**Инструкция к POSTMAN для использования**

**с протоколом API Orange Data**

Оглавление

[1. Введение 3](#_Toc62464346)

[2. Нaстройка POSTMAN 3](#_Toc62464347)

[Выбор сред Orange Data, для импорта Swagger настройки Postmana 3](#_Toc62464348)

[Настройка сертификатов и ключей 5](#_Toc62464349)

[3. Работа с POST запросами 8](#_Toc62464350)

[Генерация тестовых ключей с использованием SignatureGenerator 8](#_Toc62464351)

[Заполнение Headers в POST 9](#_Toc62464352)

[4. Работа с GET запросами 12](#_Toc62464353)

[GET запрос 12](#_Toc62464354)

# Введение

Тестирование API относится к интеграционному тестированию, а значит в ходе него можно отловить ошибки взаимодействия между модулями системы или между системами. Для тестирования используют специальные инструменты, где можно отправить входные данные в запросе и проверить точность выходных данных. Одним из таких инструментов является Postman.

В приложении Postman созданы коллекции с запросами к API Orange Data. Открыв коллекцию, сможете разобраться в работе нашего сервиса. Можете писать тесты и производить автоматизированное тестирование прямо из Postman. А спецификации Swagger JSON выбранной среды, создаст коллекции по GET и POST запросам, учитывающие необходимые фискальные требования закона 54-ФЗ по формированию чека кассового аппарата.

# Нaстройка POSTMAN

### Выбор сред Orange Data, для импорта Swagger настройки Postmana

Для использования Postmana, при тестировании запросов API Orange Data, необходимо использовать среды. Orange Data имеет две тестовые среды таблица 1 и одну производственную среду таблица 2. Тестовая среда 1 использует эмулятор, дает возможность тестирования запросов API, без проверки чеков в ОФД. Тестовая среда 2 использует МГМ для фискализации запросов в чек. С последующей передачей чеков в тестовый личный кабинет ОФД. Производственная среда использует фискальный накопитель, с передачей чеков в ОФД и дальнейшим предоставлением чеков в ФНС. Более подробное описание сред, изложено в «Протоколе взаимодействий с интернет кассой Orange Data». Протокол взаимодействия с интернет кассой и файлы для тестирования вы найдете по ссылке <https://github.com/orangedata-official/API> . Готовые варианты интеграций, доступны по ссылке

<https://github.com/orangedata-official>

Таблица 1

|  |  |  |
| --- | --- | --- |
| Параметр | Тестовая среда 1 | Тестовая среда 2 |
| Базовый URL | [https://apip.orangedata.ru:2443](https://apip.orangedata.ru:2443/) | [https://apip.orangedata.ru:12001](https://apip.orangedata.ru:12001/) |
| IP-адрес | 94.228.252.55 | 94.228.252.55 |
| Веб-кабинет | Недоступен | https://test.orangedata.ru/lk/ |
| URL-адрес проверки доступности для браузеров | https://apip.orangedata.ru:2443/api/v2/ | https://apip.orangedata.ru:12001/api/v2/ |
| Swagger | <https://apip.orangedata.ru:2443/swagger> | <https://apip.orangedata.ru:12001/swagger> |
| Swagger JSON спецификация | <https://apip.orangedata.ru:2443/swagger/v2/swagger.json> | <https://apip.orangedata.ru:12001/swagger/v2/swagger.json> |

Таблица 2

|  |  |
| --- | --- |
| Параметр | Производственная среда |
| Базовый URL | [https://api.orangedata.ru:12003](https://api.orangedata.ru:12003/) |
| IP-адрес | 62.76.112.48 и 188.170.11.161 |
| Веб-кабинет | [https://lk.orangedata.ru](https://lk.orangedata.ru/) |
| URL-адрес проверки доступности для браузеров | Настоятельно рекомендуется использовать доменное имя для производственных запросов. Добавить /api/v2/ в бар URL, чтобы проверить доступность |
| Swagger-ui | <https://api.orangedata.ru:12003/swagger> |
| Swagger JSON спецификация | <https://apip.orangedata.ru:12003/swagger/v2/swagger.json> |
|  |  |

Выбрав необходимую среду из таблицы 1 или 2, переходим по **URL Swagger JSON спецификация** и сохраняем страницу рис. 1 для импорта в Postman рис. 2.

