Описание SDK PHP

Содержание

Введение

Документ описывает Software Development Kit (SDK) на PHP.

Документ адресован разработчикам, имеющим базовые знания языка PHP, подключающим оплату через систему «MONETA.RU».

Глава 1

Скачивание и установка

Установку SDK PHP можно осуществить через Composer, либо скачав файлы SDK из репозитория.

Для проектов, использующих Composer

В проектах, где установлен и используется Composer, выполните команду:

```
composer require integrationmonitoring/moneta-php-sdk
```

Либо, в файл composer.json добавьте строки:

```
"integrationmonitoring/moneta-php-sdk": "dev-master",
"moneta/webservice": "dev-master"
```

А затем выполните команду:

```
composer update
```

Для обновления SDK PHP так же используйте команду:

```
composer update
```

Для того, чтобы использовать SDK в коде своего проекта, добавьте в код строчку:

```
use Moneta;
```

SDK готово к использованию, можно создать объект:

```
$monetaSDK = new Moneta\MonetaSdk();
```

Теперь можно использовать методы SDK Moneta.

Для проектов без Composer

Если в проекте не используется Composer, то скачайте “lib” версию SDK из репозитория или архива: *https://github.com/integrationmonitoring/moneta-sdk-lib*

Скопируйте содержимое в удобное для проекта место.

Чтобы использовать SDK в коде своего проекта, добавьте в код строчку:

```
include_once(__DIR__ . "/moneta_sdk_lib/autoload.php");
```

Где “moneta_sdk_lib” - путь к “lib” версии SDK. После этого можно будет создать объект:

```
$monetaSDK = new \Moneta\MonetaSdk();
```

Далее можно использовать методы SDK Moneta.

Структура каталогов SDK PHP

SDK включает в себя следующие каталоги:

Каталог	Описание
config	содержит примеры файлов настроек. Скопируйте имеющиеся в данном каталоге файлы с расширением “.ini.example” в ту же папку с расширениями “.ini”, либо поменяйте расширение имеющихся файлов.
events	содержит php файлы, которые исполняются при возникновении различных событий. В коде этих файлов доступен массив $data, в который заранее переданы необходимые для обработки события данные. Так же доступны такие переменные как $_GET, $_POST и прочие серверные переменные.
src	папка с исходными кодами SDK.
view	содержит шаблоны для представления информации и форм.
logs	папка для сохранения логов при работе в режиме отладки.
data	папка для сохранения данных.

Глава 2

Использование файлов настроек

Папка **config** содержит файлы настроек. При инициализации объекта **MonetaSdk** в первую очередь считываются настройки, которые обеспечат работу SDK в дальнейшем, при вызове методов SDK.

Объект **MonetaSdk** имеет конструктор, позволяющий определить путь к файлам настроек следующим образом:

```
$monetaSDK = new Moneta\MonetaSdk($configPath);
```

Здесь $configPath - это строка - путь к папке с **config** файлами, вместо предустановленного. Если параметр не указан, либо пустой, для подключения файлов настроек будет использован предустановленный путь, упомянутый в п.2. настоящего руководства.

basic_settings.ini

Данный файл должен быть изменен для конкретного интернет-магазина (или другого ресурса) и содержит следующие настройки:

Переменная	Значение
monetasdk_connection_type	тип соединения SDK с сервером системы moneta.ru. Может содержать одно из 2-х значений: "soap" или "json". Нужное значение выбирается исходя из того, установлена библиотекак php soap или php curl соответственно.
monetasdk_use_x509	может принимать значение “0” или “1”. При влючении в “1” авторизация в платежной системе moneta.ru будет осуществляться посредством открытого ключа (сертификата).
monetasdk_x509_pem_file	путь к файлу certificate.pem, к сертификату для входа в аккаунт moneta.ru
monetasdk_demo_mode	принимает одно из значений “1” или “0”. При включении SDK будет работать с demo сервером moneta.ru, если опция выключена - с продакшен сервером.
monetasdk_test_mode	“1” или “0”. Включение тестового режима взаимодействия с moneta.ru
monetasdk_debug_mode	“1” или “0”. Включает отладочное логгирование.
monetasdk_account_id	номер расширенного счета для приема платежей интернет-магазина (или другого ресурса).

