



GENESIS BLOCK
IT challenge for your business

Описание платёжного протокола системы процессинга

БАЗОВЫЕ ЗАПРОСЫ



Версии документа

Версия	Дата	Внесённые изменения
1.0	05.2021	Первая версия документа
1.1	06.2021	Добавлена информация по запросу на получение статуса платежа. Добавлена таблица статусов платежей.
1.1.1	07.2021	Актуализирована информация по процедуре создания ключей.
1.1.2	02.08.2021	Добавлена информация по созданию валидной пары ключей
1.1.3	02.08.2021	Добавлен информация по полю SESSION в запросах на проверку реквизитов и на регистрацию транзакции, изменено описание поля FIELDS.
1.1.4	02.08.2021	Добавлена информация по созданию запроса для обмена ключами

Оглавление

1 Общие сведения	3
1.1 Формат сообщения	3
1.2 Общие параметры запросов	3
1.3 ЭЦП	3
1.4 Формат ответа	3
2 Базовые запросы	4
2.1 Предварительная проверка параметров транзакции	4
2.2 Регистрация транзакции	4
2.3 Получение информации	5
2.3 Получение внутрисистемного курса обмена для пары валют	6
2.4 Получение курса обмена валют для провайдера	6
2.5 Получение списка доступных для точки провайдеров.	7
2.6 Получение структуры меню терминала	7
2.7 Получение технической информации по конфигурации провайдера	8
2.8 Получение текущего статуса принятого платежа	9
Приложение 1 – возможные коды ошибок в ответах на запросы	10
Приложение 2 - механизм обмена открытыми ключами	11
Приложение 3 – возможные коды статусов платежей	13
Приложение 4 – создание пары ключей с использованием библиотеки OpenSSL	14
Windows	14
Установка OpenSSL	14
Создание пары ключей	14
Linux	14
Установка OpenSSL (Debian-based)	14
Установка OpenSSL (RedHat-based)	14
Создание пары ключей	14
Приложение 5 – формирование запроса для обмена открытыми ключами (Linux)	15

1 Общие сведения

Канал передачи данных – HTTP(S),

Тело сообщения – PlainText,

ЭЦП передаётся в заголовке x-sign, URL-кодированное.

Относительная часть URL платёжного сервиса: */pay*.

Базовую часть URL необходимо получить у курирующего менеджера.

1.1 Формат сообщения

Сообщение состоит из массива параметров, разделённых между собой символами перевода строки (OD0A).

Пример сообщения:

AP=1234

REQ_TYPE=1

OP_ID=1000611

FIELDS=field100=9161234567

1.2 Общие параметры запросов

- AP – номер точки в системе, выдаётся при создании платёжной точки;
- REQ_TYPE – код типа запроса;
- OP_ID – код провайдера в системе.

1.3 ЭЦП

Все сообщения подписываются ЭЦП. ЭЦП формируется по алгоритму SHA1-RSA, кодируется BASE64, URL-кодировается и помещается в заголовок x-sign.

Процедура создания пары ключей описана в Приложении 4.

Процедура обмена открытыми ключами описана в Приложении 2.

Процедура отправки открытого ключа точки из Linux описана в Приложении 5.

1.4 Формат ответа

Ответ также представляет собой массив параметров, разделённых между собой символами перевода строки (OD0A).

Пример ответа:

ERROR=0

RESULT=0

ANSWER_DATA={"exchangeRate":48.3652,"currencyFrom":"USD","currencyTo":"RUB"}

Ответ считается положительным, если оба поля ERROR и RESULT содержат 0.

Тело ответа подписывается ключами сервера. Подпись ответа формируется аналогично подписи запроса и помещается в заголовок x-sign.

2 Базовые запросы

2.1 Предварительная проверка параметров транзакции

Запрос обеспечивает предварительную (до регистрации транзакции) проверку введенных пользователем параметров транзакции.