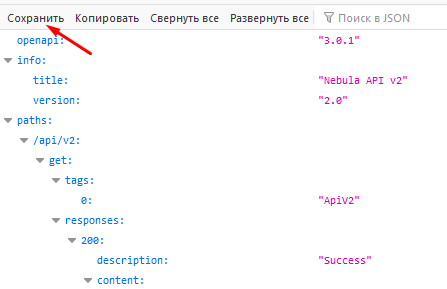
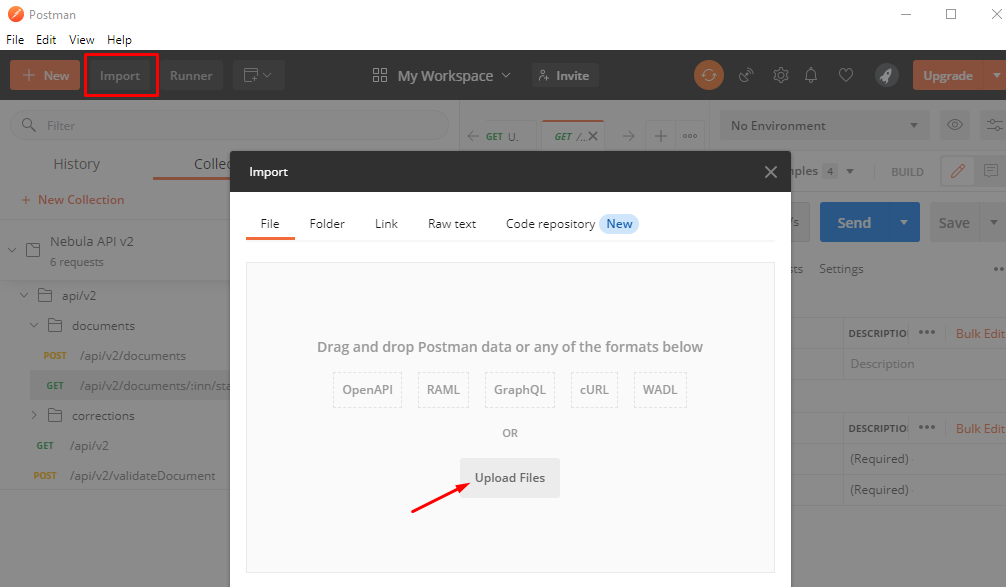


Рисунок 1

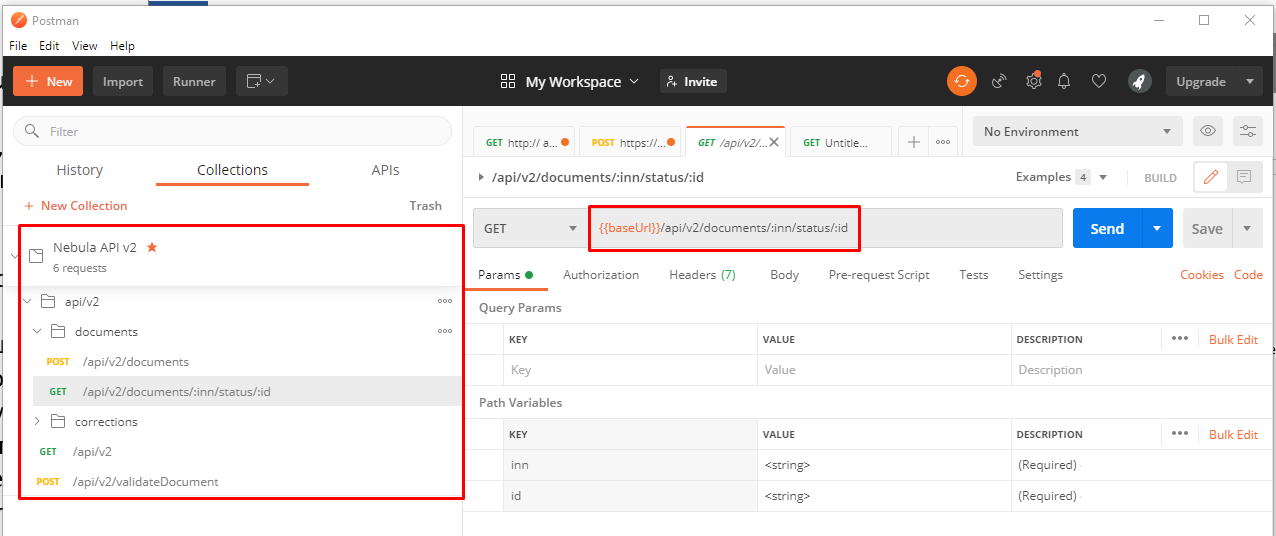
Выбрав в верхней консоли вкладку **Import**, в открывшемся окне рис. 2, нажимаем кнопку **Upload Files**. Выбрав файл **Swagger**, сохраняем данные страницы с Swagger настройками выбранной среды.

Рисунок 2



В **Collections** можно выбрать **GET** и **POST** запросы, а также **POST запросы по чеку коррекции** рис.3

Рисунок 3



### Настройка сертификатов и ключей

Запрос в OrangeData, на отправку данных, должен быть подписан. В то же время – запрос GET на получение статуса, не нуждается в подписи. С более подробной информацией о подписании документа, Вы можете ознакомится в «Протоколе взаимодействий с интернет кассой Orange Data».

