

Оглавление

01 Формирование запроса на оплату.....	1
02 Получение информации о транзакции	9
03 Получение списка транзакций.....	11
04 Формирование запроса на отмену платежа.....	17
04.1 Формирование запроса на частичную отмену платежа	19
05 Создание счета на оплату	21
06 Получение информации о выставленном счете на оплату	26
07 Уведомления о транзакциях	28
08 Рекуррентные платежи	30
09 Холдирование.....	33
Создание запроса на платеж с предавторизацией.....	33
Подтверждение предавторизации	33
Отмена предавторизации	35
Примеры расчета сигнатуры на разных языках.....	35
PHP	35
NodeJS	36
Python	37
C#.....	39

API платежного шлюза (Версия 1.2)

Общий порядок взаимодействия

1. Партнер перенаправляет пользователя на платежную страницу с помощью POST-запроса, в котором передаются:
 0. данные о платеже
 1. ID партнера
 2. URL для возврата пользователя
 3. криптографическая подпись
2. Пользователь заполняет платежную страницу, проходит 3-DS аутентификацию (при необходимости), и завершает платеж
3. Пользователь перенаправляется по адресу, указанному в п.1
4. Интернет-эквайринг уведомляет партнера о совершении транзакции (по электронной почте, с помощью callback-запроса - в зависимости от настроек)

01 Формирование запроса на оплату

Данный метод предназначен для создания запроса на оплату. В случае корректного составления запроса откроется платежная форма для плательщика.

Формат запроса

POST - <https://pay.modulbank.ru/pay>

Для проведения платежа необходимо отправить POST-запросом форму со следующими полями:

Название поля	Описание	Формат
merchant	Идентификатор магазина, который выдается в личном кабинете на этапе интеграции.	Обязательный параметр. Строка (максимум 128 символов). Допускаются только печатные ASCII символы.
amount	Сумма платежа.	Обязательный параметр. Вещественное число, два знака после точки. Формат: «100» или «200.45».
order_id	Уникальный идентификатор заказа в интернет-магазине.	Обязательный параметр. Строка (максимум 50 символов).client_phone
custom_order_id	Идентификатор заказа, который будет отображаться покупателю	Необязательный параметр Строка (максимум 50 символов)
description	Описание платежа.	Обязательный параметр. Строка (максимум 250 символов).
success_url	Адрес страницы, на который будет отправлен плательщик в случае успешной операции. *Если хотите использовать страницу успешной оплаты Модульбанка, то укажите в этом поле: https://pay.modulbank.ru/success	Обязательный параметр. Строка (максимум 128 символов). Формат: url encode.

testing	Флаг тестового режима, в котором можно совершать произвольное количество транзакций с использованием тестовых карт. При включенной интеграции с кассой чеки не формируются.	<p>Необязательный параметр.</p> <p>Значение логического типа.</p> <p>1 – тестовый платеж; 0 – реальный платеж.</p> <p>По умолчанию реальные платежи.</p>
callback_url	Адрес, на который будет отправлено уведомление в случае успешной оплаты.	<p>Необязательный параметр.</p> <p>Строка (максимум 128 символов).</p> <p>Формат: url encode.</p>
callback_on_failure	Включение/отключение отправки уведомления, когда операция завершилась неуспешно.	<p>Необязательный параметр.</p> <p>Значение логического типа.</p> <p>1 – отправлять; 0 - не отправлять.</p> <p>По умолчанию не отправляется.</p>
client_phone	Номер телефона клиента.	<p>Необязательный параметр.</p> <p>Строка (максимум 15 символов).</p> <p>Формат: «+75559091555».</p>
client_name	Имя и фамилия клиента.	<p>Необязательный параметр.</p> <p>Строка (максимум 100 символов).</p>
client_email	E-mail клиента.	<p>Необязательный параметр.</p> <p>Строка (максимум 64 символа)</p>

client_id	Идентификатор клиента.	<p>Необязательный параметр.</p> <p>Строка (максимум 128 символов)</p> <p>Допускаются только печатные ASCII символы</p>
meta	Поле для дополнительных параметров в формате JSON	<p>Необязательный параметр</p> <p>JSON-строка</p>
receipt_contact	Email получателя чека.	<p>Необязательный параметр</p> <p>Если в ЛК включена удаленная регистрация чеков через онлайн-кассу, на этот адрес отправится чек.</p> <p>Строка (максимум 64 символа).</p>
receipt_items	Позиции чека.	<p>Обязательный параметр, если в ЛК включена удаленная регистрация чеков через онлайн-кассу.</p> <p>json-объект с позициями чека.</p> <p>Обязательные поля и пример json-объекта описаны в разделе «Отправка чеков».</p>
unix_timestamp	Текущее время.	<p>Обязательный параметр.</p> <p>Дата и время.</p> <p>Формат: UNIX Time.</p>
lifetime	Время жизни страницы в секундах	<p>Необязательный параметр.</p> <p>Целое число</p>

timeout_url	URL-адрес для редиректа плательщика по истечении времени, указанного в lifetime	Строка (максимум 128 символов). Формат: url encode. Необязательный параметр.
salt	Случайная величина.	Необязательный параметр. Строка (максимум 32 символа) Допускаются только печатные ASCII символы.
signature	Криптографическая подпись.	Обязательный параметр. Строка (40 символов в нижнем регистре). Алгоритм вычисления вычисления подписи описан в разделе «Алгоритм вычисления поля signature».
start_recurrent	Указывает, что карту, с которой проводится эта транзакция, нужно сохранить в карточном хранилище для последующих рекуррентных платежей по ней. Для последующих рекуррентных платежей по этой карте будет необходимо использовать ID этой транзакции. (см. 08 Рекуррентные платежи) Доступно только после одобрения заявки на проведение рекуррентных платежей в личном кабинете.	Необязательный параметр. Значение логического типа. 1 – рекуррентная транзакция; 0 или отсутствует - не рекуррентная транзакция. По умолчанию не отправляется.

preauth	Указывает, что должно быть проведено холдирование средств, по которому впоследствии будет произведено списание (см. 09 Холдирование)	<p>Необязательный параметр.</p> <p>Значение логического типа.</p> <p>1 – запрос с холдированием; 0 или отсутствует - одностадийный платёж.</p> <p>По умолчанию не отправляется.</p>
show_payment_methods	Указывает, какие из доступных платежных методов (карты, СБП и др.) показывать покупателю.	<p>Необязательный параметр.</p> <p>Строка.</p> <p>["sbp"] - только СБП ["card"] - только карты ["sbp", "card"] - оба метода.</p> <p>Доступные значения:</p> <p>card - карты sbp - СБП googlepay - кнопка GooglePay на платежной странице applepay - кнопка ApplePay на платежной странице</p> <p>По умолчанию выводятся все доступные методы.</p>
callback_with_receipt	Добавляет в коллбек об оплате информацию о сформированных чеках для онлайн-кассы.	<p>Необязательный параметр.</p> <p>Значение логического типа.</p> <p>1 – отправлять информацию; 0 или отсутствует - не отправлять.</p> <p>По умолчанию не отправляется.</p>

Обратите внимание, что все текстовые поля должны быть в кодировке UTF8.

