

moneta | ru

**Основные способы
интеграции с PayAnyWay**

Содержание

Введение.....	4
Глава 1. PayAnyWay статическая кнопка.....	5
Формирование платежной кнопки через конструктор.....	5
Формирование платежной кнопки через MONETA.Assistant.....	5
Добавление описания операции через MONETA.Assistant.....	6
Преобразование формы оплаты в ссылку.....	8
Встраивание формы оплаты на страницу сайта.....	8
Формирование платежной кнопки через SDK.....	8
Платежная кнопка через SDK PHP.....	8
Платежная кнопка через SDK Python3.....	8
Платежная кнопка через SDK Ruby.....	9
Платежная кнопка через SDK Android.....	9
Платежная кнопка через SDK PhoneGap.....	9
Глава 2. PayAnyWay интеграция.....	10
Интеграция PayAnyWay общие сведения.....	10
Интеграция с CMS.....	13
Интеграция с PayAnyWay в мобильном приложении.....	13
Реализация Pay URL.....	13
Преобразование платежной формы в iframe.....	13
Взаимодействие с виджетом оплаты, выводимым через iframe.....	14
Особенности перехода на страницы Success, Fail URL при использовании виджета в iframe.....	14
Динамические возможности Assistant.widget.....	15
Проверка наличия товара и его цены перед оплатой (Check URL).....	16
Глава 3. PayAnyWay выписка счёта.....	20
Создание счёта через Merchant API.....	20
Пример кода на SDK php.....	21
Пример кода на SDK C#.....	21
Глава 4. PayAnyWay с холдированием.....	22
Пример кода на SDK php.....	23
Глава 5. PayAnyWay регулярные платежи.....	25
Реализация регулярных платежей на SDK PHP.....	26
Перед началом работы.....	26
Генерация формы рекуррентного платежа.....	26
Генерация формы рекуррентного платежа.....	27
Рассылка уведомлений о предстоящем платеже.....	27
Осуществление регулярных платежей.....	27
Обработчик отмены регулярного платежа.....	27
Ручное проведение регулярного платежа.....	27
Глава 6. PayAnyWay выплаты и списания.....	28
Выплаты через merchant API.....	28
Выплаты на банковскую карту.....	29
Списание с карты и выплаты для проектов с PCI DSS.....	31

Временное сохранение данных карты (в мобильном приложении).....	31
Списание с банковской карты через merchant API.....	32
Глава 7. PayAnyWay площадкам.....	33
Глава 8. Полезные ссылки.....	34
SDK на различных языках.....	34
Документация по MONETA.Assistant.....	34
Описание методов MONETA.MerchantAPI.....	34

Введение

Документ дает информацию об основных способах интеграции с PayPalWau.

Документ адресован разработчикам, имеющим базовые знания в области программирования.

Chapter

1

PayAnyWay статическая кнопка

Topics:

- [Формирование платежной кнопки через конструктор](#)
- [Формирование платежной кнопки через MONETA.Assistant](#)
- [Добавление описания операции через MONETA.Assistant](#)
- [Преобразование формы оплаты в ссылку](#)
- [Встраивание формы оплаты на страницу сайта](#)
- [Формирование платежной кнопки через SDK](#)

Формирование платежной кнопки через конструктор

Если Ваш проект предлагает небольшое количество товаров или услуг, то Вам может подойти простая кнопка оплаты.

Html-код кнопки можно вставить на любой сайт, любую страницу любой сложности. Чтобы получить код кнопки, нужно войти в личный кабинет системы Монета.ру, затем перейти в раздел **“Рабочий кабинет”**, а затем на страницу **“Список товаров и услуг”**.

На этой странице добавьте Ваш товар, укажите стоимость и нажмите **“Сохранить”**. Затем перейдите на страницу товара, кликнув на его наименование - здесь Вы увидите html код кнопки. Скопируйте его и вставьте на страницу Вашего сайта:

```
<form name="1182_buyNow_simple_button" id="1182_buyNow_simple_button" action="https://moneta.ru/shopping-11493408.htm" method="post">
  <fieldset>
    <input type="hidden" name="MNT_ITEM_ID" value="1182" />
    <input type="hidden" name="MNT_QUANTITY" value="1" />
    <input type="submit" value="Купить" />
    <!-- дополнительные параметры -->
  </fieldset>
</form>
```

Формирование платежной кнопки через MONETA.Assistant

Кнопку оплаты можно построить самостоятельно, не используя конструктор. Для этого нужно создать html-форму со следующими обязательными полями:

- **MNT_ID** - Идентификатор магазина в системе MONETA.RU. Соответствует номеру расширенного счета магазина.

- **MNT_AMOUNT** - Сумма оплаты. Десятичные символы отделяются точкой. Количество знаков после запятой - максимум два символа (если параметр не задан, то сумма будет запрошена в учетной системе магазина соответствующим проверочным запросом на Check URL, который указывается в настройках счёта в личном кабинете в системе Moneta.ru).

и необязательными:

- **MNT_TRANSACTION_ID** - Внутренний идентификатор заказа, однозначно определяющий заказ в магазине. Ограничение на размер – 255 символов.
- **MNT_CURRENCY_CODE** - ISO код валюты, в которой производится оплата заказа в магазине. Значение должно соответствовать коду валюты счёта получателя (MNT_ID). Например, RUB, USD, EUR.
- **MNT_TEST_MODE** - Указание, что запрос происходит в тестовом режиме. Возможные значения 0 или 1. Если включено (значение 1), то реального списания и зачисления средств не произойдет. В тестовом режиме не происходит передача управления от iframe с виджетом оплаты родительскому окну.
- **MNT_DESCRIPTION** - Описание оплаты. Максимальная длина 500 символов.

Полный список полей и их описание приведено в документации [MONETA.Assistant](#).

Ниже приводится пример формы, реализующей платежную кнопку, построенную на базе Moneta.Assistant:

```
<form name="payanyway_from" id="payanyway_from" action="https://moneta.ru/assistant.htm" method="post">
  <fieldset>
    <input type="hidden" name="MNT_ID" value="НОМЕР_РАСШИРЕННОГО_СЧЁТА" />
    <input type="hidden" name="MNT_TEST_MODE" value="0" />
    <p>Сумма: <input type="text" name="MNT_AMOUNT" value="12.00" /></p>
    <input type="submit" value="Купить" />
  </fieldset>
</form>
```

Обратите внимание на то, что форма отправляется на URL: <https://moneta.ru/assistant.htm>

Добавление описания операции через MONETA.Assistant

Если Вы хотите добавить поле для пользователя, в которое он будет вводить, его контактный телефон или другую информацию, то добавьте следующую строчку в html-код Вашей формы:

```
<p>Номер телефона: <input type="text" name="MNT_DESCRIPTION" /></p>
```

Перед строчкой:

```
<input type="submit" value="Купить" />
```

Переданная пользователем информация будет записана в систему Монета.ру. Её можно будет просматривать в истории операций по Вашему расширенному счёту.

Если на Вашем сайте допустима вставка JavaScript-кода, то Вы можете добавить не одно поле, а несколько. Ниже приведён JavaScript-код, который перед нажатием на кнопку “Купить”, соберет все предоставленные данные в параметр MNT_DESCRIPTION:

```
<input type="hidden" name="MNT_DESCRIPTION" id="form_description" />
<p>Номер телефона: <input type="text" name="user_phone" id="user_phone" /></p>
<p>ФИО: <input type="text" name="user_fio" id="user_fio" /></p>
<p>Комментарий: <input type="text" name="user_comment" id="user_comment" /></p>

<script type="text/javascript">
  var _mainFormId = 'payanyway_from';

  function submitForm() {
    document.getElementById(_mainFormId).submit();
  }
</script>
```

```

}

window.onload=function() {
  document.getElementById(_mainFormId).onsubmit = function(event) {
    event.preventDefault();
    var phoneElt = document.getElementById('user_phone');
    var fioElt = document.getElementById('user_fio');
    var commentElt = document.getElementById('user_comment');
    var phoneVal = phoneElt.value;
    var fioVal = fioElt.value;
    var commentVal = commentElt.value;
    var cmElementObject = document.getElementById('form_description');
    var cmElementSetValue = phoneVal + fioVal + commentVal;
    cmElementObject.value = cmElementSetValue;
    setTimeout(submitForm, 3000);
    return false;
  }
}
</script>

```

Данный код нужно вставить в html-код Вашей платежной формы перед строкой:

```
<input type="submit" value="Купить" />
```

Для проектов, которым достаточно для работы данного решения, нужно соответствующим образом настроить расширенный счёт магазина в системе Moneta.ru.

