирования  
4) Сохранить bookingid в переменную коллекции  
5) Отправить запрос  
5) Проверить статус ответа и наличие созданного объекта booking

**Ожидаемый результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит поле bookingid и объект booking с соответствующими данными

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит bookingid и объект booking

**Статус:** Pass

**Комментарии:** -

**ID: 05**  
**Название теста:** Создание бронирования с отсутствующим полем firstname  
**Тип сценария:** Негативный  
**Предусловия:**  
1)API доступен  
2)Установлен Content-type: application/json  
3)Метод Post /booking доступен без авторизации  
**Эндпоинт:** Post /booking  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:**  
{  
"lastname": "Brown",  
"totalprice": 111,  
"depositpaid": true,  
"bookingdates": {  
"checkin": "2018-01-01",  
"checkout": "2019-01-01"  
},  
"additionalneeds": "Breakfast"  
}  
**Шаги выполнения:**  
1) Выбрать метод POST, указать URL <https://restful-booker.herokuapp.com/booking>  
2) Установить заголовок Content-Type: application/json  
3) В теле запроса не указывать поле firstname  
4) Отправить запрос  
5) Проверить статус ответа и тело ответа на наличие ошибки

**Ожидаемый результат:**  
1) Статус ответа: 400 Bad Request

**Фактический результат:**  
1) Статус ответа: 500 Internal Server Error

**Статус:** Fail

**Комментарии:** - API некорректно обрабатывает запрос и возвращает 500 Internal Server Error – Баг

**ID: 06**  
**Название теста:** Создание бронирования с буквенным значением в поле totalprice  
**Тип сценария:** Негативный  
**Предусловия:**  
1) API доступен  
2) Установлен Content-type: application/json  
3) Метод Post /booking доступен без авторизации  
**Эндпоинт:** Post /booking  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:**  
{  
"firstname": "Jim",  
"lastname": "Brown",  
"totalprice": "twenty",  
"depositpaid": true,  
"bookingdates": {  
"checkin": "2018-01-01",  
"checkout": "2019-01-01"  
},  
"additionalneeds": "Breakfast"  
}  
**Шаги выполнения:**  
1) Выбрать метод POST, указать URL <https://restful-booker.herokuapp.com/booking>  
2) Установить заголовок Content-Type: application/json  
3) В теле запроса указать строковое значение "one hundred" в поле totalprice  
4) Отправить запрос  
5) Проверить статус ответа и сообщение об ошибке

**Ожидаемый результат:**  
1) Статус ответа: 400 Bad Request

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Строка totalprice принимает значение null

**Статус:** Fail

**Комментарии:** API не валидирует null в числовом поле totalprice – Баг

**ID: 07**  
**Название теста:** Создание бронирования со строковым значением в поле depositpaid  
**Тип сценария:** Негативный  
**Предусловия:**  
1) API доступен  
2) Установлен Content-type: application/json  
3) Метод Post /booking доступен без авторизации  
**Эндпоинт:** Post /booking  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:**  
{  
"firstname": "Jim",  
"lastname": "Brown",  
"totalprice": 111,  
"depositpaid": "true",  
"bookingdates": {  
"checkin": "2018-01-01",  
"checkout": "2019-01-01"  
},  
"additionalneeds": "Breakfast"  
}  
**Шаги выполнения:**  
1) Выбрать метод POST, указать URL <https://restful-booker.herokuapp.com/booking>  
2) Установить заголовок Content-Type: application/json  
3) В теле запроса передать строку "true" в поле depositpaid  
4) Отправить запрос  
5) Проверить статус ответа и значение в теле ответа

**Ожидаемый результат:**  
1) Статус ответа: 400 Bad Request

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) "depositpaid": true

**Статус:** Fail

**Комментарии:** - API некорректно валидирует тип данных, допускает строку - Баг

**ID: 08**  
**Название теста:** Создание бронирования с неверным форматом даты  
**Тип сценария:** Негативный  
**Предусловия:**  
1) API доступен  
2) Установлен Content-type: application/json  
3) Метод Post /booking доступен без авторизации  
**Эндпоинт:** Post /booking  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:**  
{  
"firstname": "Jim",  
"lastname": "Brown",  
"totalprice": 111,  
"depositpaid": true,  
"bookingdates": {  
"checkin": "02-02-2022",  
"checkout": "2022-02-02"  
},  
"additionalneeds": "Breakfast"  
}  
**Шаги выполнения:**  
1) Выбрать метод POST, указать URL <https://restful-booker.herokuapp.com/booking>  
2) Установить заголовок Content-Type: application/json  
3) Указать неверный формат даты в поле checkin  
4) Отправить запрос  
5) Проверить статус ответа и поведение API