Пример отправляемой формы:

```
<form method="post" action="https://pay.modulbank.ru/pay">

    <input type="hidden" name="testing" value="1">

    <input type="hidden" name="salt"
value="dPUTLtbMfcTGzkaBnGtseKlcQymCLrYI">

    <input type="hidden" name="order_id" value="14425840">

    <input type="hidden" name="amount" value="973">

    <input type="hidden" name="merchant" value="ad25ef06-1824-413f-8ef1-
c08115b9b979">

    <input type="hidden" name="signature"
value="9b28fa592922dc8a0c1ba2e40f2c0432aa617afd">

    <input type="hidden" name="description" value="Заказ №14425840">

    <input type="hidden" name="client_phone" value="+7 912 9876543">

    <input type="hidden" name="client_email" value="test@test.ru">

    <input type="hidden" name="success_url"
value="http://myawesomesite.com/payment_success">

    <input type="hidden" name="receipt_contact" value="test@mail.com">

    <input type="hidden" name="unix_timestamp" value="1573451160">

    <input type="hidden" name="receipt_items"
value="[{"discount_sum": 40, "name": "Товар",
"payment_method": "full_prepayment",
"payment_object": "commodity", "price": 48,
"quantity": 10, "sno": "osn", "vat":
```



```

    &quot;vat10&quot;}},

    {&quot;name&quot;: &quot;Товар 2&quot;, &quot;payment_method&quot;:
    &quot;full_prepayment&quot;,

    &quot;payment_object&quot;: &quot;commodity&quot;, &quot;price&quot;: 533,
    &quot;quantity&quot;: 1, &quot;sno&quot;: &quot;osn&quot;, &quot;vat&quot;:
    &quot;vat10&quot;}}]">

    <input type="submit" value="Отправить платеж">

</form>

```

Пример ответа:

МодульБанк

myawesomesite

Сумма 973.00 RUB
Номер заказа 14425640

Номер карты

Срок действия 00 / 00

CVC

Для оплаты тестовыми платёжами используйте тестовые карты

VISA Mastercard MIR

ОПЛАТИТЬ

VeriSign VISA Mastercard SecureCode

Ваша платежная информация передается исключительно в защищенной системе. Интернет-платежи защищены 128-битным SSL-шифрованием и протоколом PCI DSS. Конфиденциальность платежных данных обеспечивает АС МБ. МодульБанк.

В ответ пользователь перенаправляется на страницу платежа.

После проведения оплаты произойдет перенаправление на страницу, которая была указана в поле success_url. В параметры GET будет добавлен id транзакции.

Пример:

https://pay.modulbank.ru/success?transaction_id=yiSEn5H905COse1biIvoCY

02 Получение информации о транзакции

Данный метод предназначен для получения информации о транзакции по платежку.

Формат запроса

GET - <https://pay.modulbank.ru/api/v1/transaction/>

Параметры запроса

Название поля	Описание	Формат
transaction_id	Идентификатор транзакции.	Обязательный параметр. Строка (максимум 50 символов).
merchant	Идентификатор магазина, который выдается в личном кабинете на этапе интеграции.	Обязательный параметр. Строка (максимум 128 символов). Допускаются только печатные ASCII символы.
unix_timestamp	Текущее время.	Обязательный параметр. Дата и время. Формат: UNIX Time.
salt	Случайная величина.	Необязательный параметр. Строка (максимум 32 символа) Допускаются только печатные ASCII символы.
signature	Криптографическая подпись.	Обязательный параметр. Строка (40 символов в нижнем регистре). Алгоритм вычисления вычисления подписи описан в разделе «Алгоритм вычисления поля signature».

Пример запроса:

```
https://pay.modulbank.ru/api/v1/transaction/?transaction\_id=qo9Kjd1vW68Pn1h9g2173e&merchant=ad25ef06-1824-413f-8ef1-c08115b9b979&unix\_timestamp=1542080393&signature=b47025989516768fc1fc60a5b38abd5cc3a8fbf&salt=GfudKOAsXobWVpNovJHCreKmJXNkLqtf
```

Пример ответа:

```
{ "status": "ok",  
  "transaction": {  
    "amount": "1000.00",
```

```

        "auth_code": "201471",
        "auth_number": "2164586974",
        "client_email": "user@example.ru",
        "client_phone": "",
        "completed_datetime": "2019-11-05 11:20:59",
        "created_datetime": "2019-11-05 11:20:51",
        "currency": "RUB",
        "custom_order_id": "",
        "description": "Оплата заказа №7",
        "message": "",
        "meta": "",
        "mps_error_code": null,
        "order_id": "83e78010-163d-49a0-87e0-276d2f9e23bb",
        "original_amount": "1000.00",
        "pan_mask": "558462*****6488",
        "payment_method": "card",
        "refunds": [],
        "rrn": "002164586974",
        "state": "COMPLETE",
        "testing": 0,
        "transaction_id": "saJ0G2qS0A3tLI5d1CHWr4",
        "updated_datetime": "2019-11-05 11:20:59"
    }
}

```

Возможные значения поля state:

<p>PROCESSING - В процессе WAITING_FOR_3DS - Ожидает 3DS FAILED - Ошибка COMPLETE - Готово</p>

03 Получение списка транзакций

Данный метод предназначен для получения списка транзакций по платежам.