Пример запроса:

```
AP=8017
REQ_TYPE=1
OP_ID=1000624
SESSION=1602211141499810216H
AMOUNT=180
AMOUNT_ALL=200
FIELDS=field100=1111222233334444&field101=9991234567
PROVIDER_CURRENCY=RUB
TERMINAL_CURRENCY=RUB
```

, где:

- SESSION - любая уникальная в рамках терминала строка длиной до 20 символов.
- AMOUNT – сумма для проверки возможности проведения транзакции, за вычетом доп. комиссии (необходима для некоторых провайдеров);
- AMOUNT_ALL – сумма для проверки возможности проведения транзакции, до вычета доп. комиссии (необходима для некоторых провайдеров);
- PROVIDER_CURRENCY – валюта провайдера
- TERMINAL_CURRENCY - валюта точки
- FIELDS – параметры транзакции в формате
field%id1%=%field_value1%&field%id2%=%field_value2%

Пример положительного ответа:

```
ERROR=0
RESULT=0
```

Пример отрицательного ответа:

```
ERROR=1
RESULT=23
```

Возможные коды ошибок даны в приложении 1.

2.2 Регистрация транзакции

Запрос обеспечивает регистрацию в системе новой транзакции после проверки валидности полученных данных.

Пример запроса:

```
AP=1234
REQ_TYPE=3
OP_ID=1000614
```

SESSION=1602151234560310145H
AMOUNT_ALL=1000
AMOUNT=950
AMOUNT_TO_PROVIDER=23.14
FIELDS= field100=1111222233334444&field101=9991234567
PROVIDER_CURRENCY=USD
TERMINAL_CURRENCY=RUB
INIT_DATETIME=16.02.2015 12:34:56
CHEQUE_NUM=42

, где:

- SESSION – любая уникальная в рамках терминала строка длиной до 20 символов, используется в качестве идентификатора транзакции.
- AMOUNT_ALL – сумма, внесённая пользователем, до вычета доп. комиссии;
- AMOUNT – сумма, внесённая пользователем, за вычетом доп. комиссии;
- PROVIDER_CURRENCY – валюта провайдера
- TERMINAL_CURRENCY - валюта точки
- FIELDS – параметры транзакции в формате field%id1%=%field_value1%& field%id2%=%field_value2%
- CHEQUE_NUM – номер напечатанного чека
- INIT_DATETIME – локальные время и дата приёма платежа.

Пример положительного ответа:

ERROR=0
RESULT=0

Пример отрицательного ответа:

ERROR=1
RESULT=23

Возможные коды ошибок даны в приложении 1.

В случае возникновения транспортной ошибки в процессе отправки запроса, запрос необходимо повторить с тем же параметром SESSION.

2.3 Получение информации

Запрос обеспечивает независимое от процесса проведения транзакции получение информации по данному провайдеру.

Пример запроса:

AP=8017
REQ_TYPE=5
OP_ID=1000624
FIELDS=field100=89161234567&field101=42

, где:

- FIELDS – параметры транзакции в формате field%id1%=%field_value1%& field%id2%=%field_value2%

Пример положительного ответа:

ERROR=0

RESULT=0

ANSWER_DATA=%data% - полученная от шлюза информация (формат информации специфичен для конкретного провайдера).

Пример отрицательного ответа:

ERROR=1

RESULT=23

Возможные коды ошибок даны в приложении 1.

2.3 Получение внутрисистемного курса обмена для пары валют

Запрос обеспечивает получение внутрисистемных курсов обмена валют.

Пример запроса:

AP=8017

REQ_TYPE=81

SELL_CURRENCY=USD

BUY_CURRENCY=RUB

, где:

- SELL_CURRENCY – валюта продажи;
- BUY_CURRENCY - валюта покупки;

Пример положительного ответа:

ERROR=0

RESULT=0

ANSWER_DATA={"exchangeRate":48.3652,"currencyFrom":"USD","currencyTo":"RUB"}

Пример отрицательного ответа:

ERROR=1

RESULT=23

Возможные коды ошибок даны в приложении 1.

2.4 Получение курса обмена валют для провайдера.

Запрос обеспечивает получение курса обмена валюты между валютой терминала и внутренней валютой провайдера с учётом курса валюты провайдера на шлюзе.

Пример запроса:

AP=8017

REQ_TYPE=80

TERMINAL_CURRENCY=USD

PROVIDER_CURRENCY=RUB

, где:

- `TERMINAL_CURRENCY` – валюта терминала;
- `PROVIDER_CURRENCY` - валюта провайдера на шлюзе;

Пример положительного ответа:

```
ERROR=0
RESULT=0
ANSWER_DATA={"exchangeRate":48.3652,"currencyFrom":"RUB","currencyTo":"RUB"}
```

Пример отрицательного ответа:

```
ERROR=1
RESULT=23
```

Возможные коды ошибок даны в приложении 1.