**Ожидаемый результат:**  
1) Статус ответа: 400 Bad Request  
2) Ошибка валидации формата даты

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Сервер успешно создал бронирование

**Статус:** Fail

**Комментарии:** - Отсутствует валидация формата даты в поле checkin - Баг

**ID: 09**  
**Название теста:** Создание бронирования с невалидными значениями даты  
**Тип сценария:** Негативный  
**Предусловия:**  
1) API доступен  
2) Установлен Content-type: application/json  
3) Метод Post /booking доступен без авторизации  
**Эндпоинт:** Post /booking  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:**

{

"firstname": "Jim",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "0000-00-00",

"checkout": "0000-00-01"

},

"additionalneeds": "Breakfast"

}

**Шаги выполнения:**  
1) Выбрать метод POST, указать URL <https://restful-booker.herokuapp.com/booking>  
2) Установить заголовок Content-Type: application/json  
3) В теле запроса указать невалидные значения даты в полях checkin и checkout  
4) Отправить запрос  
5) Проверить статус ответа и тело ответа

**Ожидаемый результат:**  
1) Статус ответа: 400 Bad Request

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит некорректные значения: "checkin": "0NaN-aN-aN" и "checkout": "0NaN-aN-aN"

**Статус:** Fail

**Комментарии:** - API не валидирует значения дат, создает некорректные записи

**ID: 10**  
**Название теста:** Создание бронирования с датой выезда раньше даты заезда  
**Тип сценария:** Негативный  
**Предусловия:**  
1) API доступен  
2) Установлен Content-type: application/json  
3) Метод Post /booking доступен без авторизации  
**Эндпоинт:** Post /booking  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:**

{

"firstname": "Jim",

"lastname": "Bill",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2022-01-01",

"checkout": "2019-01-01"

},

"additionalneeds": "Breakfast"

}

**Шаги выполнения:**  
1) Выбрать метод POST, указать URL <https://restful-booker.herokuapp.com/booking>  
2) Установить заголовок Content-Type: application/json  
3) В теле запроса указать дату выезда раньше даты заезда  
4) Отправить запрос  
5) Проверить статус ответа и содержание данных бронирования

**Ожидаемый результат:**  
1) Статус ответа: 400 Bad Request

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Создана бронь с невалидными датами

**Статус:** Fail

**Комментарии:** - Логика API нарушена, не проверяет корректность введенных дат - Баг

**ID: 11**  
**Название теста:** Проверка доступности API через GET /ping  
**Тип сценария:** Позитивный  
**Предусловия:**  
1) API доступен  
2) Метод GET /ping не требует авторизации  
3) Установлен Content-Type: application/json

**Эндпоинт:** GET /ping  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:** -

**Шаги выполнения:**  
1) Выбрать метод GET, указать URL <https://restful-booker.herokuapp.com/ping>  
2) Установить заголовок Content-Type: application/json  
3) Отправить запрос  
4) Проверить статус ответа  
5) Проверить тело ответа

**Ожидаемый результат:**  
1) Статус ответа: 201 Created  
2) Тело ответа: "Created"

**Фактический результат:**  
1) Статус ответа: 201 Created  
2) Тело ответа: "Created"

**Статус:** Pass

**Комментарии:** -

**ID: 12**  
**Название теста:** Получение информации о бронировании по ID  
**Тип сценария:** Позитивный  
**Предусловия:**  
1) API доступен  
2) Существует бронирование с указанным ID  
3) Метод GET /booking/:id не требует авторизации  
4) Установлен Content-Type: application/json

**Эндпоинт:** GET /booking/{id}  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:** -

**Шаги выполнения:**  
1) Выбрать метод GET, указать URL https://restful-booker.herokuapp.com/booking/1  
2) Установить заголовок Content-Type: application/json  
3) Отправить запрос  
4) Проверить статус ответа  
5) Проверить наличие всех полей в теле ответа

**Ожидаемый результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит все поля бронирования

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит все поля бронирования

**Статус:** Pass

**Комментарии:** -

**ID: 13**  
**Название теста:** Получение информации по несуществующему ID  
**Тип сценария:** Негативный  
**Предусловия:**  
1) API доступен  
2) ID = 0 не существует в системе  
3) Метод GET /booking/:id не требует авторизации  
4) Установлен Content-Type: application/json