Переменная	Значение
monetasdk_account_code	код проверки целостности данных. Должен быть идентичным тому, что установлен в настройках используемого расширенного счета, указанного в предыдущем параметре. Данный код используется для проверки подлинности запросов и для подписи запросов если он (код) установлен. Если в настройках счёта выключена проверка подписи запросов, то **monetasdk_account_code** надо оставить пустым, т.е. monetasdk_account_code = “”
monetasdk_account_pay_password_enrypted	закриптованный платежный пароль.
monetasdk_account_username	логин аккаунта moneta.ru, имеющего доступ к расширенному счету интернет-магазина. Для данного логина должна быть выключена двухфакторная аутентификация. В ряде случаев целесообразно иметь отдельную пару логин/пароль, которая будет использоваться для SDK.
monetasdk_account_password	пароль аккаунта moneta.ru, имеющего доступ к расширенному счету интернет-магазина.
monetasdk_prototype_user_unit_id	ID юнита в системе монета.ру для использования в качестве прототипа при создании нового профайла пользователя.
monetasdk_event_files_path	путь к папке с файлами событий. Разработчик, использующий SDK имеет возможность указать любой, нужный ему путь, вместо предустановленного к папке events, которая упомянута в п.2 настоящего руководства.
monetasdk_view_files_path	путь к папке с файлами представлений. Любой, нужный путь вместо предустановленного к папке view.

Если указать параметр **monetasdk_x509_pem_file**, а параметр **monetasdk_use_x509** установить в единицу, то авторизация в систему монета будет осуществляться при помощи сертификата. В этом случае можно оставить пустыми параметры **monetasdk_account_username** и **monetasdk_account_password**. Авторизация в систему при помощи сертификата необходима для проектов с виртуальными счетами и балансом пользователей.

data_settings.ini

Файл содержит настройки для хранилища данных. Некоторые операции SDK используют внешнее хранилище данных. Поэтому каждый конкретный интернет-магазин должен изменить данный файл, в соответствии с используемым в нём хранилищем данных:

Переменная	Значение
monetasdk_storage_type	содержит тип хранилища и принимает одно из значений "files", “mysql”
monetasdk_storage_files_path	путь до файлов, в которые будут сохранять данные, если путь не указан, то будет создана папка files, куда будут, при необходимости, сохраняться данные.

Переменная	Значение
monetasdk_storage_mysql_host **monetasdk_storage_mysql_username** **monetasdk_storage_mysql_password** **monetasdk_storage_mysql_port** **monetasdk_storage_mysql_database**	параметры доступа к хранилищу mysql

success_fail_urls.ini

Файл содержит ссылки на страницы интернет-магазина, куда осуществляет перевод пользователя после обработки формы оплаты заказа:

Переменная	Значение
monetasdk_success_url	страница магазина с текстом об успешном приеме платежа.
monetasdk_fail_url	страница с текстом о том, что платеж не был принят.
monetasdk_inprogress_url	страница о том, что проведение платежа находится в обработке.
monetasdk_return_url	страница после возврата из формы оплаты заказа.
monetasdk_iframe_target	одно из значений "_parent", “_blank” и т.д. для настройки окна возврата при использовании страницы оплаты в iframe (т.е. при использовании виджета оплаты заказа вместо страницы оплаты заказа).

payment_urls.ini

Данный файл содержит ссылки к ресурсам системы moneta.ru, которые обеспечивают взаимосвязь с SDK. Эти настройки не зависят от настроек канкретного магазина, но гипотетически могут быть изменены в будущем системой moneta.ru. Поэтому данный файл следует обновить при скачивании новой версии SDK, наряду с остальными файлами и исходным кодом SDK. Файл содержит настройки:

Переменная	Значение
monetasdk_demo_url	ссылка на demo сервер системы moneta.ru
monetasdk_production_url	ссылка на продакшен сервер moneta.ru
monetasdk_soap_link	относительная ссылка на wsdl файл
monetasdk_json_link	относительная ссылка на сервис, принимающий json запросы
monetasdk_assistant_link	относительная ссылка на визуальное представление сервиса приема платежей
monetasdk_x509_port	порт для авторизации с помощью сертификата
monetasdk_x509_soap_link	относительная ссылка на wsdl для доступа с помощью сертификата
monetasdk_x509_json_link	относительная ссылка на сервис, принимающий json запросы при авторизации с помощью сертификата

payment_systems.ini

Файл настроек платежных систем. Содержит идентификаторы и другие параметры платежных систем, на которые интернет-магазин может принимать платежи от пользователей с помощью системы moneta.ru. В новых версиях SDK будет актуальный на текущий момент файл настроек платежных систем. Пример настроек одной из систем:

```
[monetasdk_paysys_payanyway]
group = "electronic"
title = "PayAnyWay"
accountId = "0"
unitId = "0"
createInvoice = "0"
```

additional_field_names.ini

Дополнительные поля для платежных систем. Файл содержит названия полей, которые будут дополнительно запрошены SDK при использовании некоторых способов пополнения, таких, как например “Почта России”. Фразы (названия полей) каждый интернет-магазин может изменить на усмотрение. В новых версиях SDK будет выходить файл с актуальным состоянием дополнительных полей платежных систем.

error_texts.ini

Файл для хранения человеко-читаемых представлений об ошибках. Полный перечень кодов ошибок, которые могут возникать при работе с API системы moneta.ru можно найди в документации к merchant API.

dynamic_change.ini

У разработчика может появиться необходимость изменить какую-либо из настроек в процессе выполнения кода SDK. Это можно осуществить следующим образом:

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->setSettingValue('monetasdk_account_id', $mnt_id);
```

где **$mnt_id** - номер расширенного счёта интернет-магазина для приёма платежей.
'**Monetasdk_account_id**' - идентификатор динамически изменяемого параметра.

Глава 3

Основные методы SDK для интернет-магазина

Все методы SDK можно разделить на несколько групп.

Принадлежность метода той или иной группе определяется префиксом в имени этого метода. На текущий момент в SDK всего 3 группы методов:

process

методы обработки входящих данных, в том числе из потоков $_GET, $_POST и т.д.

show

методы для вывода форм, информации и т.п.

moneta

методы merchant API системы moneta.ru

Схема работы с методами

Схема работы с методами SDK во всех случаях одинакова. После вызова метода, разработчик получает объект результат, у которого есть 3 доступных для прямого чтения свойства:

Свойство	Описание
error	флаг ошибки. Установлен в true если в процессе выполнения SDK метода произошла ошибка и false если выполнение прошло без ошибок.
data	содержит сырые данные, которые являются результатом выполнения SDK метода. Если в процессе выполнения метода произошла ошибка, то свойство data является объектом с доступными для чтения свойствами code и message, содержащими соответственно код ошибки и его строковое представление. Если в процессе выполнения метода ошибок не было, то data является массивом данных. Этот массив будет так же доступен разработчику во внешних представлениях, размещающихся в папке view.
render	содержит отрендеренное представление результатов работы метода, либо представление ошибки, если в процессе выполнения метода была ошибка. Для рендеринга сообщения об ошибке используется, к примеру, шаблон ErrorMessage.php

show Choose Payment System Form

Метод для отрисовки формы выбора платежной системы:

```
$monetaSDK->showChoosePaymentSystemForm($paySystemTypes);
```

Где **$paySystemTypes** - массив строковых представлений типов выводимых для выбора платежных систем.

Возможные значения обусловлены свойством group из файла настроек платежных систем payment_systems.ini (см. п. 3.5.). На текущий момент это: "electronic", "bank", "card", "terminal".

Результат для вывода представления успешно выполненного метода находится в доступном для чтения свойстве render результата:

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

show Payment From

Метод для отрисовки кнопки “Оплатить”. Ниже приведен пример, который выведет кнопку “Оплатить” и обнулит выбранный ранее способ оплаты:

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->processCleanChoosenPaymentSystem();
$result = $monetaSDK->showPaymentFrom($orderId, $amount, $currency = 'RUB',
 $description = null, $isIframe = false, $paymentSystem = null, $isRegular = false,
 $additionalData = null, $method = 'POST');
echo $result->render;
```

Где **$orderId** - номер операции, например 370429, **$amount** - сумма операции, **$currency** - валюта операции, например RUB, **$description** - описание операции, **$isIframe** - выводить в виде виджета в iframe, **$paymentSystem** - способ пополнения (оплаты), **$isRegular** - будет ли платеж рекурентным, **$additionalData** - дополнительные атрибуты, **$method** - метод отправки формы.