2.5 Получение списка доступных для точки провайдеров.

Запрос обеспечивает получение содержимого привязанного к терминалу профиля.

Пример запроса:

```
AP=8017
REQ_TYPE=12
```

Пример положительного ответа:

```
ERROR=0
RESULT=0
ANSWER_DATA=<?xml version="1.0" encoding="Windows-1251"?><providers><provider
id="1000552" name="Госуслуги (QR)"/><provider id="1000531" name="MTC"/><provider id="3009"
name="MTC"/>...</providers>
```

Пример отрицательного ответа:

```
ERROR=1
RESULT=30
```

Возможные коды ошибок даны в приложении 1.

2.6 Получение структуры меню терминала

Запрос обеспечивает получение структуры меню привязанного к терминалу профиля.

Пример запроса:

```
AP=8017
REQ_TYPE=10
```

Пример положительного ответа:

```
ERROR=0
RESULT=0
ANSWER_DATA=<?xml version="1.0" encoding="Windows-1251"?><root><main_menu><group
id="552" title="ЖКХ и Гос.Услуги ( QR)" image="btn_edelweiss_zhkh_and_govservices.png"
sh_n="1" enable="1" visible="1" image_bg="" type="" view_type=""><operator_id id="1000552"
name="Госуслуги (QR)" image=" GovServicesQR.png" image_bg=" GovServicesQR.png" sh_n="0"
```

```
enable="1" visible="1" /></group><group id="0" title="Оплата услуг" image="btn_main.png"
sh_n="1" enable="1" visible="1" image_bg="" type="" view_type="" /></main_menu><menu
title="Выберите вид платежа" logo="bg_banner.png" back="00.png"><group id="100" title=""
image="" sh_n="0" enable="1" visible="1" image_bg="" type="" view_type=""><operator_id
id="999" image="mob_mts_espp.png" image_bg="mob_mts_espp.png" sh_n="0" enable="1"
visible="1" /></group><group id="106" title="ДРУГИЕ УСЛУГИ" image="icon_other.png" sh_n="0"
enable="1" visible="1" image_bg="sv_bg_misc.png" type="" view_type=""><operator_id id="513"
name="Фаберлик" image="faberlic.png" image_bg="faberlic.png" sh_n="0" enable="1" visible="1"
/><group id="254" title="Игры и развлечения" image="games.png" sh_n="0" enable="1"
visible="1" image_bg="" type="" view_type=""><operator_id id="6097" name="Bloody world"
image="bloodyworld2.png" image_bg="bloodyworld2.png" sh_n="0" enable="1" visible="1"
/></group></group></menu></root>
```

, где:

- main_menu – главное меню профиля (обычно отображается на главной странице);
- menu – общее меню профиля;
- group - группа провайдеров;
- operator_id – информация о провайдере.

Пример отрицательного ответа:

```
ERROR=1
RESULT=30
```

Возможные коды ошибок даны в приложении 1.

2.7 Получение технической информации по конфигурации провайдера

Запрос обеспечивает получение информации по провайдеру.

Пример запроса:

```
AP=8017
REQ_TYPE=12
OP_ID=1000552
```

Пример положительного ответа:

```
ERROR=0
RESULT=0
ANSWER_DATA=<?xml version="1.0" encoding="Windows-1251"?><operator
id="3009"><name>MTC</name><image>mob_mts_espp.png</image><limit min="2"
max="15000"/><fields><field id="100" invoice_search="0" read_only="0" hidden="0" info_only="0"
allow_reedit="0" secrete="0" crypt="0" isName="0" required="1" amount_need="0"
type="masked"><name>Номер телефона</name><comment>Номер вводится без "[b]8[/b]".
Вместо префикса "[b]495[/b]" вводить "[b]985[/b]".</comment><mask>8 (**) ***-**-
**</mask></field></fields><processor id="412" offline="1"/></operator>
```

, где:

- operator – информация о запрошенном провайдере;
- limit – платёжные лимиты;

- **fields** – информация по заполняемым пользователями полям.