Нужно установить “Тип интерфейса” в значение “MONETA.Assistant”, поле “Код проверки целостности данных” нужно оставить пустым, а опцию “Подпись формы оплаты обязательна” установите в значение “Нет”.

Чтобы защитить Вашу форму от изменения суммы покупателем, надо передать параметр MNT_AMOUNT как скрытый:

```
<input type="hidden" name="MNT_AMOUNT" value="12.00" />
```

А также следует добавить в форму сигнатуру (подпись).

В этом случае настройки счета должны быть изменены. В поле “Код проверки целостности данных” нужно ввести произвольное значение, а опцию “Подпись формы оплаты обязательна” установить в значение “Да”.

Чтобы получить подпись платежной формы, надо выполнить следующий php-код:

```

<?php
  $MNT_ID = 'НОМЕР_РАСШИРЕННОГО_СЧЁТА';
  $MNT_TRANSACTION_ID = 'ПОРЯДКОВЫЙ_НОМЕР_ОПЕРАЦИИ';
  $MNT_AMOUNT = '12.00';
  $MNT_CURRENCY_CODE = 'RUB';
  $MNT_TEST_MODE = '0';
  $code = 'КОД_ПРОВЕРКИ_ЦЕЛОСТНОСТИ';

  echo md5( $MNT_ID . $MNT_TRANSACTION_ID . $MNT_AMOUNT . $MNT_CURRENCY_CODE .
  $MNT_TEST_MODE . $code );
?>

```

Сделать это (выполнить код) можно здесь: <http://sandbox.onlinephpfunctions.com>

Затем, полученный код, надо добавить в Вашу платежную форму:

```
<input type="hidden" name="MNT_SIGNATURE" value="ЗНАЧЕНИЕ_КОТОРОЕ_ПОКАЖЕТ_PHP_КОД" />
```

Преобразование формы оплаты в ссылку

Если стоит задача сформировать не форму, а ссылку оплаты, то следует иметь ввиду, что параметры платежной формы можно передавать через GET запрос. Например так: https://moneta.ru/shopping-11493408.htm?MNT_ITEM_ID=1182&MNT_QUANTITY=1

Ссылка на MONETA.Assistant может быть такой: https://moneta.ru/assistant.htm?MNT_ID=НОМЕР_РАСШИРЕННОГО_СЧЁТА&MNT_TEST_MODE=0&MNT_AMOUNT=12

Можно добавить и описание операции по аналогии с формой оплаты: https://moneta.ru/assistant.htm?MNT_ID=НОМЕР_РАСШИРЕННОГО_СЧЁТА&MNT_TEST_MODE=0&MNT_AMOUNT=12&MNT_DESCRIPTION=ОПИСАНИЕ_ОПЕРАЦИИ

Сформированную таким образом ссылку, можно отправить покупателю на e-mail, а также разместить в социальной сети, например: Вконтакте, Instagram и т.п.

Встраивание формы оплаты на страницу сайта

Ссылку можно включить на свою html страницу через iframe с небольшим изменением, касающимся URL:

```
<iframe width="600" height="800" src="https://moneta.ru/assistant.widget?MNT_ID=НОМЕР_РАСШИРЕННОГО_СЧЁТА&MNT_TEST_MODE=0&MNT_AMOUNT=12"></iframe>
```

В случае использования виджета через iframe, вместо кнопки “Купить” платежной формы, оплата будет возможна сразу на Вашем сайте.

Содержимое iframe можно стилизовать индивидуально по Вашему усмотрению. Для этого в личном кабинете Moneta.ru используйте ссылку “Редактировать стиль Moneta.Assistant” для вставки Вашего индивидуального css-кода, меняющего стандартный assistant.widget системы Moneta.ru. Примеры страниц, на которых выводится виджет MONETA.Assistant с измененным внешним видом при помощи дополнительного css-кода можно увидеть здесь: <https://payanyway.ru/info/w/ru/public/w/partnership/developers/widget-css.html>

Там же есть листинги css-кода по каждому из приведенных примеров.

Формирование платежной кнопки через SDK

Платежная кнопка через SDK PHP

Скачайте и настройте [SDK PHP](#) согласно [инструкции](#).

Пример кода, формирующий кнопку “Оплатить”:

```
$result = $monetaSDK->showChoosePaymentSystemForm();
echo $result->render;
```

Платежная кнопка через SDK Python3

Скачайте и настройте [SDK Python3](#) согласно [инструкции](#).

Метод для отрисовки кнопки “Оплатить”:

```
result = templatePaymentFrom(orderId, amount, currency='RUB', description='',
paysys='')
```

Где

- **orderId** - номер операции, например 370429,
- **amount** - сумма операции,
- **currency** - валюта операции, например, RUB,

- **description** - описание операции,
- **paaysys** - заданный способ оплаты заказа, строка, например, 'plastic' для оплаты банковскими картами.

Платежная кнопка через SDK Ruby

Скачайте и настройте [SDK Ruby](#) согласно [инструкции](#).

Чтобы получить ссылку на платежный шлюз для оплаты заказа пользователем, используйте метод:

```
Payanyway::Gateway.payment_url(params, use_signature = true)
```

Платежная кнопка через SDK Android

Скачайте и настройте [SDK Android](#) согласно [инструкции](#).

Чтобы пользователь увидел платежную форму, необходимо запустить следующий код, размещённый в файле ..Activity.java:

```
Double mntAmount = 12.00;
String mntPaymentSystem = "plastic";
String mntOrderId = monetask.getOrderId();
String mntCurrency = "RUB";
WebView myWebView = (WebView) findViewById(R.id.webView);
monetasdk.showPaymentFrom(mntOrderId, mntAmount, mntCurrency, mntPaymentSystem,
    myWebView, this);
```

Где

- **mntOrderId** - идентификатор операции,
- **mntAmount** - сумма операции,
- **mntCurrency** - валюта операции,
- **mntPaymentSystem** - способ оплаты.

Платежная кнопка через SDK PhoneGap

Скачайте и настройте [SDK PhoneGap](#) согласно [инструкции](#).

Чтобы пользователь увидел платежную форму, необходимо запустить следующий код, размещённый в файле index.html:

```
var mntOrderId,
    mntAmount = 12,
    mntCurrency,
    mntPaymentSystem = 'plastic',
    payFormElementId = 'payanyway',
    payFormElementWidth = 333,
    payFormElementHeight = 520;

monetaSdk.showPaymentFrom(mntOrderId, mntAmount, mntCurrency, mntPaymentSystem,
    payFormElementId, payFormElementWidth, payFormElementHeight);
```

Где

- **mntOrderId** - идентификатор операции,
- **mntAmount** - сумма операции,
- **mntCurrency** - валюта операции,
- **mntPaymentSystem** - способ оплаты,
- **payFormElementId** - id блока, в который будет выводиться форма оплаты,
- **payFormElementWidth** - ширина формы оплаты,
- **payFormElementHeight** - высота формы оплаты.

Chapter 2

PayAnyWay интеграция

Topics:

- *Интеграция PayAnyWay общие сведения*
- *Интеграция с CMS*
- *Интеграция с PayAnyWay в мобильном приложении*
- *Реализация Pay URL*
- *Преобразование платежной формы в iframe*
- *Взаимодействие с виджетом оплаты, выводимым через iframe*
- *Проверка наличия товара и его цены перед оплатой (Check URL)*

Интеграция PayAnyWay общие сведения

Интеграция оплаты товаров или услуг имеет общие принципы для всех возможных случаев, будь то интеграция с веб-сайтом или интернет-магазином. В любом случае, можно выделить общую схему взаимодействия с системой Moneta.ru:



Показать форму оплаты клиенту можно одним из трёх способов: 1) Перенаправив его на MONETA.Assistant 2) Показав форму MONETA.Assistant в iframe 3) Запросить данные банковской карты и осуществить списание через Merchant API (метод PaymentRequest) системы Moneta.ru (для этого способа магазину потребуется сертификат PCI DSS).

