**Тестовое задание на позицию «Специалист по автоматизированному тестированию»**

Допустим, у нас есть вебсайт, на котором мы продаем товар. Клиенты выбирают нужные им товары и формируют заказ, при этом оставляя свой телефонный номер. Далее наш оператор им звонит и предлагает оплатить заказ через банковский перевод, либо переводом на карту или счет. Однако, сайт конкурентов уже начал принимать оплату онлайн по номеру телефона и владелец нашего сайта решил, что нам нужно аналогично сделать, так как клиенты стали уходить.

Аналитики выяснили ,что удобнее и быстрее всего это сделать через платежного провайдера «МногоПэй». Для интеграции под наши требования «МногоПэй» разработает API сервис и нам останется только реализовать у себя на сайте вебформу для оплаты, которая будет по этому API протоколу взаимодействовать. Оплата происходит по номеру телефона. Клиент вводит свой номер телефона при оплате заказа, ему приходит СМС-сообщение с просьбой подтвердить платеж ответным сообщением. Клиент отправляет подтверждение и платеж проходит, либо не проходит и получаем сообщение с ошибкой.

Собственно на данном этапе разработки наша команда, которая разрабатывает платежную вебформу, немного запаздывает, ключевой разработчик в отпуске. Но «МногоПэй» уже реализовала API для работы с будущей платежной формой, и передала в тестирование.

Мы можем начать тестирование, так как задача срочная.

Тестировщику в рамках задачи нужно проверить, что реализация API соответствует описанию в техническом задании (Functional Requirements):

- написать тестовые кейсы для проверки функционала используя техники тест-дизайна (например в excel или google таблицах)

- выполнить тестовые кейсы используя инструмент postman или insomnia\*, оформить найденные баги в любом текстовом документе (MS Word или google Doc)

- реализовать как минимум два-три автотеста по написанным тестовым кейсам, эти тесты нужны для последующего регрессионного тестирования новых релизов.

Автотесты можно реализовать в любом удобном варианте на языке php. Можно обычными скриптами на php формировать, отправлять запросы через CURL на сервер «МногоПэй» и обрабатывать то, что приходит в ответ. И проверять, конечно же ожидаемый результат.

**PS:** Также, при желании, можно использовать тестовые фреймворки. Будет большим плюсом, установить себе фреймворк codeception и реализовать на нём через модуль REST и соответствующие методы <https://codeception.com/docs/10-APITesting>.

**PS:** Если есть желание, можно все тесткейсы автоматизировать, тогда проверять через postman или insomnia не нужно. Но перед каждым автотестом нужны комментарии, должно быть понятно что именно он проверяет.

**Ниже документация на API, которую мы отдавали «МногоПэй» и они на ее основе реализовали.**

**Техническое задание**

Документ описывает взаимодействие платежной формы вебсайта по продаже билетов в театр и вебсервиса по проведению платежей «МногоПэй».

FR1: Для передачи данных между платежной страницей и сервисом «МногоПэй» используются POST запросы.

FR2: Формат данных для передачи в теле запроса — JSON.

FR3: Для того чтобы сервер принимал запросы, нужно в заголовке HTTP запроса передавать специальный секретный ключ в параметре secret: QWERTY000097531KEY

FR4: Для проведения оплаты по номеру телефона нужно на URL адрес: *https://mock01.ecpdss.net/mnogopay/z/buy*

передать следующие параметры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Параметр** | **Тип данных** | **Обязательный** | **Описание** | **Пример значения** |
| amount | Integer | + | Сумма | 3 |
| currency | Enum | + | Код валюты EUR или RUB | RUB |
| order\_id | String | + | Номер заказа | NTS2234443 |
| type\_order | Integer | + | Тип заказа | 1 |
| day | Формат даты 00-00-0000 | + | Дата доставки товара | 21-02-2021 |
| buyer\_first\_name | string | + | Имя покупателя | Oleg |
| buyer\_last\_name | string | - | Фамилия покупателя | Testow |
| comment | string | - | Дополнительный комментарий | Something comment |
| phone | Integer (11) | + | Телефон покупателя | 79006005040 |
| email | string | - | Email покупателя | test@email.ee |
| notification | bool | - | Оповещение пользователя | true |
| address | string | +/- | Адрес для доставки товара | Some street number 81 |

описание типов заказа:

|  |  |  |
| --- | --- | --- |
| **Type\_order** | **Название** | **описание** |
| 1 | Самовывоз | Тип заказа подразумевающий обязательное наличие параметра **address** |
| 2 | Доставка | Тип заказа подразумевающий обязательное наличие параметра **address** |
| 3 | Бронирование | Подразумевает временное бронирование товара на складе, дальнейшие действия над товаром обговариваются индивидуально с менеджером после его звонка |

Сервер должен валидировать эти параметры. Если все верно, сервер должен в ответ выдавать ответ:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Параметр** | **Обязательность** | **Тип** | **Формат** | **Описание** | **Пример значения** |
| payment\_id | Обязательный | String | Любой | ID операции на сервере, должен быть уникальным | 157AAA05708228 |
| order\_id | Обязательный | int | До 32 чисел включительно | Номер заказа | NTS2234443 |
| resp\_code | Обязательный | int | Три числа | Код ответа  0 — успешная оплата  111 - Incorrect parameter  222 - "Required parameter is absent  333 - It is too much amount | 0 |
| check | Обязательный | String | До 100 символов | Показывает куда был отправлен чек по операции, появляется только если передано значение notification = true в запросе | Phone = 79007005040, email = test@ema.il |
| notification | Обязательный | bool | True / false | Показывает был ли отправлен чек на емейл или телефон | true |

Если операция не прошла — выдавать ошибки с описанием проблемы.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Параметр** | **Обязательность** | **Тип** | **Формат** | **Описание** | **Пример значения** |
| resp\_code | Обязательный | int | от 1 до 3 чисел | Код ответа | 111 |
| message | Опциональный | String | До 32 символов включительно | Сообщение об ошибке | Incorrect amount |

Пример запроса/ответа от сервера:

Запрос   
{

"amount": 333,

"currency": "RUB",

"order\_id": "testorder222",

"type\_order": 1,

"day": "24-12-2021",

"buyer\_first\_name": "rion",

"buyer\_last\_name": "basov",

"comment": "buy something",

"phone": 89996651212,

"email": "test@emails.ru",

"notification": true,

"address": "Livni"

}

Ответ:

{

"payment\_id": "payment\_202101281024",

"order\_id": "testorder222",

"resp\_code": "0",

"check": "phone=89996651212,email=test@emails.ru",

"notification": **true**

}