Пример отрицательного ответа:

ERROR=1
RESULT=30

Возможные коды ошибок даны в приложении 1.

2.8 Получение текущего статуса принятого платежа

Запрос обеспечивает получение содержимого привязанного к терминалу профиля.

Пример запроса:

AP=8017
REQ_TYPE=13
SESSION=12345678901234567890

Пример положительного ответа (в случае наличия соответствующего платежа):

ERROR=0
RESULT=0
ANSWER_DATA=<?xml version="1.0" encoding="Windows-1251"?><status>0</status >

Пример положительного ответа (в случае отсутствия соответствующего платежа):

ERROR=0
RESULT=0
ANSWER_DATA=<?xml version="1.0" encoding="Windows-1251"?><status>-1</status >

Пример отрицательного ответа:

ERROR=1
RESULT=30

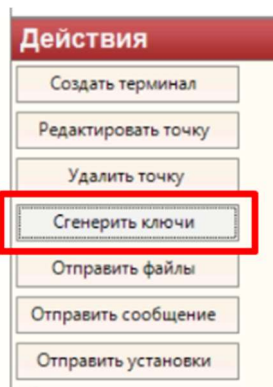
Возможные коды статусов даны в приложении 3.

Приложение 1 – возможные коды ошибок в ответах на запросы

Код	Описание
3	Терминал не найден на сервере
6	Ошибка электронной подписи
10	Неправильный формат сообщения
15	Неправильный номер провайдера
23	Неправильные поля ввода данных
24	Ошибка связи с провайдером
30	Ошибка сервера
134	Отсутствует открытый ключ терминала
1001	Неизвестный тип запроса
1002	Ошибка БД
1003	Пользователь не найден
1004	Неправильный пароль
1005	Пользователь заблокирован
1006	Провайдер заблокирован для данного субдилера
1007	Не установлен профиль для данного терминала
1008	Неизвестный запрос к шлюзу провайдера
1009	Закрыт счёт клиента

Приложение 2 - механизм обмена открытыми ключами

Осуществляет обмен открытыми ключами между сервером и терминалом. Предварительно процесс необходимо инициировать в клиенте мониторинга. Для этого необходимо выбрав нужную точку, нажать на кнопку 'Сгенерить ключи':



В открывшемся окне необходимо задать пароль операции:

Диалоговое окно с заголовком 'Сгенерить ключи'. В окне есть следующие поля для ввода: 'Код точки', 'Имя точки', 'Адрес точки', 'Агент' и 'Пароль техника' (с звездочкой и красной стрелкой). Внизу окна расположены кнопки 'Подтвердить' и 'Отменить'.

После этого можно отправлять нижеописанный запрос.

При необходимости осуществить замену ключей, процедура выполняется повторно, при этом сообщения, подписанные старыми ключами перестают приниматься системой.

Относительная часть URL сервиса по обмену публичными ключами: [/pay/genkey](#).

Базовую часть URL необходимо получить у курирующего менеджера.

Пример запроса:

```
<?xml version="1.0"?>
<root>
  <request type="genkey">
    <login>3920</login>
    <password>e10adc3949ba59abbe56e057f20f883e</password>
    <pubkey>-----BEGIN RSA PUBLIC KEY-----
MIGJAoGBAMjZJwa2QGr3hKhgAGn%2b/oOtEsICDSliTmhvaFFd9FenFhRHT9PW3I9U
BWXdPRPpWmeCOeR3ClwvCl11WOOJ1NGDk57gYV7zKlzfRTPOg3QPjBvHVqWee2R
woHKSubsi5JRmvflt%2bW8Te68ljQEWgu0OpPpfOn%2bmoAJyn33YKzZAgMBAAE=
-----END RSA PUBLIC KEY-----
    </pubkey>
  </request>
```

</root>

, где:

- login – код точки в системе;
- password – MD5-хэш заданного пароля;
- pubkey – публичный ключ точки, предварительно URL-закодированный.