Способы 1 и 2 были рассмотрены в главе 1, способ 3 будет рассмотрен в главе 6.

После успешной оплаты система Moneta.ru обращается к серверной части приложения партнера по адресу, который указывается в настройках расширенного счёта в поле “Pay URL”. На указанный URL система Moneta.ru отправляет запрос со следующими параметрами:

MNT_ID

Идентификатор магазина в системе MONETA.RU.

MNT_TRANSACTION_ID

Внутренний идентификатор заказа, однозначно определяющий заказ в магазине.

MNT_OPERATION_ID

Номер операции в системе MONETA.RU.

MNT_AMOUNT

Фактическая сумма, полученная на оплату заказа, без учета комиссии за услуги системы «MONETA.RU».

MNT_CURRENCY_CODE

ISO код валюты, в которой произведена оплата заказа в магазине.

MNT_SUBSCRIBER_ID

Внутренний идентификатор пользователя, однозначно определяющий получателя в учетной системе магазина.

MNT_TEST_MODE

Если оплата произведена в тестовом режиме, то параметр содержит «1», если в реальном – «0».

MNT_SIGNATURE

Код для идентификации отправителя и проверки целостности данных.

MNT_USER

Номер счета пользователя, если оплата производилась с пользовательского счета в системе «MONETA.RU».

paymentSystem.unitId

Идентификатор платежной системы, если оплата производилась с платежной системы отличной от «MONETA.RU».

MNT_CORRACCOUNT

Номер счета плательщика. Если оплата производилась с пользовательского счета в системе «MONETA.RU», то совпадает с MNT_USER.

MNT_CUSTOM1, MNT_CUSTOM2, MNT_CUSTOM3, Другие параметры

Параметры, переданные в запросе на оплату через MONETA.Assistant

Параметры могут быть переданы методом POST или GET в зависимости от настроек расширенного счёта.

Первое что нужно сделать в скрипте “Pay URL” - проверить подпись **MNT_SIGNATURE** (то есть достоверность запроса).

Запрос является достоверным, если присланное значение **MNT_SIGNATURE** идентично вычисленному следующим образом:

```
MNT_SIGNATURE = MD5(MNT_ID + MNT_TRANSACTION_ID + MNT_OPERATION_ID +
MNT_AMOUNT + MNT_CURRENCY_CODE + MNT_SUBSCRIBER_ID + MNT_TEST_MODE
+ КОД ПРОВЕРКИ ЦЕЛОСТНОСТИ ДАННЫХ)
```

Код проверки целостности данных - является свойством расширенного счёта и устанавливается в личном кабинете в системе Moneta.ru. Данный код должен быть так же известен серверной стороне приложения интернет-магазина, где реализуется скрипт Pay URL.

При формировании подписи, **MNT_AMOUNT** должно быть с двумя десятичными знаками, отделенными точкой.

После проверки подлинности запроса, скрипт Pay URL может изменить статус заказа в интернет-магазине на “Оплачен”, выполнить иные сопутствующие действия.

Скрипт должен отдать системе монета ответ со статусом **200 OK** в заголовке и словом **SUCCESS** в случае успеха, либо **FAIL** в случае ошибки в текстовом виде, без переносов.

Можно ответить на запрос Pay URL в XML формате (со статусом **200 OK**):

```
<?xml version="1.0" encoding="UTF-8"?>
<MNT_RESPONSE>
  <MNT_ID></MNT_ID>
  <MNT_TRANSACTION_ID></MNT_TRANSACTION_ID>
  <MNT_RESULT_CODE></MNT_RESULT_CODE>
  <MNT_DESCRIPTION></MNT_DESCRIPTION>
  <MNT_AMOUNT></MNT_AMOUNT>
  <MNT_SIGNATURE></MNT_SIGNATURE>
  <MNT_ATTRIBUTES>
    <ATTRIBUTE>
      <KEY></KEY>
      <VALUE></VALUE>
    </ATTRIBUTE>
  </MNT_ATTRIBUTES>
</MNT_RESPONSE>
```

Где:

MNT_ID

Идентификатор магазина в системе MONETA.RU.

MNT_TRANSACTION_ID

Внутренний идентификатор заказа, однозначно определяющий заказ в магазине.

MNT_RESULT_CODE

Код ответа на запрос.

MNT_DESCRIPTION

Описание состояния заказа.

MNT_AMOUNT

Сумма оплаты.

MNT_SIGNATURE

Код для идентификации отправителя и проверки целостности данных.

MNT_ATTRIBUTES

Содержит произвольные параметры, которые будут сохранены в операции.

ATTRIBUTE

Элемент представляет один произвольный параметр операции.

KEY

Уникальное название параметра операции. Не должно превышать 32 символа.

VALUE

Значение параметра операции.

Ответ подписывается подписью **MNT_SIGNATURE**, которая формируется следующим образом:

```
MNT_SIGNATURE = MD5( MNT_RESULT_CODE + MNT_ID + MNT_TRANSACTION_ID
+ КОД ПРОВЕРКИ ЦЕЛОСТНОСТИ ДАННЫХ )
```

Как уже было отмечено выше, при формировании подписи, **MNT_AMOUNT** должно быть с двумя десятичными знаками, отделенными точкой.

Коды ответа **MNT_RESULT_CODE** могут принимать следующие значения:

- **200** - заказ оплачен. Уведомление об оплате магазину доставлено,
- **100, 302, 402** - отправка уведомления об оплате должна быть и будет повторена,
- **500** - ошибка обработки. Автоматическая отправка уведомлений от системы Moneta.ru будет остановлена. Необходимо связаться с группой поддержки MONETA.RU.

Важно отметить, что ответ системе Moneta.ru на запрос Pay URL должен быть сформирован с заголовком 200 OK, даже если не было принято никаких входящих параметров, поскольку при сохранении ссылки на скрипт Pay URL в личном кабинете Moneta.ru, будет отправлен проверочный запрос без параметров, на который ожидается ответ со статусом 200 OK в заголовке.

Интеграция с CMS

Для многих CMS для интернет-магазинов уже реализованы *модули для приема оплаты через PayAnyWay*.

Модули работают по выше рассмотренной схеме, то есть подразумевают установку “Pay URL” в настройках расширенного счёта в системе Moneta.ru. Скрипт “Pay URL” сменит статус заказа на “Оплачен” в каждой конкретной CMS.

Большинство модулей PayAnyWay построено таким образом, чтобы ограничение выбора способов оплаты включалось через настройки счёта в личном кабинете системы Moneta.ru (если такая необходимость существует).

Интеграция с PayAnyWay в мобильном приложении

Приложения интернет-магазинов, как правило, разрабатываются по клиент-серверной архитектуре. Каталог товаров хранится в backend и отображается на frontend. Заказы формируются на frontend и отправляются в backend.

В соответствии с этим, на frontend отображается форма оплаты. Для этого можно использовать SDK, рассмотренные в п.1.6.4 и 1.6.5.

Pay URL следует разместить на backend.

Реализация Pay URL

Реализовать скрипт, обрабатывающий запрос об успешной оплате можно на любом языке программирования, который позволяет реализовать веб-сервис. Общее описание алгоритма работы этого скрипта приводится в п.2.1.

В SDK PHP есть метод, который позволяет реализовать скрипт Pay URL в нескольких строчках кода:

```
$monetaSDK = new Moneta\MonetaSdk();
$result = $monetaSDK->processInputData();
echo $result->render;
exit;
```

В результате работы данный код выводит **SUCCESS** или **FAIL** в зависимости от поступивших в веб-сервис данных.

Преобразование платежной формы в iframe

В модулях PayAnyWay для CMS реализована стандартная для интернет-магазинов схема работы, в которой после выбора способа оплаты выводится форма оплаты с кнопкой “Оплатить”, при нажатии на которую происходит переход на форму ввода данных плательщика MONETA.Assistant.

На некоторых сайтах можно счесть возможным размещение вместо кнопки “Оплатить” платежной формы MONETA.Assistant через iframe.