Формат запроса

GET - <https://pay.modulbank.ru/api/v1/transactions/>

Параметры запроса

Название поля	Описание	Формат
merchant	Идентификатор магазина, который выдается в личном кабинете на этапе интеграции.	Обязательный параметр. Строка (максимум 128 символов). Допускаются только печатные ASCII символы.
unix_timestamp	Текущее время.	Обязательный параметр. Дата и время. Формат: UNIX Time.
timestamp_from	Нижняя граница временного диапазона	Обязательный параметр. Значение в формате ISO8601:2004, с обязательным указанием часового пояса. В качестве разделителя между датой и временем используется символ «Т». Пример: «2018-01-31T00:00:00.123+04:00»
timestamp_to	Верхняя граница временного диапазона	Обязательный параметр. Значение в формате ISO8601:2004, с обязательным указанием часового пояса. В качестве разделителя между датой и временем используется символ «Т». Пример: «2018-01-31T00:00:00.123+04:00»
state	Фильтр по статусу транзакции. По умолчанию возвращаются все транзакции.	Необязательный параметр. Возможные значения: <ul style="list-style-type: none">• PROCESSING - В процессе;• WAITING_FOR_3DS - Ожидает 3DS;• COMPLETE - Транзакция завершена успешно;• FAILED - Транзакция завершена с ошибкой.

testing	Возвращать тестовые или боевые транзакции.	1 – возвращать только тестовые транзакции; 0 – возвращать только реальные транзакции. Для возвращения тестовых транзакций в подсчете сигнатуры используйте тестовый секретный ключ. Если значение параметра равняется 0 или отсутствует - используйте боевой секретный ключ.
salt	Случайная величина.	Необязательный параметр. Строка (максимум 32 символа) Допускаются только печатные ASCII символы.
signature	Криптографическая подпись.	Обязательный параметр. Строка (40 символов в нижнем регистре). Алгоритм вычисления вычисления подписи описан в разделе «Алгоритм вычисления поля signature». Пример: 78d9e7d4ec844885790f185ce0e7a60374d01cbc

Пример запроса:

https://pay.modulbank.ru/api/v1/transactions/?unix_timestamp=1542795615&testing=1×tamp_to=2019-01-31T00:00:00.123+04:00×tamp_from=2018-01-31T00:00:00.123+04:00&salt=GfudKOAsXobWVpNovJHCReKmJXNkLqtf&signature=31ba86af6bce6423d28b79f0ac53ea1fd7acbeca&state=COMPLETE&merchant=25b649d7-03f2-4c14-a847-83205a4d913b

Пример ответа:

```
{
  "transactions": [
    {
      "created_datetime": "2018-11-15 06:06:13",
      "transaction_id": "sRkiFz9peAQ7lAFNj7DbYi",
      "payment_method": "card",
      "meta": "{\\\"example-property\\\": 1234}",
      "testing": 1,
      "message": "",
      "order_id": "569770",
      "currency": "RUB",
      "response_action": null,
      "mps_error_code": null,
      "original_amount": "13.00",
      "pan_mask": "404730*****7550",
      "state": "COMPLETE",
      "amount": "13.00"
    },
    {
      "created_datetime": "2018-11-15 07:20:03",
      "transaction_id": "0kNPLALWDfSEAN4BoUZpfK",
      "payment_method": "card",
```

```
    "meta": "{\\"example-  
property\\": 1234}",  
  
    "testing": 1,  
  
    "message": "",  
  
    "order_id": "965949",  
  
    "currency": "RUB",  
  
    "response_action": null,  
  
    "mps_error_code": null,  
  
    "original_amount": "11.00",  
  
    "pan_mask":  
"404730*****7550",  
  
    "state": "COMPLETE",  
  
    "amount": "11.00"  
  
  },  
  
  {  
  
    "created_datetime": "2018-  
11-15 07:24:46",  
  
    "transaction_id":  
"5dy2ecgVzsOriFuZL16m50",  
  
    "payment_method": "card",  
  
    "meta": "{\\"example-  
property\\": 1234}",  
  
    "testing": 1,  
  
    "message": "",  
  
    "order_id": "201338",  
  
    "currency": "RUB",  
  
    "response_action": null,  
  
    "mps_error_code": null,
```

```
        "original_amount": "49.00",

        "pan_mask":
"404730*****7550",

        "state": "COMPLETE",

        "amount": "49.00"

    },

    {

        "created_datetime": "2018-
11-19 08:34:41",

        "transaction_id":
"Dd1iZh1UbmKyFRLp5Yz9Nt",

        "payment_method": "card",

        "meta": "{\\"example-
property\\": 1234}",

        "testing": 1,

        "message": "",

        "order_id": "15529392",

        "currency": "RUB",

        "response_action": null,

        "mps_error_code": null,

        "original_amount":
"969.00",

        "pan_mask":
"404730*****7550",

        "state": "COMPLETE",

        "amount": "969.00"

    },

],
```



```
"status": "ok"

}
```

04 Формирование запроса на отмену платежа

Данный метод предназначен для отмены платежа и возврата средств плательщику. Отмену платежа можно выполнить через личный кабинет либо через вызов метода API.

Формат запроса

POST - <https://pay.modulbank.ru/api/v1/refund>

Параметры запроса

Название поля	Описание	Формат
merchant	Идентификатор магазина, который выдается в личном кабинете на этапе интеграции.	Обязательный параметр. Строка (максимум 128 символов) Допускаются только печатные ASCII символы
amount	Сумма платежа.	Обязательный параметр. Вещественное число, два знака после точки. Формат: «100» или «200.45».
transaction	Идентификатор транзакции, по которой необходимо осуществить возврат.	Обязательный параметр. Строка (максимум 50 символов).
unix_timestamp	Текущее время.	Обязательный параметр. Дата и время. Формат: UNIX Time.
salt	Случайная величина.	Необязательный параметр. Строка (максимум 32 символа) Допускаются только печатные ASCII символы.

signature	Криптографическая подпись.	<p>Обязательный параметр.</p> <p>Строка (40 символов в нижнем регистре).</p> <p>Алгоритм вычисления вычисления подписи описан в разделе «Алгоритм вычисления поля signature».</p>
-----------	----------------------------	---

Пример запроса:

```
"merchant": "46197191-2311-40b5-a8d3-6659602fd332",
"amount": "100",
"transaction": "nRo9bxb7hLOqVwtWMbHp5Y",
"unix_timestamp": "1538039634",
"signature": "995da849833a82e5706660e17cfc48d109eb5aa7"
```

Пример ответа:

```
{
  "message": "Успешно",
  "refund": {
    "amount": "100.00",
    "created_datetime": "2017-01-01 00:01:39.488918+00:00",
    "id": ""nRo9bxb7hLOqVwtWMbHp5Y"",
    "message": " ",
    "state": "COMPLETE"
  },
  "status": "ok",
  "transaction": {...} // информация о транзакции
}
```

Возможные статусы отмены платежа:

Статус	Описание
--------	----------

PENDING	Запрос на отмену ожидает исполнения.
PROCESSING	Запрос на отмену выполняется в данный момент.
WAITING_FOR_RESULT	Запрос на отмену выполнен, ожидается результат.
COMPLETE	Отмена была выполнена успешно.
FAILED	Произошла ошибка.

04.1 Формирование запроса на частичную отмену платежа

Важно!

Проведение частичного возврата доступно только на следующий день после операции оплаты, иначе в ответе message будет содержать "Операция не найдена". При повторном проведении частичного возврата по такой операции на следующий день ошибка автоматически устранится.