**Эндпоинт:** GET /booking/0  
**Заголовки запроса:** Content-Type: application/json  
**Тело запроса:** -

**Шаги выполнения:**  
1) Выбрать метод GET, указать URL <https://restful-booker.herokuapp.com/booking/0>  
2) Установить заголовок Content-Type: application/json  
3) Отправить запрос  
4) Проверить статус ответа  
5) Проверить, что тело пустое или содержит сообщение об ошибке

**Ожидаемый результат:**  
1) Статус ответа: 404 Not Found  
2) Тело ответа пустое или содержит сообщение "Not Found"

**Фактический результат:**  
1) Статус ответа: 404 Not Found  
2) Тело ответа: "Not Found"

**Статус:** Pass

**Комментарии:** -

**ID: 14**  
**Название теста:** Успешное обновление бронирования по ID  
**Тип сценария:** Позитивный  
**Предусловия:**  
1) API доступен  
2) Существует бронирование с ID {{auth\_bookingid}}  
3) Пользователь авторизован и доступен токен  
4) Установлен Content-Type: application/json  
5) Метод PUT /booking/:id требует авторизацию ( Post / Auth )

**Эндпоинт:** PUT /booking/{{auth\_bookingid}}  
**Заголовки запроса:**  
Content-Type: application/json  
Cookie: token={{auth\_token}}

**Тело запроса:**

{

"firstname": "BlaBla",

"lastname": "BlaP",

"totalprice": 999,

"depositpaid": true,

"bookingdates": {

"checkin": "2028-01-01",

"checkout": "2029-01-01"

},

"additionalneeds": "Breakfast"

}

**Шаги выполнения:**  
1) Выбрать метод PUT, указать URL https://restful-booker.herokuapp.com/booking/{{auth\_bookingid}}  
2) Установить заголовок Content-Type: application/json  
3) Добавить Cookie с токеном авторизации  
4) В теле запроса ввести актуальные данные  
5) Отправить запрос  
6) Проверить статус ответа и содержание тела

**Ожидаемый результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит обновленные поля бронирования

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит обновленные данные бронирования

**Статус:** Pass

**Комментарии:** -

**ID: 15**  
**Название теста:** Обновление бронирования без авторизации  
**Тип сценария:** Негативный  
**Предусловия:**  
1) API доступен  
2) Существует бронирование с ID {{auth\_bookingid}}  
3) Токен авторизации отсутствует   
4) Установлен Content-Type: application/json

**Эндпоинт:** PUT /booking/{{auth\_bookingid}}  
**Заголовки запроса:**  
Content-Type: application/json

**Тело запроса:**

{

"firstname": "BlaBla",

"lastname": "BlaP",

"totalprice": 999,

"depositpaid": true,

"bookingdates": {

"checkin": "2028-01-01",

"checkout": "2029-01-01"

},

"additionalneeds": "Breakfast"

}

**Шаги выполнения:**  
1) Выбрать метод PUT, указать URL https://restful-booker.herokuapp.com/booking/{{auth\_bookingid}}  
2) Установить заголовок Content-Type: application/json  
3) Не добавлять токен авторизации (ни Cookie, ни Header)  
4) Ввести тело запроса с валидными данными  
5) Отправить запрос  
6) Проверить ответ

**Ожидаемый результат:**  
1) Статус ответа: 403 Forbidden  
2) Тело ответа содержит сообщение об ошибке авторизации (если предусмотрено)

**Фактический результат:**  
1) Статус ответа: 403 Forbidden  
2) Тело ответа: "Forbidden"

**Статус:** Pass

**Комментарии:** -

**ID: 16**  
**Название теста:** Частичное обновление бронирования по несуществующему ID  
**Тип сценария:** Негативный

**Предусловия:**  
1) API доступен  
2) Токен авторизации сохранён и передаётся в Cookie (или выполнить запрос Puth / Auth, так как Токен имеет ограниченное количество применений)  
3) Установлен Content-Type: application/json  
4) ID бронирования не существует (например, 0)

**Эндпоинт:** PATCH /booking/0

**Заголовки запроса:**  
Content-Type: application/json  
Cookie: token={{auth\_token}}

**Тело запроса:**

{

"firstname" : "BlaBla",

"lastname" : "BlaP",

"totalprice" : 999,

"depositpaid" : true,

"bookingdates" : {

"checkin" : "2028-01-01",

"checkout" : "2029-01-01"

},

"additionalneeds" : "Breakfast"