Для того, чтобы не меняя код платежного модуля и CMS преобразовать платежную форму в iframe надо на страницу оформления заказа, либо на все страницы сайта поместить следующий Javascript-код:

```
<script>
jQuery(document).ready(function($) {
  // page is ready
  var pawFormContent = $('#payment-form').html();
  if (pawFormContent != undefined && pawFormContent != '') {
    var pawFormParent = $('#payment-form').parent();
    var pawSuccess = $('.success').html();

    // generate iframe code
    var pawFormData = $('#payment-form').serialize();
    var pawIframeLink = 'https://www.payanyway.ru/assistant.widget?' + pawFormData;
    var pawIframe = '<h1>0плата заказа:</h1><iframe width="600" height="800" src="' +
    pawIframeLink + '">iframe is not supported</iframe>';

    if (pawSuccess != undefined && pawSuccess != '') {
      $('.success').html(pawIframe);
    }
    else {
      pawFormParent.html(pawIframe);
    }

    // remove paw payment button
    $('#payment-form').remove();
  }
});
</script>
```

Где:

#payment-form - id формы оплаты,

.success - класс блока для вставки виджета оплаты.

Данный код можно использовать в CMS Webasyst без каких-либо доработок. При использовании в другой CMS, следует внести изменения в части идентификаторов.

Взаимодействие с виджетом оплаты, выводимым через iframe

При использовании виджета оплаты, выводимого через iframe, вместо перехода на платежную форму MONETA.Assistant, можно изменять визуальное отображение его элементов, установив собственные CSS стили. Сделать это можно через личный кабинет в системе Moneta.ru

Примеры оформления платежной формы, выводимой через iframe можно посмотреть [здесь \(перейдите по ссылке\)](#).

Особенности перехода на страницы Success, Fail URL при использовании виджета в iframe

Платежная форма позволяет перенаправить покупателя на страницы магазина после успешной оплаты (Success URL), после отмененной оплаты (Fail URL), после запроса на холдирование средств на карте покупателя (InProgress URL), после добровольного отказа от оплаты (Return URL). Логично организовать переход так, чтобы он происходил не внутри iframe, а в родительском окне. Для этого в настройках расширенного счёта в личном кабинете Moneta.ru есть настройка **“Target (для возврата iframe)”**. В рассматриваемом случае следует выбрать `_parent` или `_top`.

Некоторые CMS могут иметь встроенный Javascript-код, который блокирует переходы из дочернего (по отношению к сайту интернет-магазина) iframe на указанный URL. Поэтому, если управление переходом на Success и Fail URL работает некорректно, воспользуйтесь следующими рекомендациями:

1) В качестве страницы Success URL создайте минимальную html страницу (без дизайна) с кодом:

```
<p>Спасибо за заказ!</p>
<a onclick="parent.callsuccess();" href="#">Вернуться на сайт</a>
```



```

        console.log('Parent, status(JSON): ' + msg.status);
        break;
    }
}
}
}

if (window.addEventListener) {
    window.addEventListener("message", listener, false);
} else {
    window.attachEvent("onmessage", listener);
}

function setBodyClass() {
    var msg = {
        m_type: "bodyClass",
        m_val: "test_class_name"
    };

    var ifr = document.getElementById('pawassistantiframe').contentWindow;
    ifr.postMessage(JSON.stringify(msg), '{url}');
}

```

Примеры postMessage запросов. Сообщение, передаваемое через интерфейс postMessage текстовое и представляет собой сериализованный JSON.

В каждом передаваемом объекте есть ключ m_type. На данный момент, возможные значения ключа таковы: widgetSize, status, bodyClass.

Сообщения с типами widgetSize и status передаются родителю при изменении соответствующих параметров. Так же данные сообщения с актуальными параметрами можно получить по запросу.

Примеры сообщений:

```

1) {"m_type": "widgetSize", "width": "300", "height": "64"}
2) {"m_type": "status", "status": "none"}

```

Примеры запросов:

```

1) Запрос статуса: {"m_type": "request", "m_val": "status"}
2) Запрос размера виджета: {"m_type": "request", "m_val": "widgetSize"}
3) Запрос на submit формы: {"m_type": "request", "m_val": "submitForm"}
4) Запрос на установку css класса для body виджета:
{"m_type": "bodyClass", "m_val": "class_name"}

```

Ответы виджета:

```

1) Класс добавлен: {"m_type": "bodyClass", "result": "success", "comment": ""}
2) Класс уже был, повторно не добавился:
{"m_type": "bodyClass", "result": "error", "comment": "Class already exist"}
3) Недопустимые символы в названии класса (регуляры названия класса: '([\^A-Za-z0-9\-\_])'):
{"m_type": "bodyClass", "result": "error", "comment": "Wrong class name"}

```

Проверка наличия товара и его цены перед оплатой (Check URL)

Интернет-магазину может понадобиться предварительная проверка параметров заказа перед оплатой. Например, в случае продажи товаров, остатки которых могут быстро меняться, либо товаров, по которым может меняться цена с течением времени.

В подобных случаях, можно использовать механизм проверочных запросов, которые система Moneta.ru отправляет на URL, указанный в свойствах расширенного счёта в поле "Check URL".

Проверочные запросы служат для того, чтобы:

- Убедиться, что заказ существует, что заказ еще не оплачен, что срок действия заказа не истек.
- Указать сумму заказа, если магазин ранее не передал сумму заказа в HTML-форме.

- Проверить статус заказа, если товар уже оплачен.

Проверочные запросы MONETA.Assistant отсылает на указанную страницу магазина Check URL в виде HTTP запроса методом GET или POST. Ответ на HTTP запрос должен быть в формате XML. Во время оплаты одного заказа MONETA.Assistant может несколько раз отсылать проверочные запросы.

В зависимости от ответа, который получит система Moneta.ru, возможны следующие варианты поведения формы оплаты MONETA.Assistant:

- Меняется сумма заказа,
- Нормальное продолжение оплаты заказа,
- Выводится сообщение об ошибке, прием оплаты прекращается.

В запросе на Check URL от системы Moneta.ru передаются следующие GET или POST параметры:

MNT_COMMAND CHECK

Установлен в значение CHECK для проверочных запросов.

MNT_ID

Идентификатор магазина в системе MONETA.RU.

MNT_TRANSACTION_ID

Внутренний идентификатор заказа, однозначно определяющий заказ в магазине.

MNT_OPERATION_ID

Номер операции в системе MONETA.RU. Если операция еще не создана в системе MONETA.RU, то это поле не будет отправлено.

MNT_AMOUNT

Фактическая сумма, полученная на оплату заказа, без учета комиссии за услуги системы MONETA.RU. Если магазин не передал сумму заказа в HTML форме, то данного параметра может не быть. Если в запросе нет данного параметра, то магазин обязан вернуть сумму заказа в ответе на данный запрос, иначе MONETA.Assistant завершит обработку с ошибкой.

MNT_CURRENCY_CODE

ISO код валюты

MNT_SUBSCRIBER_ID

Внутренний идентификатор пользователя.

MNT_TEST_MODE

В тестовом режиме - «1», если в реальном – «0».

MNT_SIGNATURE

Код для идентификации отправителя и проверки целостности данных.

MNT_USER

Номер счета пользователя, если пользователь выбрал оплату с пользовательского счета в системе «MONETA.RU». Данного параметра может не быть.

paymentSystem.unitId

Идентификатор платежной системы. Данного параметра может не быть.

MNT_CORRACCOUNT

Номер счета плательщика. Если оплата производилась с пользовательского счета в системе «MONETA.RU», то совпадает с MNT_USER. Данного параметра может не быть.

MNT_CUSTOM1, MNT_CUSTOM2, MNT_CUSTOM3,

Другие параметры, переданные в запросе на оплату через MONETA.Assistant.