Каждый клиент генерирует свой собственный сертификат SSL.Этот сертификат SSL обеспечивает безопасное соединение между приложением клиента и OrangeData.В производственной среде SSL-сертификат генерируется в личном кабинете OrangeData https://lk.orangedata.ru .

В двух тестовых средах используется тестовый сертификат по умолчанию. Вы можете скачать тестовый сертификат из Git <https://github.com/orangedataofficial/API/tree/master/files_for_test/client_ca.crt>

**Описание файлов для производственной среды**:

**"ИНН".crt**, **"ИНН".key** - Сертификат клиента и закрытый ключ клиента сгенерированный на основе **client\_ca.crt** (пароль 1111)

**client\_ca.crt** - Корневой сертификат Orangedata (с открытым ключом внутри)

**private\_key.xml** - Закрытый ключ клиента для подписания запроса.

**cacert.pem** - набор корневых сертификатов (пароль 1111)

**Nebula.KeysGenerator**-1.2.0.0.zip - Генератор ключей для работы с продуктовой средой Orange Data. После генерации получается ключ подписи, состоящий из двух частей: приватной и публичной. Клиенты генерируют ключ подписи и передают нам публичную часть.

В двух тестовых средах используется тестовый сертификат. В обеих тестовых средах (1 и 2) используется https соединение с сертификатами. Все сертификаты и ключи для тестовых сред (1 и 2) неизменны, публикуются в Git <https://github.com/orangedata-official/API/tree/master/files_for_test>

**Описание файлов file\_for\_test GIT для тестовой среды:**

**client.crt**- **client.key** – Сертификат клиента и закрытый ключ клиента сгенерированы на основе корневого сертификата **client\_ca.crt** (пароль 1234)

**client\_ca.crt** - Корневой сертификат Orangedata (с открытым ключом внутри).

**cacert.pem** - набор корневых сертификатов

**private\_key\_test.xml** - Закрытый ключ клиента для подписи запросов

ТАБЛИЦА 3

|  |  |
| --- | --- |
| Файл | Описание |
| client\_ca.crt | Корневой сертификат Orangedata (с открытым ключом внутри). Все сертификаты клиентов подписываются закрытым ключом, соответствующим данному сертификату. Если сертификат клиента был подписан любым другим эмитентом - он будет отклонен сервером |
| client.key | Закрытый ключ клиента для подключения SSL для тестовой среды |
| client.crt | Сертификат клиента для тестовой среды (подписанный закрытым ключом, соответствующим корневому сертификату **client\_ca.crt**). |
| client.pfx | Сертификат PFX клиента для тестовой среды. Файлы формата PFX содержат одновременно и открытый ключ и закрытый. Т.е. такой файл соответствует одновременно двум файлам – **client.key** и **client.crt** и может быть получен из этих двух файлов при помощи утилиты openssl. В файле **client.pfx** используется пароль 1234.  В зависимости от используемого Вами программного обеспечения, Вам может потребоваться либо файл **client.pfx**, либо Вы можете использовать **client.key** и **client.crt.** |
| private\_key\_test.xml | Закрытый ключ клиента для подписания запроса. Более подробная информация описана в «Протоколе взаимодействий с интернет кассой Orange Data» |
| cacert.pem | набор корневых сертификатов |

**SignatureGenerator**-программа для генерации подписи **X-Signature** при отправке запросов к API OrangeData через POSTMAN (в качестве тестов)

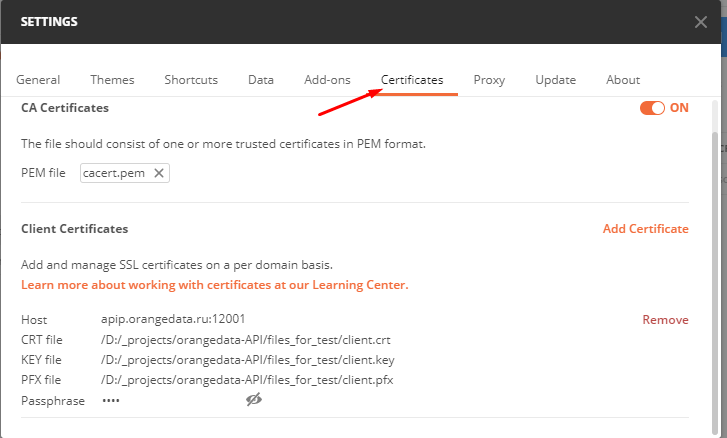
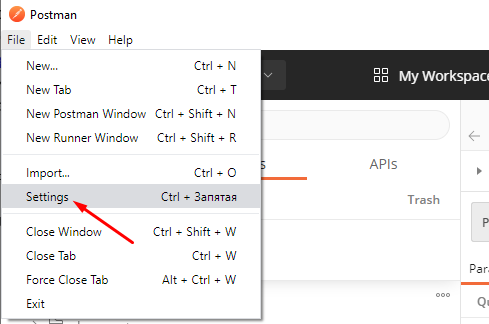
Описание файла в каталоге GIT **SignatureGenerator** таблица 4.