Если у Вашего магазина нет интеграции с онлайн-кассой, то для частичной отмены/возврата достаточно в запросе указать в поле amount возвращаемую сумму.

Если касса подключена, то нужно указать, какие именно позиции и в каком количестве Вы возвращаете. Это нужно для того, чтобы Интернет-Эквайринг корректно сформировал чек возврата.

Внимание!

Для работы с чеками частичного возврата Вам нужно получать информацию о чеке в коллбеке об оплате. Для этого нужно отправить в запросе на оплату 1 в поле callback_with_receipt или обратиться в чат поддержки, чтобы получение информации о чеке в коллбеке включили для Вашего магазина на постоянной основе.

В таком случае в основной запрос добавляется поле refund_items, содержащее json с возвращаемыми позициями. Формат аналогичен полю receipt_items из запроса на формирование оплаты.

Его содержание:

Имя	описание	обязательность
initial_item	ID айтема, по которому будет возврат. Возвращается в коллбеке об оплате.	+
amount	сумма возврата	-
quantity	количество возврата	+

Пример:

была продажа с чеком

```
[
{
"name": "Консультация",
"sno": "osn",
"payment_object": "service",
"payment_method": "full_prepayment",
"price": "1000",
"vat": "none",
"quantity": 1
},
{
"name": "Консультация доп.",
"sno": "osn",
"payment_object": "service",
"payment_method": "full_prepayment",
"price": "200",
"vat": "none",
"quantity": 5
}
]
```

Если нужно вернуть только 1 консультацию по 1000р (или все пять по 200р) - отправляем refund_item вида

```
{
"quantity": 1,
"initial_item": "01934a95-4edf-47aa-b177-4e7858b6b983"
}
```

или

```
{
"quantity": 5,
"initial_item": "a07534aff-43da-445a-b177-b4c395acb111"
}
```

Если нужно вернуть произвольную сумму, например 600р из 1000р по позиции Консультация (или 300р из 400р за две из пяти шт. по позиции Консультация доп.) - отправляем так:

```
{  
  "quantity": 1,  
  "initial_item": "01934a95-4edf-47aa-b177-4e7858b6b983",  
  "amount": 600  
}
```

В таком случае сначала сформируется чек полного возврата позиции Консультация (на 1000р), а затем чек продажи позиции Консультация по новой цене (400р.). Покупателю на карту вернется 600р.

Во втором примере:

```
{  
  "quantity": 2,  
  "initial_item": "a07534aff-43da-445a-b177-b4c395acb111",  
  "amount": 300  
}
```

В таком случае сначала сформируется чек полного возврата двух шт. по позиции Консультация доп. (400р), а затем чек продажи позиции Консультация доп. по новой цене (на 100р, количество - 2, цена - 50р). Покупателю на карту вернется 300р.

05 Создание счета на оплату

Данный метод предназначен для создания счета на оплату покупателю. В случае корректного составления запроса создается ссылка на оплату, которую необходимо отправить покупателю.

Формат запроса

POST - <https://pay.modulbank.ru/api/v1/bill/>

Параметры запроса

Название поля	Описание	Формат
merchant	Идентификатор магазина, который выдается в личном кабинете на этапе интеграции.	Обязательный параметр. Строка (максимум 128 символов) Допускаются только печатные ASCII символы

amount	Сумма платежа.	Обязательный параметр. Вещественное число, два знака после точки. Формат: «100» или «200.45».
description	Описание платежа.	Обязательный параметр. Строка (максимум 250 символов)
testing	Флаг тестового режима, в котором можно совершать произвольное количество транзакций с использованием тестовых карт. При включенной интеграции с кассой чеки не формируются.	Необязательный параметр. Значение логического типа. 1 – тестовый платеж; 0 – реальный платеж. По умолчанию реальные платежи.
client_email	Е-mail клиента. Флаг отправки письма с уведомлением о выставленном счете	Необязательный параметр. Строка (максимум 64 символа) Если указано это поле и send_letter, клиенту будет отправлено письмо с уведомлением о выставлении счета.
client_name	Имя и фамилия клиента.	Необязательный параметр. Строка (максимум 100 символов).
custom_order_id	Идентификатор заказа, который будет отображаться покупателю	Необязательный параметр Строка (максимум 50 символов)
lifetime	Срок актуальности счета в секундах. По истечению счет будет недоступен к оплате.	Необязательный параметр. Целое число По умолчанию срок актуальности равен одной неделе
send_letter	Флаг отправки письма с уведомлением о выставленном счете.	Необязательный параметр. 1 – отправить письмо со ссылкой на оплату; 0 – не отправлять письмо. Если указать этот флаг (и client_email), то клиенту будет отправлено письмо, в котором будет ссылка на оплату.

receipt_contact	Email получателя чека.	Обязательный параметр (если в ЛК включена удаленная регистрация чеков через онлайн-кассу). Строка (максимум 64 символа).
receipt_items	Позиции чека	Обязательный параметр (если в ЛК включена удаленная регистрация чеков через онлайн-кассу). json-объект с позициями чека. Обязательные поля и пример json-объекта описаны в разделе «Отправка чеков».
unix_timestamp	Текущее время	Обязательный параметр. Дата и время Формат: UNIX Time
salt	Случайная величина.	Необязательный параметр. Строка (максимум 32 символа) Допускаются только печатные ASCII символы
signature	Криптографическая подпись.	Обязательный параметр. Строка (40 символов в нижнем регистре). Алгоритм вычисления вычисления подписи описан в разделе «Алгоритм вычисления поля signature». Пример: 78d9e7d4ec844885790f185ce0e7a60374d01cbc

reusable	<p>Одноразовый/многократный счет.</p> <p>Многократный счет предоставляет ссылку, по которой оплата может быть проведена неограниченное количество раз любыми покупателями. Для многократного счета не указываются поля send_letter, receipt_contact, client_email и lifetime, так как конечного получателя для счета нет. Email указывает сам покупатель на платежной странице.</p>	<p>Необязательный параметр.</p> <p>Значение логического типа.</p> <p>1 – многократный; 0 - одноразовый.</p> <p>По умолчанию не отправляется.</p>
callback_url	<p>Адрес, на который будет отправлено уведомление в случае успешной оплаты.</p>	<p>Необязательный параметр.</p> <p>Строка (максимум 128 символов).</p> <p>Формат: url encode.</p>
callback_on_failure	<p>Включение/отключение отправки уведомления, когда операция завершилась неуспешно.</p>	<p>Необязательный параметр.</p> <p>Значение логического типа.</p> <p>1 – отправлять; 0 - не отправлять.</p> <p>По умолчанию не отправляется.</p>

start_recurrent	<p>Указывает, что карту, с которой проводится транзакция, нужно сохранить в карточном хранилище для последующих рекуррентных платежей по ней. Для последующих рекуррентных платежей по этой карте будет необходимо использовать ID этой транзакции. (см. 08 Рекуррентные платежи)</p> <p>Доступно только после одобрения заявки на проведение рекуррентных платежей в личном кабинете.</p>	<p>Необязательный параметр.</p> <p>Значение логического типа.</p> <p>1 – рекуррентная транзакция; 0 или отсутствует - не рекуррентная транзакция.</p> <p>По умолчанию не отправляется.</p>
show_payment_methods	<p>Указывает, какие из доступных платежных методов (карты, СБП) показывать покупателю.</p>	<p>Необязательный параметр.</p> <p>Строка.</p> <p>["sbp"] - только СБП ["card"] - только карты ["sbp", "card"] - оба метода.</p> <p>По умолчанию выводятся все доступные методы.</p>