В параметре **MNT_SIGNATURE** передается подпись проверочного запроса, которая формируется следующим образом:

```
MNT_SIGNATURE = MD5( MNT_COMMAND + MNT_ID + MNT_TRANSACTION_ID + MNT_OPERATION_ID
+ MNT_AMOUNT + MNT_CURRENCY_CODE + MNT_SUBSCRIBER_ID + MNT_TEST_MODE
+ КОД ПРОВЕРКИ ЦЕЛОСТНОСТИ ДАННЫХ )
```

Формат ответа должен быть следующим:

```
<?xml version="1.0" encoding="UTF-8"?>
<MNT_RESPONSE>
  <MNT_ID></MNT_ID>
  <MNT_TRANSACTION_ID></MNT_TRANSACTION_ID>
  <MNT_RESULT_CODE></MNT_RESULT_CODE>
  <MNT_DESCRIPTION></MNT_DESCRIPTION>
  <MNT_AMOUNT></MNT_AMOUNT>
  <MNT_SIGNATURE></MNT_SIGNATURE>
  <MNT_ATTRIBUTES>
    <ATTRIBUTE>
      <KEY></KEY>
      <VALUE></VALUE>
    </ATTRIBUTE>
  </MNT_ATTRIBUTES>
</MNT_RESPONSE>
```

Параметры ответа:

MNT_ID

Номер расширенного счёта магазина в системе MONETA.RU.

MNT_TRANSACTION_ID

Внутренний идентификатор заказа.

MNT_RESULT_CODE

Код ответа на запрос.

MNT_DESCRIPTION

Описание состояния заказа.

MNT_AMOUNT

Сумма оплаты.

MNT_SIGNATURE

Код для идентификации отправителя и проверки целостности данных.

MNT_ATTRIBUTES

Содержит произвольные параметры, которые будут сохранены в операции. Необязательный элемент.

ATTRIBUTE

Элемент представляет один произвольный параметр операции.

KEY

Уникальное название параметра операции. Не должно превышать 32 символа.

VALUE

Значение параметра операции.

В ответе на проверочный запрос подпись формируется следующим образом:

```
MNT_SIGNATURE = MD5(MNT_RESULT_CODE + MNT_ID + MNT_TRANSACTION_ID
+ КОД ПРОВЕРКИ ЦЕЛОСТНОСТИ ДАННЫХ)
```

При формировании подписи, **MNT_AMOUNT** должно быть с двумя десятичными знаками, отделенными точкой.

При помощи параметра **MNT_RESULT_CODE** меняется поведение формы оплаты. Наиболее частые варианты ответа такие:

- **MNT_RESULT_CODE** = 100, **MNT_AMOUNT** - установлено в некоторое значение. В этом случае меняется сумма к оплате текущего заказа. Изначально, сумма могла быть не указана вовсе.
- **MNT_RESULT_CODE** = 402, **MNT_AMOUNT** - не принимается во внимание. В таком случае оплата будет продолжена в штатном режиме.
- **MNT_RESULT_CODE** = 500, **MNT_AMOUNT** - не принимается во внимание. Будет выведено сообщение **MNT_DESCRIPTION** в качестве сообщения об ошибке, оплата не будет продолжена.

Следует учитывать, что система Moneta.ru может отправлять несколько проверочных запросов на Check URL. После отправки ответа с **MNT_RESULT_CODE** равным 100, в ответе на следующий запрос **MNT_RESULT_CODE** должен быть 402, либо 500, иначе MONETA.Assistant выведет ошибку и прервет оплату.

Список всех возможных значений **MNT_RESULT_CODE**:

- **100** - ответ содержит сумму заказа для оплаты. Данным кодом следует отвечать, когда в параметрах проверочного запроса не был указан параметр **MNT_AMOUNT**, либо сумма заказа могла измениться.
- **200** - заказ уже был оплачен. Уведомление об оплате магазину доставлено.
- **302** - заказ находится в обработке. Точный статус оплаты заказа определить невозможно. В этом случае в форме оплаты будет выведено **MNT_DESCRIPTION** в качестве сообщения об ошибке, оплата не будет продолжена.
- **402** - заказ создан и готов к оплате. Уведомление об оплате магазину не доставлено.
- **500** - заказ не является актуальным в магазине (например, заказ отменен).

Chapter

3

PayAnyWay выписка счёта

Topics:

- [Создание счёта через Merchant API](#)

Система Moneta.ru позволяет партнерам работать, выставляя их покупателям счёта на оплату. Для работы по такой схеме удобнее открывать отдельный расширенный счёт в системе Moneta.ru, с которого и будут выписываться счета на оплату.

В настройках счёта в личном кабинете Moneta.ru нужно очистить поля Check URL и Pay URL, в поле “Тип интерфейса” нужно указать “Нет”.

Для создания нового счёта на оплату перейдите в раздел “Рабочий кабинет” и далее к опции “Счёт к оплате”.

После создания счёта в системе Moneta.ru будет создана платежная операция, на e-mail покупате будет отправлено письмо со ссылкой на оплату данной операции. Поле “Номер счёта плательщика” - не обязательное. Если оставить его пустым, то плательщику будут доступны все возможные способы оплаты на выбор.

Создание счёта через Merchant API

Выписку счетов на оплату можно автоматизировать. В этом случае, скрипт Pay URL будет полезен.

Для создания счёта на оплату в Merchant API есть метод InvoiceRequest. В качестве аргумента в метод передается объект с типом InvoiceRequestType, который имеет следующие атрибуты:

payer

Номер счета плательщика. Необязательный элемент. Тип: string

payee

Номер счета получателя.

amount

Сумма. Необязательный элемент.

clientTransaction

Внешний номер операции.

description

Описание операции. Необязательный элемент.

mnt_custom1, mnt_custom2, mnt_custom3, ...

Произвольные параметры.

operationInfo

Набор полей, которые необходимо сохранить в качестве атрибутов операции. Значения дат в формате dd.MM.yyyy HH:mm:ss. Для того чтобы провести прямое дебетование, следует передать атрибут с ключом "SUBSCRIBERID" и в значении указать ID подписчика. Прямое дебетование будет проведено, если ID подписчика указано верно, если пользователь разрешил проводить прямое дебетование, если указанная сумма есть на балансе пользователя. В данном случае деньги будут списаны со счета пользователя, в ответе на запрос в поле Статус будет указано INPROGRESS, зачисление денег на счет получателя будет

проведено в асинхронном режиме. Если прямое дебетование не может быть проведено или во время его выполнения произойдет ошибка, то инвойс будет создан и в ответе на запрос в поле Статус будет указано CREATED. USERCONTACT - ссылка на оплату счёта будет отправлена в письме или SMS (можно перечислить получателей через запятую). Необязательный элемент.

Метод InvoiceRequest возвращает ID созданного счёта на оплату. Для оплаты счёта нужно перенаправить пользователя по ссылке:

```
https://www.payanyway.ru/assistant.htm?operationId=ID_созданного_счёта
```

В данную ссылку можно передавать так же параметры MONETA.Assistant, которые рассмотрены в Главе 1.

Можно поместить данную форму в iframe, заменив .htm на .widget, можно разместить ссылку в социальной сети.

Поскольку merchant API подразумевает автоматизацию генерации ссылок на оплату счёта, целесообразно так же настроить скрипт Pay URL, который будет вызван системой Монета.ру после успешного зачисления средств на расширенный счёт интернет-магазина.