Таблица 4

|  |  |
| --- | --- |
| Файл | Описание |
| SignatureGenerator.exe.config | Подпись генератора config файл |
| SignatureGenerator.exe | Генератор подписи |
| SignatureGenerator.pdb | Файл данных генератора подписей |

Для передачи запросов в Postman, необходимо добавить сертификат и ключ, для тестовой среды или для производственной среды. В зависимости от того. какими средами будете пользоваться.  
Для этого в меню **File**, выбираем **Settings** и переходим в раздел **Certificates** рис. 4.

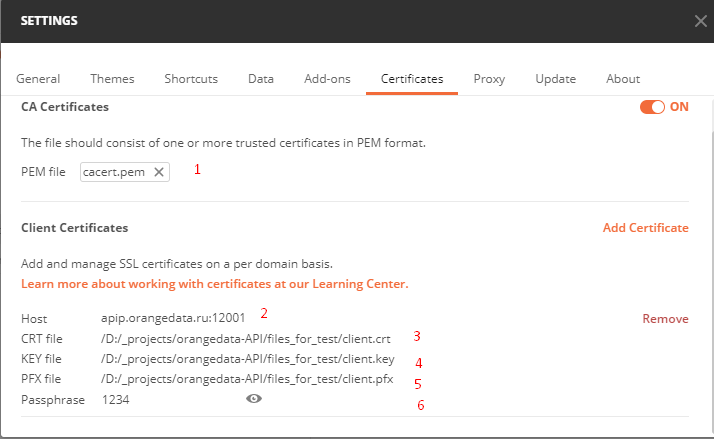
Рисунок 4



Далее в примере мы используем сертификат и ключи для тестовой среды 2 с МГМ рис. 5.

1. **cacert.pem** набор корневых сертификатов
2. Базовый URL: **apip.orangedata.ru:12001**
3. **client.crt**
4. **client.key**
5. **client.pfx**
6. пароль **1234**

Рисунок 5



# Работа с POST запросами

Взаимодействие с интернет-кассой осуществляется посредством обмена сообщениями с web-сервисом. Транспорт – https с использованием публичным(клиентским) сертификатом клиента и серверного сертификата, для взаимной аутентификации. Формат сообщения — JSON. Размер JSON не имеет ограничений, но ФН имеет аппаратное ограничение 32768 байт для каждого финансового документа в двоичной форме. Это ограничение размера документа предусматривает около 250 позиций в чеке. ФН отклонит ваш чек, если вы превысите двоичное ограничение 32768 байтов. Мы рекомендуем использовать менее 250 позиций в чеке.

### Генерация тестовых ключей с использованием SignatureGenerator

Все запросы POST, подписываются закрытым ключом клиента. Для генерации тестового ключа запроса используем SignatureGenerator, программу для генерации подписи X-Signature при отправке тестовых запросов к API OrangeData через POSTMAN (в качестве тестов).

SignatureGenerator необходимо скачать по адресу <https://github.com/orangedata-official/API>.

Подпись SHA256-RSA с ключом не менее 2048 Кбит/с и Pkcs1 используется для защиты клиентских запросов. Подпись передается в http-заголовке **X-Signature**, закодированным с использованием кодировки base64. Для обеих тестовых сред (1 и 2) **private\_key\_test.xml** должен использоваться из Git [**https://github.com/orangedata-official/API/tree/master/files\_for\_test**](https://github.com/orangedata-official/API/tree/master/files_for_test)

Существует генерация подписи в **php** с [**https://github.com/phpseclib/phpseclib**](https://github.com/phpseclib/phpseclib), а также генерация подписей в **C#** с использованием стандартной библиотеки класса **.net core**. Более подробная информация изложена в «Протоколе взаимодействий с интернет кассой Orange Data».

Генерация подписи для тестовых сред, может быть выполнено приложением **Signature Generator**.

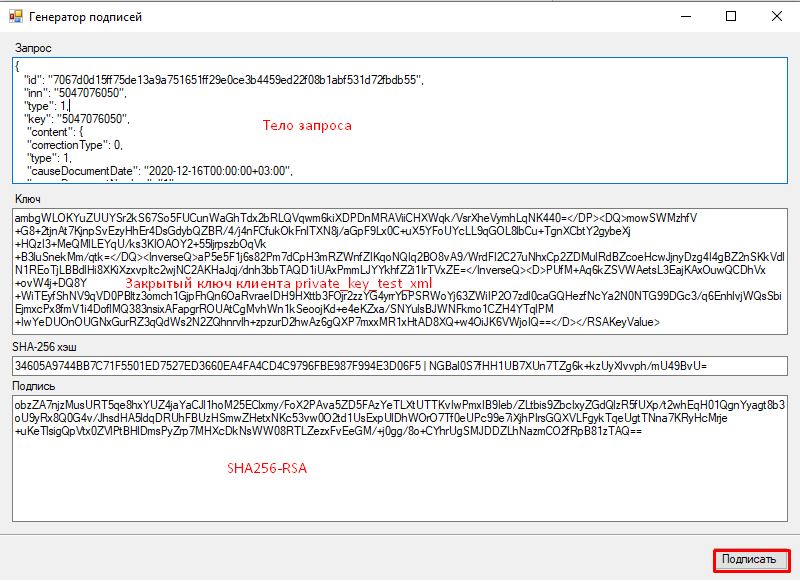
В таблице **SignatureGenerator** (генератора подписей) рис.6

