**IT Academia**

**Restful Booker**

**Тест план**

**Версия 1.0**

**Купрейчик Татьяна**

**Вступление**

**Что мы будем делать?**

Тестирование API бекенда restfull приложения Booker. Суть приложения заключается в том, чтобы пользователь через наш сервис мог оформлять, менять и отменять бронь номеров.

**Зачем?**

Тестирование API необходимо, чтобы выловить все ошибки разработки до этапа передачи приложения конечному пользователю. Чтобы пользователь мог нормально пользоваться всеми функциями приложения без обращения в техподдержку.

**Для какого клиента?**

Для человека, кто хочет забронировать номер в отеле

**Кто будет использовать продукт**

Люди старше 18 лет, туристы

**Для чего будет использоваться продукт?**  
Через продукт пользователь будет осуществлять бронирование номеров в отеле

**Фокус работы**

Компоненты и функции, которые должны быть протестированы:

API restful-booker, сотоящее из трех модулей ( auth, booking, ping )

* **auth** состоит из одного запроса:

название - *createToken*

тип - post

линк - https://restful-booker.herokuapp.com/auth

описание - Создает новый токен авторизации для доступа к обновления и удаленияю booking

* **booking** состоит из следующих запросов:

название - *GetBookingIds*

тип - get

линк - https://restful-booker.herokuapp.com/booking

описание - Возвращает идентификаторы всех бронирований, существующих в API. Может принимать необязательные строки запроса для поиска и возврата подмножества идентификаторов бронирования.

название - *GetBooking*

тип - get

линк - https://restful-booker.herokuapp.com/booking/:id

описание - Возвращает конкретное бронирование на основе предоставленного идентификатора бронирования

название - *CreateBooking*

тип - get

линк - https://restful-booker.herokuapp.com/booking

описание - Создает новое бронирование в API

название - *UpdateBooking*

тип - put

линк - https://restful-booker.herokuapp.com/booking/:id

описание - Обновляет текущее бронирование

название - *PartialUpdateBooking*

тип - patch

линк - https://restful-booker.herokuapp.com/booking/:id

описание - Обновляет текущее бронирование с частичной полезной нагрузкой

название - *DeleteBooking*

тип - delete

линк - https://restful-booker.herokuapp.com/booking/1

описание - Возвращает идентификаторы всех бронирований, существующих в API. Может принимать необязательные строки запроса для поиска и возврата подмножества идентификаторов бронирования.

* **ping** состоит из одного запроса:

название - *HealthCheck*

тип - get

линк - HealthCheck

описание - Простая конечная точка проверки работоспособности, чтобы убедиться, что API запущен и работает.

Для тестирования нам понадобятся следующие программные инструменты: postman, testrail, qase, jmeter, jira, компьютер на windows (10 или новее) / macOS (12.4 или новее), java (7 или новее)

**Качество и критерии приемки**

**Критерии приемки:**

1. Все функциональности приложения должны быть покрыты тестами.
2. При тестировании всех функциональностей приложения, фактические результаты всех тест кейсов должны совпасть с ожидаемыми
3. 80% функциональностей приложения должно успешно пройти тестирование
4. Допускается 20% функциональностей, частично или полностью не прошедших тестирование

**Критические факторы успеха**

Чтобы проект завершился успешно, должны быть выполнены следующие требования:

1. Должен быть предоставлен полноценный доступ в багтрекинговую систему и тест менеджмент систему
2. Требования не должны меняться в процессе разработки и тестирования API
3. Команда разработки должна сдавать приложение в сроки
4. Догововеренности должны сохраняться до конца тестирования

**Оценка риска**

1. Разработка может просрочить дедлайны (Предотвращается грамотно просчитанным временем на выполнение и запасом времени на непредвиденные задержки)
2. Могут возникнуть проблемы с тест девайсами (протестировать девайсы заранее на предмет неисправностей и если что заменить)
3. Приложения могут не работать на тест девайсах (Узнать все ограничения и условия работы приложения у разработчиков заранее, чтобы выбрать нужные девайсы)
4. У тестировщика недостаточно квалификации для проверки всех функциональностей приложения (Узнать заранее, что это за приложение и что в нем придется тестировать, обучиться необходимым навыкам)
5. Тестировщик может заболеть (иметь запас по времени и средства для быстрого лечения)

**Ресурсы**

**Временные**: 30 дней

**Человеческие**: 1 тестировщик

**Аппаратные**: Macbook Pro 14, 16GB, Intel 7 x64

**Программные**: Mac OS Monterey 12.4, qase, testrail, jira, postman,

jmeter, google chrome

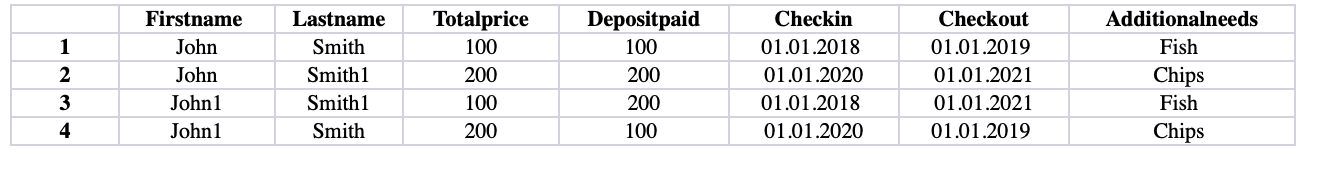
**Тест документация**

Тест документация состоит из следующих частей:

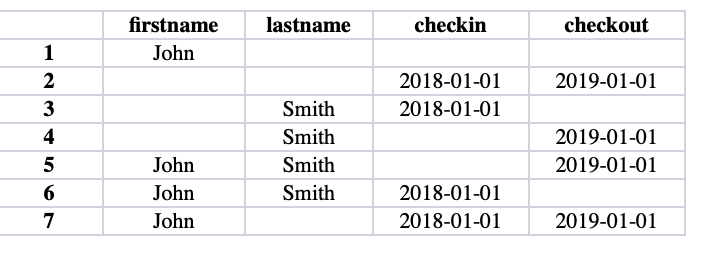
1. Документация API со всеми возможными запросами (вкл. все необходимые данные и примеры)
2. Тест план
3. Тест кейсы
4. Баг репорты
5. Отчет о результатах тестирования(TRR)

**Тест стратегия**  
В проекте должны использовальзоваться техники тест дизайна, в частности - попарное тестирование

Вот примеры использования с PUT



Примеры использования с PATCH



Благодаря этим техникам удастся максимально эффективно покрыть тестами функционал PUT и PATCH методов API

Также должны использоваться таблицы принятия решений

Для авторизации (с логином и паролем)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Логин** | **Из документации** | **Из документации** | **Случаный, но валидный** | **Случаный, но валидный** |
| **Пароль** | **Из документации** | **Случаный, но валидный** | **Из документации** | **Случаный, но валидный** |
| **Результат** | **Получен новый токен** | **Bad credentials** | **Bad credentials** | **Bad credentials** |

**Нефункциональные проверки:**

1. Проверить firstname / lastname / additionalneeds с помощью SQL injection
2. Проверить firstname / lastname / additionalneeds с помощью XSS injection

**Расписание тестирования**

1. Изучение документации ( 16-18 января)
2. Составление тест плана (19-25 января)
3. Создание тест кейсов (с 26 января по 5 февраля)
4. Тест ран (6 февраля)
5. Составление отчетов тестирования (7-8 февраля)
6. Залитие всех документов на git (9 февраля)