Пример кода на SDK php

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->checkMonetaServiceConnection();
$operationId = $monetaSDK->sdkMonetaCreateInvoice($payer = null, $payee, $amount,
    $orderId, $paymentSystem = 'payanyway', $isRegular = false, $additionalData = null);
```

Пример кода на SDK C#

```
MonetaSDK monetaSDK = new MonetaSDK();
MonetaSdkResult result = monetaSDK.sdkMonetaCreateInvoice(string payer, long payee,
    decimal amount, string clientTransaction, bool isRegular);
```

Chapter

4

PayAnyWay с холдированием

Topics:

- [Пример кода на SDK php](#)

У некоторых интернет-магазинов остатки товаров, представленных на сайте могут быть ограниченными. В таком случае, важно принимать оплату с покупателя лишь в том случае, если товар или услуга действительно могут быть ему предоставлены. Если у магазина есть учетная система, которая синхронизирует остатки с интернет-магазином, то целесообразно реализовать Check URL как описано в п.2.7. Если же остатки товара уточняются в ручном режиме, то можно использовать оплату с холдированием. При этом деньги с банковской карты покупателя не списываются сразу же в момент оплаты товара, а блокируются на карте покупателя до принятия решения об отгрузке интернет-магазином, который после проверки остатков принимает решение о списании денег в свою пользу, либо об отмене операции оплаты. При отмене операции, ранее заблокированная на карте покупателя сумма разблокируется. Механизм приема платежей с холдированием выгодно использовать интернет-магазину с плавающими остатками, чтобы не терять на комиссии, которая оплачивается с каждой успешно завершённой операции оплаты.

Последовательность действий, при приеме платежей с холдированием такова:

1. Создать счёт при помощи merchant API (метод InvoiceRequest) на оплату способом “банковские карты” с атрибутом **AUTHORIZEONLY** установленном в 1 (единицу).
2. Переадресовать покупателя на платежную форму MONETA.Assistant с параметром operationId, который был получен на предыдущем шаге, для оплаты счёта. Параметры выставленного счёта, а также ID операции operationId, сохранить в базу данных интернет-магазина.
3. В заранее подготовленном скрипте, принять URL уведомление от системы Moneta.ru “Вызвать URL после авторизации средств”, в этом скрипте по параметру MNT_OPERATION_ID отметить запись в базе данных по строке с operationId как активную (готовую к обработке). По активной операции деньги холдированы (зарезервированы) на банковской карте покупателя.
4. Если интернет-магазин убедился что может поставить нужный покупателю товар, то он (магазин) подтверждает операцию с ID равным operationId на сумму равную или меньше чем указанная ранее (при холдировании). Подтвердить операцию можно через merchant API (метод ConfirmTransactionRequest), или вручную в личном кабинете системы Moneta.ru. После подтверждения операции система Moneta.ru отправит URL уведомление на Pay URL.
5. Если товара нет в наличии, либо существуют иные причины, по которым магазин не может выполнить заказ, магазин Отменяет операцию с ID равным operationId. Отменить операцию можно через merchant API (метод CancelTransactionRequest), или вручную в личном кабинете системы Moneta.ru.

Покупатель имеет право отменить блокировку своих средств на банковской карте, написав заявление на отмену блокировки в банк. Попытка подтвердить не оплаченную ранее операцию приведёт к её заморозке в системе Moneta.ru.

URL уведомление “Вызвать URL после авторизации средств” приходит с параметрами аналогичными тем что описаны выше для Check URL (в разделе 2.7), за исключением **MNT_COMMAND CHECK**. Настройка ссылки на скрипт, принимающий уведомление производится в личном кабинете системы Moneta.ru в пункте меню “Действия при зачислении/списании средств”.

При отмене операции с холдированием, комиссия за проведение операции не взимается. Комиссия возникает лишь при подтверждении операции. Это делает платежи с холдированием более выгодными чем оплаты и возвраты средств для проектов с подвижными или неопределёнными остатками.

Пример кода на SDK php

Создать счёт при помощи merchant API (метод InvoiceRequest) на оплату способом “банковские карты” с атрибутом **AUTHORIZEONLY** установленном в 1 (единицу).

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->checkMonetaServiceConnection();
$additionalData = array('AUTHORIZEONLY' => '1');
$operation_id = $monetaSDK->sdkMonetaCreateInvoice(null, $settings->MNT_ID, $amount,
    $transaction_id, 'payanyway', false, $additionalData);
```

Подтвердить операцию с ID равным operationId (метод ConfirmTransactionRequest):

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->checkMonetaServiceConnection();
try {
    $confirmRequest = new \Moneta\Types\ConfirmTransactionRequest();
    $confirmRequest->transactionId = $operationId;
    $confirmRequest->amount = number_format($amount, 2, '.', '');

    $operationInfo = $monetaSDK->ConfirmTransaction($confirmRequest);
    $result = json_decode(json_encode($operationInfo, true));

    // отметка завершения операции в БД магазина
    // ...
}
catch (\Exception $e) {
    // 'ERROR: Проведение операции не удалось, сделайте это в личном кабинете Moneta.ru'
}
```

Отменить операцию (метод CancelTransactionRequest):

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->checkMonetaServiceConnection();
try {
    $confirmRequest = new \Moneta\Types\CancelTransactionRequest();
    $confirmRequest->transactionId = $operationId;

    $operationInfo = $monetaSDK->CancelTransaction($confirmRequest);
    $result = json_decode(json_encode($operationInfo, true));

    if (is_object($result) && is_object($result->data) && isset($result->data->
        >operationStatus) && $result->data->operationStatus == 'CANCELED') {
        // всё ОК
        // Операция {$model->operation_id} отменена
        // отметить операцию в БД магазина
    }
}
```

```
        // ...
    }
    else {
        // 'ERROR: Отмена не удалась, сделайте отмену операции в личном кабинете
        Moneta.ru'
    }
}
catch (\Exception $e) {
    // непредвиденная ошибка
    $message = 'ERROR: ' . print_r($e, true);
}
```

Chapter

5

PayAnyWay регулярные платежи

Topics:

- [Реализация регулярных платежей на SDK PHP](#)

В проектах, предоставляющих услуги или товары на регулярной основе (ежемесячно, ежегодно, ежедневно) удобным способом списания денег у покупателей является рекуррентный (регулярный) платеж. При этом сумма списания может быть постоянной, либо может изменяться.

Для реализации такой схемы работы в системе Монета.ру есть специальные методы в merchant API. Последовательность действий, выполняемая магазином для реализации данной схемы, будет такой:

1. Создать счёт при помощи merchant API (метод InvoiceRequest) на оплату способом “банковские карты” с атрибутом **PAYMENTTOKEN** в значении 'request' (строка).
2. Перенаправить покупателя на форму MONETA.Assistant с параметром operationId для оплаты счёта. Значение operationId получено на предыдущем шаге. На платежной форме для операций с установленным PAYMENTTOKEN=request будет отображена дополнительная галочка "Запомнить карту", требующая явного согласия пользователя. Если пользователь согласия не даст, то операция будет обработана как обычный платеж и параметр PAYMENTTOKEN будет записан как refused.
3. Получить URL уведомление на Pay URL после успешной оплаты выставленного счёта. Значение параметра **paymenttoken** (если оно не равно refused или request) сохранить для дальнейшего использования. Сохранить дату осуществления платежа. Поставить данного покупателя в расписание на регулярное (ежемесячное, ежедневное, еженедельное и т.д.) списание средств.
4. За несколько дней до регулярного списания отправить покупателю сообщение на e-mail с напоминанием о предстоящем регулярном списании средств и ссылкой, по которой покупатель сможет исключить себя из расписания на регулярное списание средств.
5. При переходе по ссылке на прекращение регулярных списаний, исключить покупателя из расписания на регулярное списание средств.
6. Согласно установленному расписанию, произвести списание средств с покупателя, при помощи ранее полученного значения **paymenttoken**. Для этого нужно выполнить метод PaymentRequest через merchant API, передав параметр **paymenttoken** и его значение как дополнительный атрибут объекта OperationInfo. Сумма операции устанавливается произвольной (по соглашению сторон).
7. Если в ответ на запрос PaymentRequest получена ошибка, нужно исключить покупателя из расписания на регулярное списание средств и заново произвести платеж как было описано в пунктах 1-3.
8. Повторять шаги 4-7 с заданной регулярностью.

Проверять ответ на запрос PaymentRequest необходимо, поскольку покупатель может обратиться в его банк для прекращения регулярных

списаний с его карты. Если это произошло, реализуются действия, описанные в шаге 7.

Для того, чтобы осуществлять рекуррентные платежи, в системе монета необходимо наделить счёт, принимающего платежи, интернет-магазина специальными дополнительными правами.