Тело запроса помещаем в верхнее окно-**Запрос**.

Закрытый ключ клиента **private\_key\_test.xml** помещаем окно-**Ключ**

Нажимаем на кнопку **Подписать**. Генерируем ключ **SHA256-RSA**, для подписи POST запроса.

Рисунок 6



### Заполнение Headers в POST

Выберите в **Collections**, в папку **Nebula API v2**. За тем выберете в папке **api/v2 POST** запрос. В открывшемся окне **POST** во вкладке **Headers**, заполняем заголовки как показано на рис. 7.

**Cache-Control** no-cache

**Postman-Token** <calculated when request is sent>

**Content-Type** application/json

**Content-Length** <calculated when request is sent>

**Host** <calculated when request is sent>

**Accept** \*/\*

**Accept-Encoding** gzip, deflate, br

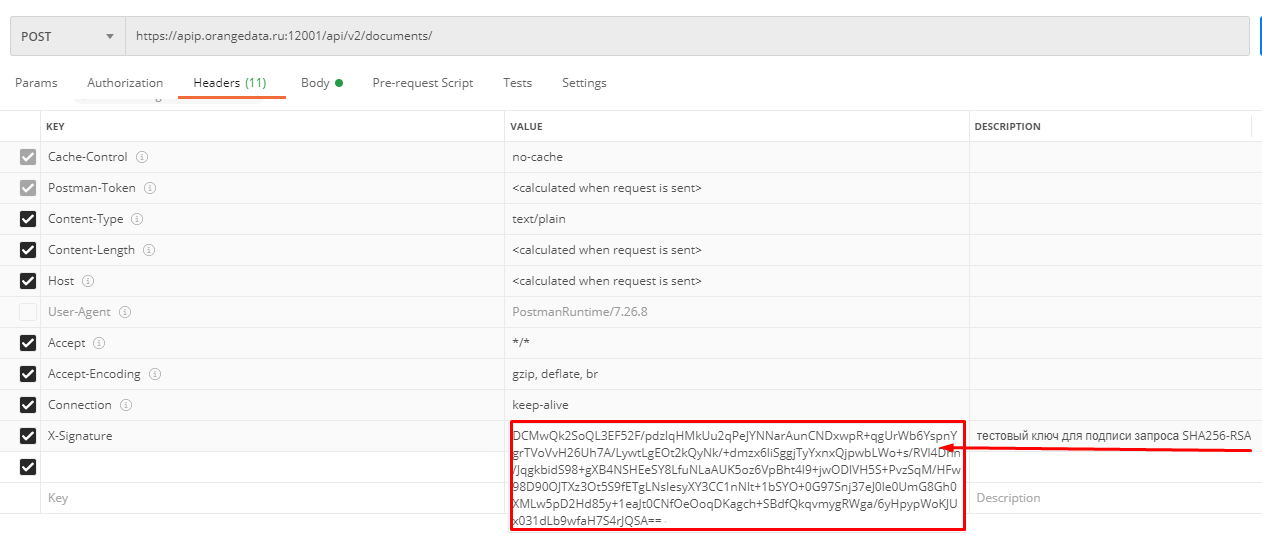
**Connection** keep-alive

**X-Signature** сгенерированный Signature Generator, тестовый ключ для подписи запроса SHA256-RSA

Подпись должна быть добавлена к параметру **HEADER X-Signature** рис. 7.