}

**Шаги выполнения:**  
1) Выбрать метод PATCH, указать URL <https://restful-booker.herokuapp.com/booking/0>  
2) Установить заголовок Content-Type: application/json  
3) Добавить Cookie с авторизационным токеном  
4) Ввести валидное тело запроса  
5) Отправить запрос  
6) Проверить ответ

**Ожидаемый результат:**  
1) Статус ответа: 404 Not Found / 405 Method Not Allowed  
2) Сообщение в теле (при 405): "Method Not Allowed"

**Фактический результат:**  
1) Статус ответа: 405 Method Not Allowed  
2) Тело ответа: "Method Not Allowed"

**Статус:** Pass

**Комментарии:** -

**ID: 17**  
**Название теста:** Частичное обновление бронирования (firstname и lastname)  
**Тип сценария:** Позитивный

**Предусловия:**  
1) API доступен  
2) Существует бронирование с ID {{auth\_bookingid}}  
3) Токен авторизации сохранён и передаётся в Cookie  
4) Установлен Content-Type: application/json

**Эндпоинт:** PATCH /booking/{{auth\_bookingid}}

**Заголовки запроса:**  
Content-Type: application/json  
Cookie: token={{auth\_token}}

**Тело запроса:**

{

"firstname": "Alexey",

"lastname": "Skvortcov"

}

**Шаги выполнения:**  
1) Выбрать метод PATCH, указать URL https://restful-booker.herokuapp.com/booking/{{auth\_bookingid}}  
2) Установить заголовок Content-Type: application/json  
3) Добавить Cookie с авторизационным токеном  
4) Указать тело запроса с firstname и lastname  
5) Отправить запрос  
6) Проверить ответ

**Ожидаемый результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит обновлённые значения

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит обновлённые поля firstname и lastname

**Статус:** Pass

**Комментарии:** -

**ID: 18**  
**Название теста:** Частичное обновление бронирования без авторизации  
**Тип сценария:** Негативный

**Предусловия:**  
1) API доступен  
2) Существует бронирование с ID {{auth\_bookingid}}  
3) Токен авторизации отсутствует  
4) Установлен Content-Type: application/json

**Эндпоинт:** PATCH /booking/{{auth\_bookingid}}

**Заголовки запроса:**  
Content-Type: application/json

**Тело запроса:**

{

"firstname": "Alexey",

"lastname": "Skvortcov"

}

**Шаги выполнения:**  
1) Выбрать метод PATCH, указать URL https://restful-booker.herokuapp.com/booking/{{auth\_bookingid}}  
2) Установить заголовок Content-Type: application/json  
3) Не добавлять токен авторизации   
4) Указать тело запроса с firstname и lastname  
5) Отправить запрос  
6) Проверить ответ

**Ожидаемый результат:**  
1) Статус ответа: 403 Forbidden

**Фактический результат:**  
1) Статус ответа: 403 Forbidden  
2) Тело ответа: "Forbidden"

**Статус:** Pass

**Комментарии:** -

**ID: 19**  
**Название теста:** Успешное удаление бронирования  
**Тип сценария:** Позитивный

**Предусловия:**  
1) API доступен  
2) Существует бронирование с ID {{auth\_bookingid}}  
3) Авторизационный токен сохранён и передаётся в Cookie  
4) Установлен Content-Type: application/json

**Эндпоинт:** DELETE /booking/{{auth\_bookingid}}

**Заголовки запроса:**  
Content-Type: application/json  
Cookie: token={{auth\_token}}

**Тело запроса: -**

**Шаги выполнения:**  
1) Выбрать метод DELETE, указать URL https://restful-booker.herokuapp.com/booking/{{auth\_bookingid}}  
2) Установить заголовок Content-Type: application/json  
3) Добавить авторизационный токен в Cookie ( Post / Auth )  
4) Отправить запрос  
5) Проверить ответ

**Ожидаемый результат:**  
1) Статус ответа: 201 Created

**Фактический результат:**  
1) Статус ответа: 201 Created

**Статус:** Pass

**Комментарии:** -

**ID: 20**  
**Название теста:** Удаление бронирования с несуществующим ID  
**Тип сценария:** Негативный

**Предусловия:**  
1) API доступен  
2) Токен авторизации сохранён и передаётся в Cookie  
3) Установлен Content-Type: application/json  
4) ID бронирования не существует (например, 0)

**Эндпоинт:** DELETE /booking/0

**Заголовки запроса:**  
Content-Type: application/json  
Cookie: token={{auth\_token}}

**Тело запроса:** -

**Шаги выполнения:**  
1) Выбрать метод DELETE, указать URL <https://restful-booker.herokuapp.com/booking/0>  
2) Установить заголовок Content-Type: application/json  
3) Добавить авторизационный токен в Cookie ( Post / Auth)  
4) Отправить запрос  
5) Проверить ответ