Пример запроса:

```
"signature": "2aad6f9bb04458b39b77811a7efcfbc10f0b09d8",
"amount": 83,
"send_letter": "1",
"unix_timestamp": 1542081557,
"testing": "1",
"description": "Оплата товара в магазине myawesomesite.com",
"salt": "GfudKOAsXobWVpNovJHCreKmJXNkLqtf",
"client_email": "test@test.ru",
"merchant": "ad25ef06-1824-413f-8ef1-c08115b9b979"
```

Пример ответа:

```
{ "bill":  
  { "amount": "83",  
    "client_email": "test@test.ru",  
    "created_datetime": "2018-11-13 05:19:49",  
    "currency": "RUB",  
    "description": "Оплата товара в магазине myawesomesite.com",  
    "expired": 0,  
    "id": "GGgEgDY3fvrDpVb25HMg2o",  
    "merchant": "ad25ef06-1824-413f-8ef1-c08115b9b979",  
    "number": 138,  
    "paid": 0,  
    "send_letter": 1,  
    "testing": 1,  
    "transaction": {},  
    "url": "https://pay.modulbank.ru/bill-GGgEgDY3fvrDpVb25HMg2o" },  
  "status": "ok" }
```

(i) Перейдя по ссылке, пришедшей в поле “url”, плательщик попадает на страницу оплаты.

Возможные статусы платежа на оплату:

Название поля	Описание
expired	Платеж просрочен. Значения: 0 - действующий 1 - просрочен
paid	Платеж оплачен. Значения: 0 - не оплачен 1 - оплачен

06 Получение информации о выставленном счете на оплату

Данный метод предназначен для получения информации о выставленном счете на оплату.

Формат запроса

GET - <https://pay.modulbank.ru/api/v1/bill/>

Параметры запроса

Название поля	Описание	Формат
id	Идентификатор счета.	Обязательный параметр. Строка (максимум 50 символов).
merchant	Идентификатор магазина, который выдается в личном кабинете на этапе интеграции.	Обязательный параметр. Строка (максимум 128 символов). Допускаются только печатные ASCII символы.
unix_timestamp	Текущее время.	Обязательный параметр. Дата и время. Формат: UNIX Time.
salt	Случайная величина.	Необязательный параметр. Строка (максимум 32 символа) Допускаются только печатные ASCII символы.
signature	Криптографическая подпись.	Обязательный параметр. Строка (32 символа в нижнем регистре). Алгоритм вычисления подписи описан в разделе «Алгоритм вычисления поля signature».

Пример запроса:

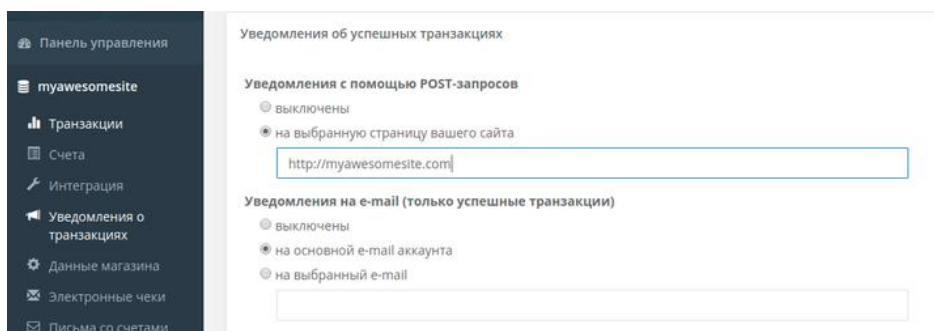
https://pay.modulbank.ru/api/v1/bill/?id=GGgEgDY3fvrDpVb25HMg2o&merchant=ad25ef06-1824-413f-8ef1-c08115b9b979&unix_timestamp=1542080393&signature=7e0829ce9a2aee5a27f9837580f211419c97f2b9&salt=GfudKOAsXobWVpNovJHCreKmJXNkLqtf

Пример ответа:

```
{ "bill":
  { "amount": "83",
    "client_email": "test@test.ru",
    "created_datetime": "2018-11-13 05:19:49",
    "currency": "RUB",
    "description": "Оплата товара в магазине myawesomesite.com",
    "expired": 0,
    "id": "GGgEgDY3fvrDpVb25HMg2o",
    "merchant": "ad25ef06-1824-413f-8ef1-c08115b9b979",
    "number": 138,
    "paid": 0,
    "send_letter": 1,
    "testing": 1,
    "transaction": {},
    "url": "https://pay.modulbank.ru/bill-GGgEgDY3fvrDpVb25HMg2o" },
  "status": "ok" }
```

07 Уведомления о транзакциях

После проведения оплаты сервер отправляет POST-запрос с уведомлением на url, указанный для оповещения. Вы можете указать его в запросе на оплату в поле `callback_url` либо в личном кабинете.



The screenshot shows a web interface for managing a site. On the left is a dark sidebar with a menu containing: 'Панель управления', 'myawesomesite', 'Транзакции', 'Счета', 'Интеграция', 'Уведомления о транзакциях' (highlighted), 'Данные магазина', 'Электронные чеки', and 'Письма со счетами'. The main content area is titled 'Уведомления об успешных транзакциях' and contains two sections. The first section, 'Уведомления с помощью POST-запросов', has two radio buttons: 'выключены' and 'на выбранную страницу вашего сайта' (which is selected). Below this is a text input field containing 'http://myawesomesite.com/'. The second section, 'Уведомления на e-mail (только успешные транзакции)', also has two radio buttons: 'выключены' and 'на основной e-mail аккаунта' (which is selected). Below this is an empty text input field.

В случае, если url указан и в запросе на оплату, и в ЛК, приоритетным считается указанный в запросе.