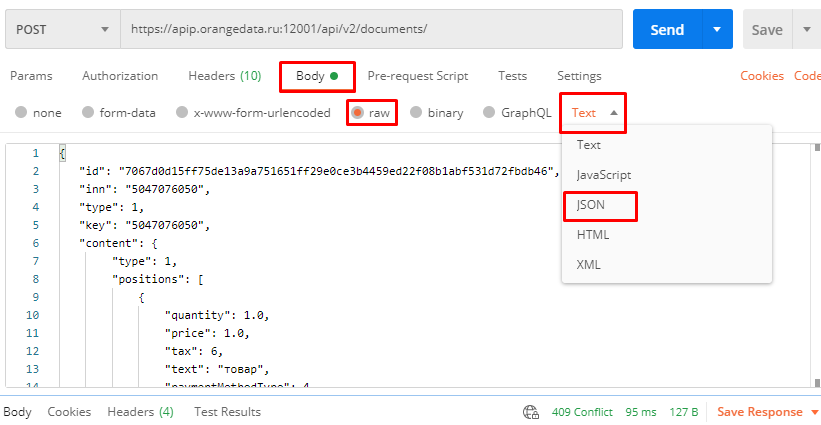
Подпись SHA256-RSA с ключом не менее 2048 Кбит/с и Pkcs1 используется для защиты клиентских запросов. Подпись передается в http-заголовке **X-Signature**, закодированным с использованием кодировки base64. Для обеих тестовых сред (1 и 2) **private\_key\_test.xml** должен использоваться из Git [**https://github.com/orangedata-official/API/tree/master/files\_for\_test**](https://github.com/orangedata-official/API/tree/master/files_for_test)

Рисунок 7



Перейдите в подраздел **Body** и выберете вкладку **raw**, переведите **Text** в **JSON** что бы указать формат Ваших данных рис. 8.

Рисунок 8



В поле URL запроса {{baseUrl}}/api/v2/**validateDocument?signature=<string>** рис. 9, введите

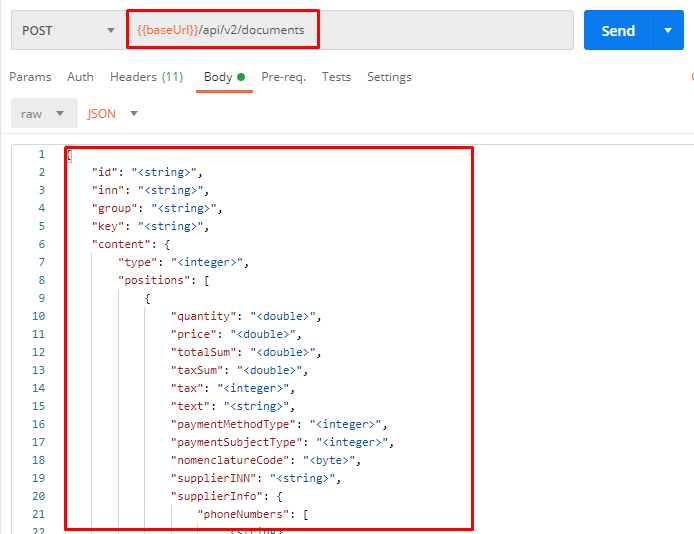
{{baseUrl}} - Базовый URL из таблиц тестовых сред (1 или 2) или производственной среды.

**validateDocument?signature=<string>** - /api/v2/documents/

О необходимых параметрах тела запроса можете узнать в нашем «Протоколе взаимодействий с интернет кассой Orange Data»

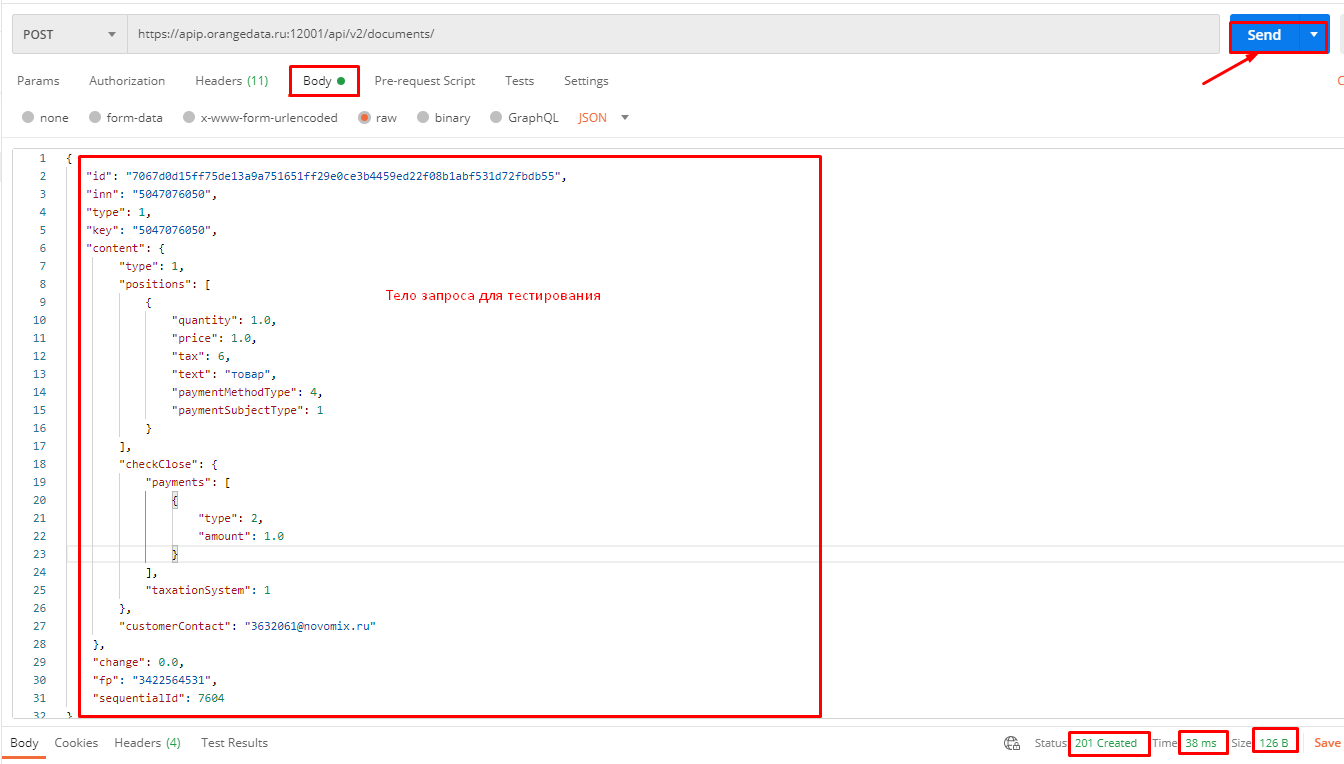
После проведеных настроек, в открывшемся окне **Body**, создаем тело запроса или копируем из своего примера.