Рассмотренная схема может быть использована так же для реализации платежей без повторного ввода данных банковской карты покупателя. В этом случае, запрос PaymentRequest с параметром paymenttoken и его значением можно отправить не по расписанию, а по нажатию кнопки “Оплатить” покупателем в любой момент времени. Схема работы при списании средств без повторного ввода данных будет выглядеть так:

1. Создать счёт при помощи merchant API (метод InvoiceRequest) на оплату способом “банковские карты” с атрибутом **PAYMENTTOKEN** в значении 'request' (строка).
2. Перенаправить покупателя на форму MONETA.Assistant с параметром operationId для оплаты счёта. Значение operationId получено на предыдущем шаге.
3. Получить URL уведомление на Pay URL после успешной оплаты выставленного счёта. Значение параметра **paymenttoken** сохранить для дальнейшего использования.
4. Произвести списание средств с покупателя, при помощи ранее полученного значения **paymenttoken**. Для этого нужно выполнить метод PaymentRequest через merchant API, передав параметр **paymenttoken** и его значение как дополнительный атрибут объекта OperationInfo. Сумма операции устанавливается произвольной (по соглашению сторон).
5. Если в ответ на запрос PaymentRequest получена ошибка, нужно заново произвести платеж как было описано в пунктах 1-3.
6. Повторять шаги 4-5 при нажатии покупателем на кнопку “Оплатить” без повторного ввода данных банковской карты.

Реализация регулярных платежей на SDK PHP

Перед началом работы

Для того, чтобы организовать работу сервиса с приемом регулярных платежей, Вам понадобится подключение к источнику данных. Для подключения нужно внести настройки для Вашего проекта в файл data_storage.ini, параметр monetask_storage_type может принимать значение: files или mysql

Если установленное значение - пустая строка, сохранение данных в локальное хранилище не будет производиться.

Генерация формы рекуррентного платежа

Чтобы вывести форму рекуррентного платежа, можно использовать следующий код:

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->processCleanChosenPaymentSystem();
$result = $monetaSDK->showPaymentFrom(null, 4, 'RUB', 'Рекуррентный платеж', false,
    'plastic', true);
echo $result->render;
```

Аргумент метода **\$isRegular** устанавливается в true, поэтому отображенная данным кодом форма будет готова к приему рекуррентных платежей.

Генерация формы рекуррентного платежа

Код обработчика будет таким:

```
$monetaSDK = new Moneta\MonetaSdk();
$result = $monetaSDK->processInputData();
echo $result->render;
```

При поступлении подтверждения об успешном рекуррентном платеже, обработчик выполнит все необходимые для поддержания регулярных платежей действия, а именно - сохранит платежный токен в хранилище данных.

Рассылка уведомлений о предстоящем платеже

Для рассылки уведомлений о предстоящем регулярном платеже со ссылкой на скрипт отмены, добавьте следующий код в расписание для запуска не менее чем каждый день:

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->processRecurrentPaymentNotificationCronTask();
```

Код отправит на e-mail уведомления, сформированные из view RegularNotification.php. Настройки уведомлений находятся в файле regular_payments.ini

Осуществление регулярных платежей

Для осуществления регулярных ежемесячных платежей добавьте следующий код в расписание для запуска не реже чем каждый день:

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->processRecurrentPaymentTransferCronTask();
```

Обработчик отмены регулярного платежа

Ранее рассматривался конфигурационный файл regular_payments.ini, в котором есть параметр regular_payments_cancel_url. По указанному адресу нужно разместить следующий код:

```
$monetaSDK = new Moneta\MonetaSdk();
$result = $monetaSDK->processInputData();
```

Данный код отменит регулярные платежи по ссылке из письма, которое отправлялось ранее.

Ручное проведение регулярного платежа

Имея номер успешно проведенной операции, можно провести повторный (регулярный) платеж в ручном режиме следующим методом:

```
$monetaSDK = new Moneta\MonetaSdk();
$result = $monetaSDK->processPayRecurrent($operationId, $description);
```

где **\$operationId** - идентификатор успешно проведенной ранее операции,

\$description - описание для новой (регулярной) операции.

Chapter

6

PayAnyWay выплаты и списания

Topics:

- [Выплаты через merchant API](#)
- [Выплаты на банковскую карту](#)
- [Списание с карты и выплаты для проектов с PCI DSS](#)

Выплаты через merchant API

Метод **Payment** можно использовать для вывода средств в различные платежные системы. Для выбора платежной системы в методе Payment требуется указать нужный элемент `payee`. Так же, в элементе `operationInfo`, передаются параметры необходимые для вывода средств, которые зависят от выбранной платежной системы.

Для вывода средств со счета Moneta.Ru выполните следующие шаги:

1. Рекомендуется (но необязательно) сначала выполнить запрос `VerifyPayment`, чтобы проверить параметры платежа.
2. Если плательщик получает платежный пароль в SMS, выполните метод `GetAccountPaymentPasswordChallenge` для получения плательщиком платежного пароля.
3. Выполните метод `Payment`. Если операция проведена успешно, то ответ будет содержать атрибут **statusid** со значением **SUCCEED**.

Следующая таблица показывает значения элемента `payee` и возможные значения элемента `operationInfo`, которые нужно передать для вывода средств со счета системы Moneta.Ru в различные платежные системы:

Платежная система	Получатель (payee)	Значения в OperationInfo
Яндекс.Деньги	13	YANDEXACCOUNT Номер кошелька в системе Яндекс.Деньги.
Vkontakte	139	VKONTAKTEID E-mail или имя пользователя ВКонтакте.
WebMoney	2,3,4	WEBMONEYWMID Идентификатор пользователя (WMID). WEBMONEYPURSE Номер кошелька WMR (если <code>payee</code> = 2). Номер кошелька WMZ (если <code>payee</code> = 3). Номер кошелька WME (если <code>payee</code> = 4).
CyberPlat	1144	CYBERPLATPROVIDERID Значение CYBERPLATPROVIDERID . CUSTOMFIELD:100
Банковский перевод	5	WIREUSERNAME Наименование получателя. WIREUSERINN ИНН получателя. WIREPAYMENTPURPOSE Назначение платежа. WIREBANKACCOUNT Номер расчетного счета.

Платежная система	Получатель (payee)	Значения в OperationInfo
Банковские карты (через Банк Русский Стандарт) Только для проектов с PCI DSS	275	WIREBANKBIK БИК банка. WIREBANKNAME Наименование банка. WIREBANKKS Корр. счет банка. WIREKPP КПП. WIREKVK КБК. WIREOKTMO ОКТМО. WIREDOCINDEX УИИ (индекс документа). PAYERCOUNTRY Код страны. Возможные значения: RUS (перевод средств на карту, выпущенную российским банком). PAYEECARDNUMBER Полный номер карты.
Банковские карты (через Альфа-Банк) Только для проектов с PCI DSS	279	PAYEECARDNUMBER Полный номер карты. CARDEXPIRATION Срок действия карты в формате: MM/YYYY
Деньги@Mail.ru	268	CUSTOMFIELD:RECIPIENT E-mail или 16-ти значный номер пользователя в Деньги@Mail.ru CUSTOMFIELD:PAYMENT_PURPOSE Назначение платежа.
QIWI	255	EXTERNALACCOUNTID QIWI account.

Параметры для вывода средств в методе Payment зависят от платежной системы, куда выводятся средства. Чтобы их получить:

1. Сделайте вывод небольшой суммы со счета в Moneta.Ru в нужную платежную систему через интерфейс пользователя для получения номера операции.
2. Используйте метод GetOperationDetailsById для получения деталей операции.
3. В ответе получите элемент OperationInfo. Этот элемент содержит все параметры необходимые для вывода средств через MONETA.MerchantAPI.

Теперь Вы можете отредактировать элемент OperationInfo и использовать результат в методе Payment для вывода средств.

Выплаты на банковскую карту

Если проект имеет сертификат PCI DSS хотя бы в минимальном исполнении, то пользователя можно отправить на страницу <https://www.moneta.ru/secureCardData.htm> для ввода данных банковской карты со следующими параметрами:

```
<form action="https://www.moneta.ru/secureCardData.htm" method="GET">
  <input type="hidden" name="publicId" value="d21ba7f4-1272-43c9-bbc5-e8c7cb75789e"/>
  ><!-- Мой счет -> Безопасность -> Публичный идентификатор -->
  <input type="hidden" name="MNT_ID" value="12345678"/><!-- Номер счета -->
  <input type="hidden" name="MNT_TRANSACTION_ID"
  value="внешний идентификатор операции"/><!-- Идентификатор заказа магазина -->
  <input type="hidden" name="redirectUrl" value="https://сайт.ru/moneta_callback.php"/>
  ><!-- URL, куда будет перенаправлен пользователь -->
  <input type="hidden" name="MNT_SIGNATURE" value="289b0373ec7d25dff0f25534e21adc27"/>
  ><!-- md5(MNT_ID+MNT_TRANSACTION_ID+redirectUrl+Код_проверки_на_счете_MNT_ID) -->
  <input type="hidden" name="secure[PAYEECARDNUMBER]" value="required"/>
  ><!-- Номер карты на форме нужен -->
  <input type="hidden" name="secure[CARDEXPIRATION]" value="required"/><!-- Exp
  date на форме нужен -->
  <input type="hidden" name="secure[CARDCVV2]" value="required"/><!--
  CVV на форме нужен, для вывода на карту параметр передавать не нужно -->
```

```

<input type="hidden" name="formTarget" value="_top"/
><!-- Вернуться после ввода данных на форме в основное окно (_top, _self, _parent) -->
<input type="hidden" name="MNT_DESCRIPTION" value="Описание платежа"/
><!-- Необязательное текстовое описание / заголовок формы -->
</form>

```

То же самое в виде ссылки:

```

https://www.moneta.ru/secureCardData.htm?publicId=d21ba7f4-1272-43c9-bbc5-
e8c7cb75789e&MNT_ID=12345678&MNT_TRANSACTION_ID=%D0%B2%D0%BD%D0%B5%D1%88%D0%BD
%D0%B8%D0%B9+%D0%B8%D0%B4%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE
%D1%80+%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8&redirectUrl=https%3A%2F%2F
%D1%81%D0%B0%D0%B9%D1%82.ru
%2Fmoneta_callback.php&MNT_SIGNATURE=289b0373ec7d25dff0f25534e21adc27&secure
%5BPAYEECARDNUMBER%5D=required&secure%5BCARDEXPIRATION%5D=required&secure
%5BCARDCVV2%5D=required&formTarget=_top&MNT_DESCRIPTION=%D0%9E%D0%BF
%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5+%D0%BF%D0%BB%D0%B0%D1%82%D0%B5%D0%B6%D0%B0

```

Успешный ответ придёт на redirectUrl:

```

redirectUrl?secureToken=K5JelWrviLGSqfff-1406-05f77bc082ce3c0cd242dd70613e7518ab7...
&expirationDate=2015-08-13T09:08:13.381+03:00&MNT_SIGNATURE=bc6e91a851af8458dd8cf...
&MNT_ID=12345678&MNT_TRANSACTION_ID=внешний идентификатор операции&key=value

```

Где:

```
MNT_SIGNATURE = md5(MNT_ID+MNT_TRANSACTION_ID+secureToken+Код_проверки_на_счете_MNT_ID)
```

Ответ с ошибкой будет таким:

```

redirectUrl?faultstring=Пользователь не найден&faultDetail=500.1.14&MNT_ID=32841945
&MNT_TRANSACTION_ID=123&key=value

```

Либо, если мы смогли вычислить MNT_SIGNATURE, то:

```

redirectUrl?faultstring=Подпись на форме оплаты обязательна&faultDetail=500.3.2.56.2
&MNT_SIGNATURE=d390e5dc6ed4ae227e452473519f92&MNT_ID=12345678
&MNT_TRANSACTION_ID=внешний идентификатор операции&key=value

```

Где:

```
MNT_SIGNATURE = md5(MNT_ID+MNT_TRANSACTION_ID+faultDetail+Код_проверки_на_счете_MNT_ID)
```

Далее используем secureToken, полученный в ответе.

Перевод денег на карту с использованием SecureToken (JSON):

```

{"Envelope": {
  "Header": {
    "Security": {
      "UsernameToken": {
        "Username": "USERNAME",
        "Password": "PASSWORD"
      }
    }
  },
  "Body": {
    "PaymentRequest": {
      "version": "VERSION_2",
      "payer": "СЧЕТ ПЛАТЕЛЬЩИКА",
      "payee": "279",
      "amount": СУММА,
      "isPayerAmount": true,
      "clientTransaction": "ВАШ ID ОПЕРАЦИИ",
      "operationInfo": {
        "attribute": [
          {
            "key": "SECURETOKEN",

```

```

    "value":
      "jz8QdLCXQTewXHNS-1075-788270e2fcf170c60136a2a2ee0248ff85f7cac"
    }
  ]
}
}
}}

```

Списание денег с карты с использованием SecureToken (JSON):

```

{"Envelope": {
  "Header": {
    "Security": {
      "UsernameToken": {
        "Username": "USERNAME",
        "Password": "PASSWORD"
      }
    }
  },
  "Body": {
    "PaymentRequest": {
      "version": "VERSION_2",
      "payer": "303",
      "payee": "СЧЕТ ПОЛУЧАТЕЛЯ",
      "amount": СУММА,
      "isPayerAmount": false,
      "clientTransaction": "ВАШ ID ОПЕРАЦИИ",
      "operationInfo": {
        "attribute": [
          {
            "key": "SECURETOKEN",
            "value":
              "jz8QdLCXQTewXHNS-1075-788270e2fcf170c60136a2a2ee0248ff85f7cac"
          }
        ]
      }
    }
  }
}
}}

```

Списание с карты и выплаты для проектов с PCI DSS

Сертификация PCI DSS позволяет проекту осуществлять ввод данных банковских карт на своей стороне.

Временное сохранение данных карты (в мобильном приложении)

Чтобы сохранить данные карты в системе Moneta.ru с возможностью затем осуществить списание или выплату на эту карту, используется метод SecureDataRequest, который работает только для JSON протокола (для SOAP не работает) - запрос должен быть отправлен на url: <https://www.moneta.ru/services/public>

```

{
  "SecureDataRequest": {
    "publicId": "d21ba7f4-1272-43c9-bbc5-e8c7cb75789e",
    "attribute": [
      {
        "key": "CARDNUMBER",
        "value": "1234567890123456"
      },
      {
        "key": "CARDEXPIRATION",
        "value": "01\2020"
      },
      {
        "key": "CARDVV2",

```

```

    "value": "123"
  }
]
}

```

Данные сохраняются на 15 минут.

Ответ на запрос будет таким (JSON):

```

{
  "SecureDataResponse": {
    "expirationDate": "2015-08-10T09:35:02.168+03:00",
    "secureToken": "jz8QdLCXQTewXhNS-1075-788270e2fcf170c60136a2a2ee0248ff85f7cac"
  }
}

```

Запрос можно сделать с помощью ajax с html страницы (пример с использованием jquery), а так же и без помощи ajax. За дополнительной технической информацией по этому поводу обратитесь в техническую поддержку.

Имея secureToken можно осуществить списание денег с карты и перевод денег на карту. За дополнительной технической информацией по этому поводу обратитесь в техническую поддержку.

Списание с банковской карты через merchant API

Запрос методом PaymentRequest можно отправить с данными банковской карты. Если карта не имеет защиты 3DS, то платеж будет сразу же проведён. Если же 3DS у данной карты включен, то в ответе на запрос PaymentRequest придёт ссылка на страницу ввода кода 3DS. За дополнительной технической информацией по этому поводу обратитесь в техническую поддержку.

Chapter 7

PayAnyWay площадкам

Банковским организациям, желающим предоставлять их клиентам услуги эквайринга, в том числе в концепции White label, можно использовать функционал PayAnyWay для площадок. В этом случае станет возможным открывать аналитические счета в системе Moneta.ru для каждого клиента организации (банка), а так же принимать online платежи на такие счета. За дополнительной технической информацией по этому поводу обратитесь в техническую поддержку.

Chapter

8

Полезные ссылки

Topics:

- [SDK на различных языках](#)
- [Документация по MONETA.Assistant](#)
- [Описание методов MONETA.MerchantAPI](#)

SDK на различных языках

- [C#](#)
- [PHP](#)
- [Python3](#)
- [Ruby](#)
- [Java](#)

Документация по MONETA.Assistant

- <https://www.moneta.ru/doc/MONETA.Assistant.ru.pdf>

Описание методов MONETA.MerchantAPI

- <https://www.moneta.ru/doc/MONETA.MerchantAPI.v2.ru.pdf>