Ответ от сервера приходит в виде POST-запроса, содержащего данные о транзакции. Signature в данном случае считается из всех полей, содержащихся в ответе.

Сервер ожидает, что callback будет обработан на стороне магазина с ответом со статусом 200 OK. Иначе будут совершены попытки переотправки в количестве 14 штук, с интервалом времени, возрастающим по экспоненте.

Вид ответа:

```
'testing': '0',
'pan_mask': '220011*****4440',
'unix_timestamp': '1570161434',
'salt': 'DB9481A6554924BFD2F2279B5AD05B9D',
'rrn': '927703219385',
'transaction_id': '0EyuFLLZ9DagCXY8O67Q6x',
'original_amount': '10.00',
'auth_number': '2164219385',
'amount': '10.00',
'created_datetime': '2019-10-04 03:56:09',
'auth_code': '201471',
'signature': '622e1486dba17d05d080c6734131205a75d59188',
'client_phone': '+79999999999',
'client_email': 'example@example.ru',
'state': 'COMPLETE',
'order_id': '697144',
'currency': 'RUB',
'merchant': '51cb8a0f-6fb8-4a20-98b1-9fd85dc47500',
'payment_method': 'card',
'meta': '{"bill_id": "v1ICmFjY7nST9KARa5RsSJ"}' - в случае если был
выставлен счет
```

В случае, если транзакция завершилась неуспешно, и в запросе был указан параметр `callback_on_failure`, ответ будет иметь следующий вид:

```
'testing': '0',
'pan_mask': '220011*****4440',
'unix_timestamp': '1570161434',
'salt': 'DB9481A6554924BFD2F2279B5AD05B9D',
'transaction_id': '0EyuFLLZ9DagCXY8O67Q6x',
'original_amount': '10.00',
'amount': '10.00',
'created_datetime': '2019-10-04 03:56:09',
'signature': '161787e7ddb31ffa6ac1988ceff16ceffa69b3324',
'client_phone': '+79999999999',
'client_email': 'example@example.ru',
'state': 'FAILED',
'mps_error_code': '101',
'message': 'Отказ при выполнении авторизации',
'order_id': '697144',
'currency': 'RUB',
'merchant': '51cb8a0f-6fb8-4a20-98b1-9fd85dc47500',
'payment_method': 'card',
'meta': '{"bill_id": "v1ICmFjY7nST9KARa5RsSJ"}' - в случае если был
выставлен счет
```

Для проверки корректности уведомления на стороне интернет-магазина нужно провести расчет сигнатуры по пришедшим в ответе полям и убедиться, что она сходится с той, что содержится в пришедшем json от эквайринга.

Обратите внимание, что сигнатура для тестовых и реальных платежей считается по тестовому и реальному секретным ключам соответственно. Подробнее о расчете сигнатуры можно прочитать в разделе Алгоритм расчета поля signature.

08 Рекуррентные платежи

Рекуррентные платежи по умолчанию запрещены. Для включения обратитесь к поддержке в чат банка.

Шаги для привязки карты и совершения по ней рекуррентных платежей:

1. Провести первый рекуррентный платеж, указав в запросе на оплату параметр `start_recurrent`: 1. Платеж проводится как обычный, с вводом карты покупателя и проведением 3DS аутентификации. Покупатель должен быть уведомлен о том, что при проведении этого платежа его карта сохраняется, и по ней будут впоследствии происходить списания.
2. При успешном завершении первого рекуррентного платежа карта покупателя сохраняется в карточном хранилище Модульбанка.
3. Для проведения последующих транзакций по этой карте без участия кардхолдера нужно формировать запрос на платеж с указанием `transaction_id` первой транзакции. По этой транзакции будет получена карта из хранилища, проведен платеж и вернется ответ со статусом транзакции. Если в личном кабинете или запросе на платеж будет указан адрес для колбека, он также будет прислан.

Формат запроса

POST - https://pay.modulbank.ru/api/v1/next_recurrent/

Для проведения платежа необходимо отправить POST-запросом форму со следующими полями:

Название поля	Описание	Формат
merchant	Идентификатор магазина, который выдается в личном кабинете на этапе интеграции.	Обязательный параметр. Строка (максимум 128 символов). Допускаются только печатные ASCII символы.
first_recurrent_transaction	Идентификатор первой рекуррентной транзакции по этой карте.	Обязательный параметр. Строка.
amount	Сумма платежа.	Обязательный параметр. Вещественное число, два знака после точки. Формат: «100» или «200.45».
order_id	Уникальный идентификатор заказа в интернет-магазине.	Обязательный параметр. Строка (максимум 50 символов).
custom_order_id	Идентификатор заказа, который будет отображаться покупателю	Необязательный параметр Строка (максимум 50 символов)
description	Описание платежа.	Обязательный параметр. Строка (максимум 250 символов).

testing	Флаг тестового режима, в котором можно совершать произвольное количество транзакций с использованием тестовых карт.	<p>Необязательный параметр.</p> <p>Значение логического типа.</p> <p>1 – тестовый платеж; 0 – реальный платеж.</p> <p>По умолчанию реальные платежи.</p>
callback_url	Адрес, на который будет отправлено уведомление POST-запросом в случае успешной оплаты.	<p>Необязательный параметр.</p> <p>Строка (максимум 128 символов).</p> <p>Формат: url encode.</p>
client_phone	Номер телефона клиента.	<p>Необязательный параметр.</p> <p>Строка (максимум 64 символа).</p> <p>Формат: «+75559091555».</p>
client_name	Имя и фамилия клиента.	<p>Необязательный параметр.</p> <p>Строка (максимум 128 символов).</p>
client_email	E-mail клиента.	<p>Необязательный параметр.</p> <p>Строка (максимум 64 символа)</p>
client_id	Идентификатор клиента.	<p>Необязательный параметр.</p> <p>Строка (максимум 128 символов)</p> <p>Допускаются только печатные ASCII символы</p>
meta	Поле для дополнительных параметров в формате JSON	<p>Необязательный параметр</p> <p>JSON-строка</p>
receipt_contact	Email получателя чека.	<p>Обязательный параметр, если в ЛК включена удаленная регистрация чеков через онлайн-кассу.</p> <p>Если в ЛК включена удаленная регистрация чеков через онлайн-кассу, на этот адрес отправится чек.</p> <p>Строка (максимум 64 символа).</p>

receipt_items	Позиции чека.	Обязательный параметр, если в ЛК включена удаленная регистрация чеков через онлайн-кассу. json-объект с позициями чека. Обязательные поля и пример json-объекта описаны в разделе «Отправка чеков» .
unix_timestamp	Текущее время.	Обязательный параметр. Дата и время. Формат: UNIX Time.
salt	Случайная величина.	Необязательный параметр. Строка (максимум 32 символа) Допускаются только печатные ASCII символы.
signature	Криптографическая подпись.	Обязательный параметр. Строка (40 символов в нижнем регистре). Алгоритм вычисления вычисления подписи описан в разделе «Алгоритм вычисления поля signature» .