Рисунок 9



Нажимаем кнопку **Send**, в статусе ответа запроса рис. 10, видим код 201, время ответа, размер запроса.

Рисунок 10



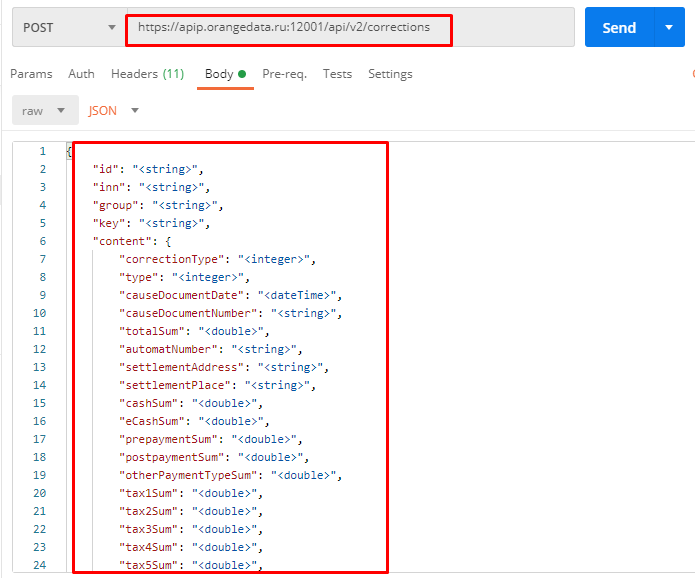
Для POST запросов, на создание чека коррекции, переходим в **Collections** в папку **Nebula API v2**. Выбираем подраздел **corrections**. В поле URL {{baseUrl}}**/api/v2/corrections**.

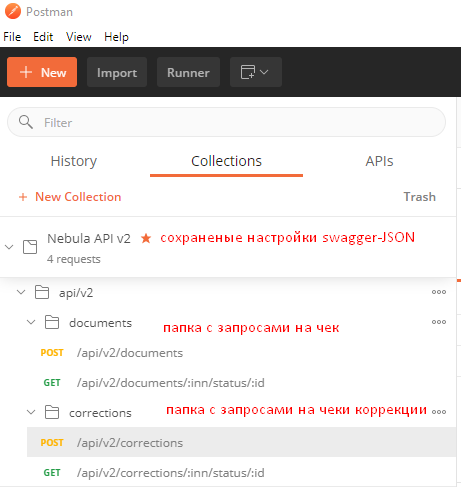
{{baseUrl}} Базовый URL из таблиц тестовых сред (1 или 2) или производственной среды.

В окне **Body** создаем тело запроса или копируем уже готовый запрос рис 11.

Нажимаем кнопку **Send**, в статусе ответа запроса видим код 201, время ответа, размер запроса.

Рисунок 11





В ответ: api может возвращать следующие http статус-коды:

201 Created – чек создан и добавлен в очередь на обработку, пустое тело ответа

401 Unauthorized – клиентский сертификат не прошел проверку

409 Conflict– чек с данным идентификатором уже был создан в системе, пустое тело ответа

400 Bad Request – переданные данные содержат ошибки валидации, либо подпись не прошла проверку, тело ответа.

503, Service Unavailable – очередь документов переполнена, в ответе возвращается header Retry-After с таймаутом в секундах, через который стоит повторить запрос, тело ответа.