**Ожидаемый результат:**  
1) Статус ответа: 405 Method Not Allowed

**Фактический результат:**  
1) Статус ответа: 405 Method Not Allowed

**Статус:** Pass

**Комментарии:** -

**ID: 21**  
**Название теста:** Фильтрация списка бронирований по имени (firstname=John)  
**Тип сценария:** Позитивный

**Предусловия:**  
1) API доступен  
2) В системе есть хотя бы одно бронирование с firstname = "John"  
3) Метод GET доступен без авторизации

**Эндпоинт:** GET /booking?firstname=John

**Заголовки запроса:**  
Content-Type: application/json

**Тело запроса:** -

**Шаги выполнения:**  
1) Выбрать метод GET, указать URL <https://restful-booker.herokuapp.com/booking?firstname=John>  
2) Установить заголовок Content-Type: application/json  
3) Отправить запрос  
4) Проверить статус и тело ответа

**Ожидаемый результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит массив объектов с bookingid

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит список bookingid

**Статус:** Pass

**Комментарии:** -

**ID: 22**  
**Название теста:** Фильтрация с некорректными значениями даты (checkin=n, checkout=m)  
**Тип сценария:** Негативный

**Предусловия:**  
1) API доступен  
2) Метод GET /booking поддерживает фильтрацию по параметрам checkin и checkout  
3) Передаются некорректные значения дат (буквенные символы)

**Эндпоинт:** GET /booking?checkin=n&checkout=m

**Заголовки запроса:**  
Content-Type: application/json

**Тело запроса:** -

**Шаги выполнения:**  
1) Выбрать метод GET, указать URL <https://restful-booker.herokuapp.com/booking?checkin=n&checkout=m>  
2) Установить заголовок Content-Type: application/json  
3) Отправить запрос  
4) Проверить статус ответа и тело

**Ожидаемый результат:**  
1) Статус ответа: 400 Bad Request

**Фактический результат:**  
1) Статус ответа: 500 Internal Server Error

**Статус:** Fail

**Комментарии:** Сервер не обрабатывает некорректные значения даты - Баг

**ID: 23**  
**Название теста:** Фильтрация с некорректными значениями даты (checkin=2023-00-00, checkout=0000-00-00)  
**Тип сценария:** Негативный

**Предусловия:**  
1) API доступен  
2) Метод GET /booking поддерживает фильтрацию по параметрам checkin и checkout  
3) Передаются значения дат в невалидном формате (нулевые или несуществующие значения)

**Эндпоинт:** GET /booking?checkin=2023-00-00&checkout=0000-00-00

**Заголовки запроса:**  
Content-Type: application/json

**Тело запроса:** -

**Шаги выполнения:**  
1) Выбрать метод GET, указать URL <https://restful-booker.herokuapp.com/booking?checkin=2023-00-00&checkout=0000-00-00>  
2) Установить заголовок Content-Type: application/json  
3) Отправить запрос  
4) Проверить статус ответа и тело

**Ожидаемый результат:**  
1) Статус ответа: 400 Bad Request

**Фактический результат:**  
1) Статус ответа: 500 Internal Server Error

**Статус:** Fail

**Комментарии:** Сервер не обрабатывает значения – Баг

**ID: 24**

**Название теста:** Фильтрация по дате checkin и checkout (валидные значения)  
**Тип сценария:** Позитивный  
**Предусловия:**  
1) API доступен  
2) Существуют бронирования с датами checkin: 2000-01-01 и checkout: 2021-01-01  
3) Метод GET /booking поддерживает фильтрацию по параметрам checkin и checkout  
**Эндпоинт:** GET /booking?checkin=2000-01-01&checkout=2021-01-01  
**Заголовки запроса:**  
Content-Type: application/json  
**Тело запроса:** -

**Шаги выполнения:**  
1) Выбрать метод GET, указать URL <https://restful-booker.herokuapp.com/booking?checkin=2000-01-01&checkout=2021-01-01>  
2) Установить заголовок Content-Type: application/json  
3) Отправить запрос  
4) Проверить статус ответа и тело

**Ожидаемый результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит массив с одним или несколькими объектами

**Фактический результат:**  
1) Статус ответа: 200 OK  
2) Тело ответа содержит массив с объектами

**Статус:** Pass  
**Комментарии:** -