09 Холдирование

Создание запроса на платеж с предавторизацией

В запрос на платеж, описанный в 01 Формирование запроса на оплату, добавляется необязательный параметр preauth, принимающий 0 или 1 (0 или отсутствие - обычный платеж, 1 - платеж с предавторизацией)

При включенной онлайн-кассе в случае платежа с предавторизацией чек уходит в момент подтверждения (и на сумму подтверждения), поэтому на данном этапе поля receipt_contact и receipt_items могут быть пропущены.

Подтверждение предавторизации

Формат запроса

POST - <https://pay.modulbank.ru/api/v1/capture/>

Поле	Описание	Формат
transaction	Идентификатор транзакции с предавторизацией	Обязательный параметр. Строка (максимум 50 символов).
amount	Сумма списания. Должна быть меньше или равна сумме предавторизации.	Обязательный параметр. Вещественное число, два знака после точки. Формат: «100» или «200.45».
merchant	Идентификатор мерчанта	Обязательный параметр. Строка (максимум 128 символов) Допускаются только печатные ASCII символы
salt	Случайная величина	Необязательный параметр. Строка (максимум 32 символа) Допускаются только печатные ASCII символы.
unix_timestamp	Текущее время	Обязательный параметр. Дата и время. Формат: UNIX Time.
receipt_contact	Email получателя чека	Обязательный параметр (если в ЛК включена удаленная регистрация чеков через онлайн-кассу). Строка (максимум 64 символа).
receipt_items	Позиции чека	Обязательный параметр (если в ЛК включена удаленная регистрация чеков через онлайн-кассу). json-объект с позициями чека. Обязательные поля и пример json-объекта описаны в разделе «Отправка чеков».

signature	Криптографическая подпись	<p>Обязательный параметр.</p> <p>Строка (32 символа в нижнем регистре).</p> <p>Алгоритм вычисления вычисления подписи описан в разделе «Алгоритм вычисления поля signature».</p> <p>Пример:</p> <p>78d9e7d4ec844885790f185ce0e7a60374d01cbc</p>
-----------	---------------------------	---

В случае, если сумма списания будет меньше суммы предавторизации, плательщику будет оформлен возврат разницы в виде отмены платежа.

Callback о подтверждении будет отправлен на тот же адрес, что указан в запросе на предавторизацию или в личном кабинете.

Отмена предавторизации

В случае, если по предавторизации не будет подтверждения, необходимо осуществить ее отмену.

Для отмены предавторизации необходимо отправить запрос на возврат средств согласно 04 Формирование запроса на отмену платежа. В случае, если транзакция была предавторизована, но не подтверждена, средства вернутся плательщику в виде отмены, иначе - в виде возврата.

В случае, если по предавторизации не будет получено ни отмены, ни подтверждения, отмена произойдет автоматически через 30 дней.

Примеры расчета сигнатуры на разных языках

PHP

Online пример - <https://repl.it/@miloserdovayed/CalmBlushingIntegrationtesting#main.php>

PHP сигнатура

/* Определяем массив с параметрами для расчета.

В этот массив должны войти все параметры, отправляемые в вашей форме (за исключением самого поля signature, значение которого вычисляем).

```

Получив вашу форму, система ИЭ аналогичным образом вычислит из ее параметров
signature и сравнит значение с вычисленным на стороне вашего магазина. */
$items = array(
    'testing' => '1',
    'salt' => 'dPUTLtbMfcTGzkaBnGtseKlcQymCLrYI',
    'order_id' => '14425840',
    'amount' => '973',
    'merchant' => 'ad25ef06-1824-413f-8ef1-c08115b9b979',
    'description' => 'Заказ №14425840',
    'client_phone' => '+7 912 9876543',
    'client_email' => 'test@test.ru',
    'success_url' => 'http://myawesomesite.com/payment_success',
    'receipt_contact' => 'test@mail.com',
    'receipt_items' => '[{"discount_sum": 40, "name": "Товар 1",
"payment_method": "full_prepayment", "payment_object": "commodity", "price":
48, "quantity": 10, "sno": "osn", "vat": "vat10"}, {"name": "Товар 2",
"payment_method": "full_prepayment", "payment_object": "commodity", "price":
533, "quantity": 1, "sno": "osn", "vat": "vat10"}]',
    'unix_timestamp' => '1573451160'
);

echo 'Signature: '. get_signature($items);

/* Двойное шифрование sha1 на основе секретного ключа
подставьте ваш секретный ключ вместо 00112233445566778899aabbccddeeff */
function double_sha1($data) {
    for ($i = 0; $i < 2; $i++) {
        $data = sha1('00112233445566778899aabbccddeeff' . $data);
    }
    return $data;
}

/* Вычисляем подпись (signature). Подпись считается на основе склеенной
строки из отсортированного массива параметров, исключая из расчета пустые
поля и элемент "signature" */
function get_signature(array $params, $key = 'signature') {
    $keys = array_keys($params);
    sort($keys);
    $chunks = array();
    foreach ($keys as $k) {
        $v = (string) $params[$k];
        if (($v !== '') && ($k !== 'signature')) {
            $chunks[] = $k . '=' . base64_encode($v);
        }
    }
    $data = implode('&', $chunks);

    echo 'data: '.$data."\n";
    $sig = double_sha1($data);

    return $sig;
}