# Работа с GET запросами

### GET запрос

Каждый запрос, который вы отправляете в Postman, требует URL-адреса API, с которым вы работаете. Запрос в API доступен по определенному URL-адресу—это то, что вы вводите в Postman для доступа к API.Запросы используют для получения информации с данного URL: **/api/v2/documents/{inn}/status/{document\_id}**. При этом не будет внесено ни каких изменений, по этому адресу API. В **Collections** в папке **Nebula API v2**, подраздел **documents**, выбираем **GET**.

В поле URL запроса {{baseUrl}}/api/v2/documents**/{inn}/**status/**{document\_id}** введите:

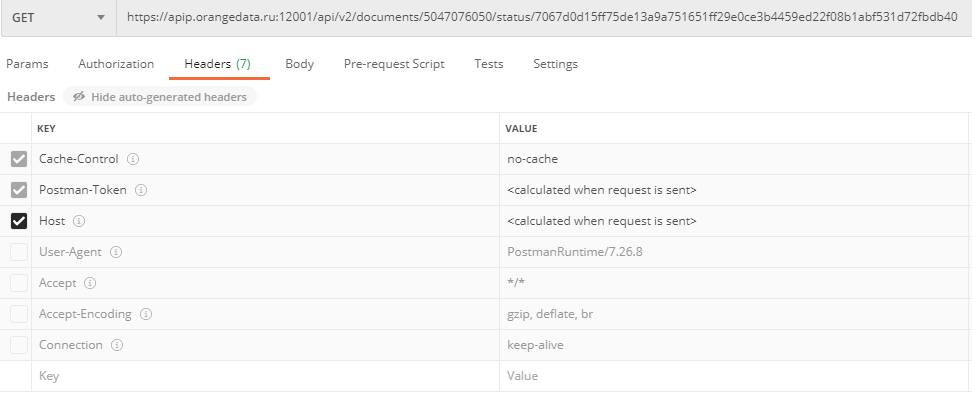
{{baseUrl}} – Базовый URL из таблиц тестовых сред 1 или производственной среды 2

**{inn}** – ИНН организации, для которой пробивается чек

**{document\_id}** –идентификатор документа, который был указан при его создании

В данном запросе подпись **SHA256-RSA** не используется. Выбираем подраздел **Headers** и заполняем ключ **Host** со значением **<calculated when request is sent>** рис. 12.

рисунок 12



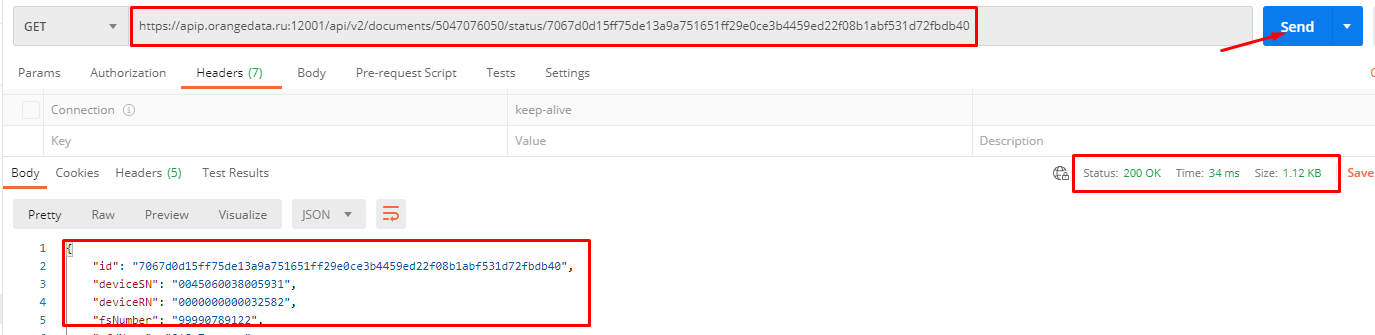
В данном примере для запроса GET использовали URL из тестовой среды 2 (с МГМ) из

таблицы 1

<https://apip.orangedata.ru:12001/api/v2/documents/5047076050/status/7067d0d15ff75de13a9a751651ff29e0ce3b4459ed22f08b1abf531d72fbdb40>

Заполнив все необходимые данные рис. 12, можем выполнить запрос, нажав кнопку **Send**. Видим, что запрос прошёл успешно: код 200, тело ответа, время ответа и размер запроса.

Рисунок 13



Ответ: api может возвращать следующие статус-коды:

* 202 Accepted – чек создан и добавлен в очередь на обработку, но еще не обработан, пустое тело ответа
* 400 Bad Request – организация не найдена, чек с указанным идентификатором не найден
* 401 Unauthorized – клиентский сертификат не прошел проверку
* 200 OK – чек обработан, пустое тело ответа
* 524, Document Expired Before Processing – серверу не удалось за отведенное время обработать документ, отправьте чек с новым идентификатором для повторной обработки, пустое тело ответа