Пример положительного ответа:

```
<?xml version="1.0"?>
<root><answer><result>0</result><server_public_key>-----BEGIN RSA PUBLIC KEY-----
MIGJAoGBAMjZJwa2QGr3hKhgAGn%2b/o0tEsICDSliTmhvaFFd9FenFHrHT9PW3I9U
BWXdapRPpWmeCOeR3ClwvCl11WOOJ1NGDk57gYV7zKlzfRTPOg3QPjBvHVqWee2R
woHKSubsi5JRmvft%2bW8Te68ljQEWgu0OpPpfOn%2bmoAJyn33YKzZAgMBAAE=
-----END RSA PUBLIC KEY-----
</server_public_key></answer></root>
```

Пример отрицательного ответа:

```
<?xml version="1.0"?>
<root><answer><result>1</result></answer></root>
```

, возможные коды ошибок описаны в следующей таблице:

Код	Описание
0	Запрос прошёл успешно, ключ точки зарегистрирован в системе.
1	Ошибка обработки запроса (возможно неправильный формат ключа).
2	Неправильный логин (код точки) или пароль. Возможно, процедура обмена ключами не была инициирована в мониторинге.

Приложение 3 – возможные коды статусов платежей

Код	Описание
-1	Платёж с таким идентификатором отсутствует в системе.
0	Платёж принят к проведению.
4	Данные платежа проверены.
5	Платёж принят к проведению внешним сервисом.
6	Платёж в проведении (этап ожидания проведения платежа внешним сервисом).
7	Платёж успешно проведён.
8	Платёж не проведён.
10	Платёж не проведён (превышен дневной лимит для точки).
106	Платёж перепроведён.
201	Платёж не проведён - недостаточная сумма платежа.
202	Платёж не проведён - отсутствует курс обмена валют.
203	Платёж не проведён - неправильная валюта платежа.

Приложение 4 – создание пары ключей с использованием библиотеки OpenSSL

Windows

Установка OpenSSL

Скачиваем и устанавливаем https://slproweb.com/download/Win64OpenSSL_Light1_1_1k.msi

Открываем командную строку (CMD) с правами администратора и выполняем:

```
SET OPENSSL_CONF=C:\OpenSSL-Win32\bin\openssl.cfg
```

Создание пары ключей

Создаем закрытый ключ

```
openssl genrsa -out privatekey.pem 2048
```

С помощью этой команды мы создали закрытый ключ privatekey.pem размером 2048 бит

Создаем публичный ключ

```
openssl rsa -in privatekey.pem -outform PEM -pubout -out publickey.pem
```

Пара ключей находится в той же директории в которой были запущены команды

Linux

Установка OpenSSL (Debian-based)

```
sudo apt update && sudo apt install openssl
```

Установка OpenSSL (RedHat-based)

```
sudo yum install openssl
```

Создание пары ключей

Команды для создания ключей такие же, как и для windows версии

```
openssl genrsa -out privatekey.pem 2048
```

```
openssl rsa -in privatekey.pem -outform PEM -pubout -out publickey.pem
```

Приложение 5 – формирование запроса для обмена открытыми ключами (Linux)

Устанавливаем необходимые зависимости:

```
apt update && apt install syslinux-utils curl sed gridsite-clients
```

В консоли стартуем следующую конструкцию:

```
LOGIN1="1234"
PASS1="123456"
cat <<EOF > xml.txt
<?xml version="1.0"?>
<root>
  <request type="genkey">
    <login>${LOGIN1}</login>
    <password>`echo -n $PASS1 | md5sum|sed 's/ //g;s/-$/g'`</password>
    <pubkey>-----BEGIN PUBLIC KEY-----
EOF
urlencode `sed '1d;$d' ./publickey.pem` >> xml.txt
cat <<EOF >> xml.txt
-----END PUBLIC KEY-----
  </pubkey>
</request>
</root>
EOF
```

,где:

- LOGIN1 - код точки в системе
 - PASS1 - пароль plaintext
- и отправляем наш запрос на сервер

```
curl -X POST --data-binary @xml.txt -H 'Accept: application/xml' -H 'Content-Type: application/xml'
https://domain/pay/genkey
```

при отсутствии ошибок мы получим ответ вида

```
<?xml version="1.0" encoding="Windows-1251"?>
```

```
<answer><result>0</result><server_public_key>-----BEGIN PUBLIC KEY-----
```

```
.....
```

Внимание!

Сохраните полученный публичный ключ сервера для дальнейшей валидации ответов системы.