```

NodeJS

Online пример - <https://repl.it/@miloserdovayed/EasygoingBountifulEngineer>

JS расчет сигнатуры

```
const Base64 = require('js-base64').Base64;
const SHA1 = require('crypto-js/sha1');

/* Определяем массив с параметрами для расчета.
В этот массив должны войти все параметры, отправляемые в вашей форме (за исключением самого поля signature, значение которого вычисляем).
Получив вашу форму, система ИЭ аналогичным образом вычислит из ее параметров signature и сравнит значение с вычисленным на стороне вашего магазина. */
const data = {
  'testing': '1',
  'salt': 'dPUTLtbMfcTGzkaBnGtseKlcQymCLrYI',
  'order_id': '14425840',
  'amount': '973',
  'merchant': 'ad25ef06-1824-413f-8ef1-c08115b9b979',
  'description': 'Заказ №14425840',
  'client_phone': '+7 912 9876543',
  'client_email': 'test@test.ru',
  'success_url': 'http://myawesomesite.com/payment_success',
  'receipt_contact': 'test@mail.com',
  'receipt_items': '[{"discount_sum": 40, "name": "Товар 1",
"payment_method": "full_prepayment", "payment_object": "commodity", "price":
48, "quantity": 10, "sno": "osn", "vat": "vat10"}, {"name": "Товар 2",
"payment_method": "full_prepayment", "payment_object": "commodity", "price":
533, "quantity": 1, "sno": "osn", "vat": "vat10"}]',
  'unix_timestamp': '1573451160'
};

/* подставьте ваш секретный ключ вместо 00112233445566778899aabbccddeeff */
const key = '00112233445566778899aabbccddeeff';

console.log('Signature: ', GetSignature(key, data));

/* Вычисляем подпись (signature). Подпись считается на основе склеенной
строки из отсортированного массива параметров, исключая из расчета пустые
поля и элемент "signature" */
function GetSignature(secretKey, formData) {
  const values = Object.keys(formData)
    .filter(key => key !== 'signature')
    .filter(key => formData[key] !== '')
    .sort()
    .map(key => `${key}=${Base64.encode(formData[key])}`)
    .join('&');

  console.log('data: ', values)

  /* Двойное шифрование sha1 на основе секретного ключа */
  const signature = SHA1(secretKey + SHA1(secretKey + values));

  return signature.toString();
}
```

Python

Online пример - <https://repl.it/@miloserdovayed/DrabCoarseGraphicslibrary#main.py>

Python сигнатура

```

import base64
import hashlib

def get_raw_signature(params):
    chunks = []

    for key in sorted(params.keys()):
        if key == 'signature':
            continue

        value = params[key]

        if isinstance(value, str):
            value = value.encode('utf8')
        else:
            value = str(value).encode('utf-8')

        if not value:
            continue

        value_encoded = base64.b64encode(value)
        chunks.append('%s=%s' % (key, value_encoded.decode()))

    raw_signature = '&'.join(chunks)
    return raw_signature

'''Двойное шифрование sha1 на основе секретного ключа'''
def double_shal(secret_key, data):
    sha1_hex = lambda s: hashlib.sha1(s.encode('utf-8')).hexdigest()
    digest = sha1_hex(secret_key + sha1_hex(secret_key + data))
    return digest

'''Вычисляем подпись (signature). Подпись считается на основе склеенной
строки из отсортированного массива параметров, исключая из расчета пустые
поля и элемент "signature" '''
def get_signature(secret_key: str, params: dict) -> str:
    return double_shal(secret_key, get_raw_signature(params))

'''Определяем словарь с параметрами для расчета.
В этот массив должны войти все параметры, отправляемые в вашей форме (за
исключением самого поля signature, значение которого вычисляем).
Получив вашу форму, система ИЭ аналогичным образом вычислит из ее параметров
signature и сравнит значение с вычисленным на стороне вашего магазина.
подставьте ваш секретный ключ вместо 00112233445566778899aabbccddeeff
'''
if __name__ == '__main__':
    items = {
        "testing": '1',
        "salt": 'dPUTLtbMfcTGzkaBnGtseKlcQymCLrYI',
        "order_id": '14425840',
        "amount": '973',
        "merchant": 'ad25ef06-1824-413f-8ef1-c08115b9b979',
        "description": 'Заказ №14425840',
        "client_phone": '+7 912 9876543',
        "client_email": 'test@test.ru',
        "success_url": 'http://myawesomesite.com/payment_success',
        "receipt_contact": 'test@mail.com',
        "receipt_items": '[{"discount_sum": 40, "name": "Товар 1",
"payment_method": "full_prepayment", "payment_object": "commodity", "price":
48, "quantity": 10, "sno": "osn", "vat": "vat10"}, {"name": "Товар 2",

```

```

"payment_method": "full_prepayment", "payment_object": "commodity", "price":
533, "quantity": 1, "sno": "osn", "vat": "vat10"}]'],
    "unix_timestamp": '1573451160'
}
print(get_signature('00112233445566778899aabbccddeeff', items))

```

C#

Online пример - <https://repl.it/@miloserdovayed/NotableSardonicGeneric#main.cs>

C# сигнатура

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;

class MainClass {
    public static void Main (string[] args) {
        var data = new Dictionary<string, string>()
        {
            {"testing", "1"},
            {"salt", "dPUTLtbMfcTGzkaBnGtseKlcQymCLrYI"},
            {"order_id", "14425840"},
            {"amount", "973"},
            {"merchant", "ad25ef06-1824-413f-8ef1-c08115b9b979"},
            {"description", "Заказ №14425840"},
            {"client_phone", "+7 912 9876543"},
            {"client_email", "test@test.ru"},
            {"success_url", "http://myawesomesite.com/payment_success"},
            {"receipt_contact", "test@mail.com"},
            {"receipt_items", "[{"discount_sum": 40, "name": "Товар 1",
            {"payment_method": "full_prepayment", "payment_object": "commodity",
            {"price": 48, "quantity": 10, "sno": "osn", "vat": "vat10"},
            {"name": "Товар 2", "payment_method": "full_prepayment",
            {"payment_object": "commodity", "price": 533, "quantity": 1, "sno":
            {"osn", "vat": "vat10"}]}"],
            {"unix_timestamp", "1573451160"}
        };

        Console.WriteLine(SignatureCalculator.GetSignature("00112233445566778899aabbccddeeff", data));
    }
}

public static class SignatureCalculator
{
    public static string GetSignature(string secretKey,
    IDictionary<string, string> formData)
    {
        var keys = formData
            .Where(x=>x.Value != "")
            .Where(x=> x.Key != "signature")
            .Select(x => new { x.Key, Value =
            $"{x.Key}={GetBase64Val(x.Value)}" })
    }
}

```

```

        .OrderBy(x => x.Key)
        .ToArray();

    var values = string.Join("&", keys.Select(x => x.Value));

    Console.WriteLine(values);
    var signature = SHA1(secretKey + SHA1(secretKey + values));

    return signature;
}

private static string GetBase64Val(object plainText)
{
    var value = plainText == null ? string.Empty :
plainText.ToString();

    var plainTextBytes = Encoding.UTF8.GetBytes(value);
    return Convert.ToBase64String(plainTextBytes);
}

// ReSharper disable once InconsistentNaming
private static string SHA1(string input)
{
    using (var sha1 = new SHA1Managed())
    {
        var hash = sha1.ComputeHash(Encoding.UTF8.GetBytes(input));
        var sb = new StringBuilder(hash.Length * 2);

        foreach (var b in hash)
        {
            // can be "x2" if you want lowercase
            sb.Append(b.ToString("x2"));
        }

        return sb.ToString();
    }
}
}

```