Если вместо номера операции передать null, то система moneta.ru назначит уникальный идентификатор для новой операции. Метод processCleanChoosenPaymentSystem вызывается перед showPaymentFrom в связи с тем, что выбранный способ оплаты сохраняется в куку браузера. Обнуление сработает при следующей загрузке страницы, а значит showPaymentFrom увидит в куке установленное ранее с помощью showChoosePaymentSystemForm значение. Чтобы получить виджет платежной формы (в iframe), пятым параметром следует передать true.

process Clean Choosen Payment System

Очистка выбранного ранее функцией showChoosePaymentSystemForm способа оплаты.

Данный метод можно вызывать только перед выводом какой-либо информации, поскольку метод производит запись в куку.

process Input Data

Метод для обеспечения реакции на внешний запрос.

Примером внешнего запроса может быть запрос Pay Url от системы moneta.ru. Таким образом, чтобы организовать страницу, подключенную к “Pay URL” системы moneta.ru, а так же к другим подобным call back запросам от moneta.ru, достаточно создать страницу со следующим кодом:

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

Чтобы обеспечить смену статуса заказа (например на статус “Оплачено”), нужно поместить код, меняющий состояние заказа в php файл события *MonetaPaySuccess.php*, находящийся в папке **events** или другой (устанавливается в config файле **basic_settings.ini**).

При каждом запросе, приходящем от moneta.ru будет отрабатываться событие *MonetaSendCallBack.php*, приходящие данные будут доступны в массивах $_GET, $_POST и $data.

show Create User Form

Метод выводит форму для добавления нового пользователя и счета.

При добавлении пользователя срабатывает событие CreateUserResult.php, в которое передается unitId, а также исходные данные, по которым создается новый пользователь: firstName, lastName, email, gender.

При добавлении нового счёта для нового пользователя срабатывает событие CreateAccountResult, в которое передаются следующие данные: unitId, accountId, paymentPassword, alias.

Код для для вывода формы:

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

В $result->data будут собраны все вышеперечисленные данные (так же переданные в события).

show Account History Form

Метод показывает форму для получения истории операций по счёту.

Для представления используется view AccountHistoryForm.php

После сабмита данные передаются в то же представляение для отображения. Данные из метода так же приходят в $result->data.

```
$monetaSDK = new Moneta\MonetaSdk();
$result = $monetaSDK->showAccountHistoryForm(78008544);
echo $result->render;
```

show Operation Info

Метод отображает данные по заданной операции.

Для представления используется view OperationInfo.php

“Сырые” данные приходят в $result->data.

```
$monetaSDK = new Moneta\MonetaSdk();
$result = $monetaSDK->showOperationInfo(3645590);
echo $result->render;
```

Глава 4

Типовые решения и рецепты

Для некоторых типовых проектов можно рекомендовать следующие рецепты.

Для интернет-магазина

Для типового интернет-магазина схема работы может быть такой:

1. Создается заказ, которому присваивается уникальный идентификатор, чаще всего ID.
2. С помощью SDK, покупателю выводится форма выбора способа оплаты (данный шаг можно пропустить):

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

3. Выводится кнопка “Оплатить” в корзине закказа:

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->processCleanChoosenPaymentSystem();
$result = $monetaSDK->showPaymentFrom(ID, AMOUNT, 'RUB', 'Оплата заказа N ID');
echo $result->render;
```

Где **ID** - уникальный идентификатор заказа интернет-магазина, **AMOUNT** - сумма заказа с разделителем десятых через точку, **'RUB'** - строка - валюта проводимой операции.

4. Для того, чтобы статус заказа в интернет-магазине поменялся на “Оплачено”, в настройках расширенного счета интернет-магазина следует установить “Pay URL”. По указанному адресу следует разместить следующий код:

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

Код, меняющий статус заказа следует поместить в файл события **MonetaPaySuccess.php**, размещенный в папке **events** или другой, в зависимости от настроек. **ID** успешно проведенной операции, а так же другие данные заказа будут доступны в переменных $_POST, $_GET (в зависимости от настроек расширенного счета) и в массиве **$data.**

Для сервиса с виртуальными счетами и балансом пользователей

Сервис с виртуальными счетами пользователей более сложная и ответственная задача.

Для повышения безопасности такого сервиса рекомендуем не хранить состояние счетов пользователей. В этом нет необходимости, поскольку можно получить состояние любого счета одним из методов SDK в любой момент, в том числе перед проведением платежной операции.

Рекомендуется так же организовать авторизацию при помощи сертификата, т.к. это более безопасно чем использование логина и пароля, которые сохраняются в конфигурационном файле **basic_settings.ini**

Формирование сертификата для авторизации

1. Установите OpenSSL
2. Создайте запрос на сертификат следующей командой:

```
openssl req -new -newkey rsa:2048 -sha512 -nodes -out request.txt -keyout
 private.key
```

   Пароль сертификата оставляйте пустым. OpenSSL создаст запрос на сертификат в файле request.txt. Используйте этот запрос на сертификат для получения клиентского сертификата в Moneta.Ru.
3. Получите клиентский сертификат в moneta.ru. Для этого войдите в личный кабинет moneta.ru под логином/паролем, которые планируется использовать для соединение из кода Вашего сайта (пара логин/пароль без двухфакторной авторизации). Перейтите в пункт меню “Безопасность”. Затем перейдите в “Мой счет” > “Безопасность”. Далее нажмите “Добавить Сертификат”. Скопируйте запрос на сертификат из файла request.txt, который Вы создали ранее в поле Запрос на сертификат и нажмите “Продолжить”. **Важно:** Скопируйте все содержимое файла, включая строки BEGIN CERTIFICATE REQUEST и END CERTIFICATE REQUEST. Проверьте информацию указанную в сертификате и нажмите “Сохранить”. Moneta.Ru создаст клиентский сертификат. Нажмите ссылку “Скачайте сертификат”.
4. В скаченный сертификат добавьте содержимое сгенерированного ранее файла private.key вместе со строками BEGIN PRIVATE KEY и END PRIVATE KEY.
5. Полученный результат сохраните и положите на сервер в папку с ограниченным снаружи доступом, например **/usr/local/etc**
6. В конфиг файле basic_settings.ini в строку monetasdk_x509_pem_file полное имя файла сертификата с полным путем к нему. Значение monetasdk_use_x509 установите в единицу (“1”), monetasdk_account_username и monetasdk_account_password оставьте пустыми.

Перед началом работы

Для того, чтобы сервис с виртуальными счетами и балансом пользователей начал работать, необходимо согласование с сервисом moneta.ru, а так же применение индивидуальных настроек на стороне сервиса moneta.ru.

Следующим шагом необходимо придумать строку, которая в дальнейшем будет использоваться для шифрования / дешифрования платежного пароля основного счёта и / или счетов клиентов.

1. Придумайте строку с секретной фразой и запишите её в профайл основного счёта при помощи метода SDK:

```
$monetaSDK = new Moneta\MonetaSdk();
$result = $monetaSDK->processPutSecretToAccountProfile('test');
```

   где **'test'** - секретная фраза.
2. Получите зашифрованный платежный пароль основного счета и запишите его в конфигурационный файл. Получить зашифрованное значение платежного пароля можно при помощи метода SDK:

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

   где $payPassword - платежный пароль в явном виде.

   Зашифрованный пароль придет в переменной:

```
$this->data['result']
```

   Это значение запишите в параметр **monetasdk_account_pay_password_enrypted** в файл **basic_settings.ini** Ваших настроек **config**.

Используемые методы

Для типового решения с использованием виртуальных счетов пользователей могут быть использованы следующие методы.

Создание нового пользователя

При регистрации нового пользователя на сайте, который является сервисом с виртуальными счетами и балансом, необходимо при помощи SDK создать профиль нового пользователя и его нового счета в системе moneta.ru:

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

В результате работы данного метода будут отработаны два события: CreateUserResult.php и CreateAccountResult.php, в которые будут переданы все данные нового пользователя и его счёта, в том числе unitId нового пользователя, accountId нового счета пользователя, платежный пароль нового счёта в открытом виде, платежный пароль в зашифрованном виде.

Отображение остатков счёта пользователя

Пользователь должен иметь возможность видеть остатки его счета. Чтобы отобразить их, следует воспользоваться методом SDK:

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

где **$accountId** - номер счёта нового пользователя.

Пополнение баланса пользователя

Пользователь должен иметь возможность пополнить баланс его счета. Чтобы это осуществить нужно использовать метод SDK:

```
$monetaSDK = new Moneta\MonetaSdk();
$monetaSDK->processCleanChoosenPaymentSystem();
$result = $monetaSDK->showPaymentFrom($accountId, $amount, 'RUB', 'Пополнение счета');
echo $result->render;
```

где **$accountId** - номер счёта пользователя, **$amount** - сумма, на которую будет произведено пополнение счёта.

Для того, чтобы списать со счёта пользователя средства в пользу магазина, следует воспользоваться методом SDK:

```
$monetaSDK = new Moneta\MonetaSdk();
$result = $monetaSDK->sdkMonetaTransfer($fromAccountId, $fromAccountPaymentPassword,
 $toAccountId, $amount, $description = '');
```

Если в качестве платежного пароля передать NULL, то будет использован платежный пароль основного счёта.

Просмотр истории операций пользователя

Воспользуйтесь методом showAccountHistoryForm, который описан выше. В качестве аргумента метода передайте номер счета пользователя.

Для приема регулярных платежей

Сервис, принимающий регулярные платежи имеет свои особенности и набор методов

Перед началом работы

Для того, что бы 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

monetasdk_storage_type может принимать следующие значения:

- **files**
- **mysql**

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

Генерация формы рекурентного платежа

Чтобы вывести форму рекурентного платежа, можно использовать следующий код:

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

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

Обработчик Pay Url

Обработчик ответа от системы moneta.ru о проведении платежей будет выглядеть так же как и в других случаях:

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

При поступлении подтверждения об успешном рекурентном платеже, обработчик выполнит все необходимые для поддержания регулярных платежей действия.

Рассылка уведомлений о предстоящем платеже

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

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

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 запука не менее чем каждый день:

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

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 проведённой операции методом, описанном в п. 6.3.2., 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.

Глава 5

Пробитие чека 54-ФЗ

Пробитие чека 54-ФЗ

В новой ревизии SDK добавляются методы для пробития чека по 54-ФЗ.

Конфигурация (настройки) производятся с помощью файла **kassa_settings.ini**.

Параметр **monetasdk_kassa_type** может принимать одно из трех значений:

- **payanyway** - если вы используете сервис **kassa.payanyway.ru** для пробития чеков и хранения настроек кассы
- **module** - для отправки команд напрямую в API Модуль.кассы из SDK
- **atolonline** - для АТОЛ онлайн
- **buhsoft** - Бухсофт
- **dreamkass** - Дримкас
- **iretail** - i-Retail
- **komtet** - КОМТЕТ Касса
- **orangedata** - Orange data
- **starrys** - ЧЕК онлайн (СТАРРУС)

Пример кода, использующий SDK для пробития чека:

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

$docId = time();
$currentDate = date(DATE_ATOM);
$clientEmail = 'test@test.ru';
$responseURL = null;

$document = array(
    'id' => $docId,
    'checkoutDateTime' => $currentDate,
    'docNum' => $docId,
    'docType' => 'SALE', // или 'SALE_RETURN' для чека на возврат
    'email' => $clientEmail,
);

if ($responseURL) {
    $document['responseURL'] = $responseURL;
}

// товары
$inventoryPositions = array(
    array('name' => 'tovar 1', 'price' => 1, 'quantity' => 2, 'vatTag' =>
 $kassa::VATNOVAT),
    array('name' => 'tovar 2', 'price' => 3, 'quantity' => 1, 'vatTag' =>
 $kassa::VATNOVAT),
);
$document['inventPositions'] = $inventoryPositions;
```

```
// платеж
$moneyPositions = array(
    array('paymentType' => 'CARD', 'sum' => 5),
);
$document['moneyPositions'] = $moneyPositions;

$document = json_encode($document, JSON_PRETTY_PRINT | JSON_UNESCAPED_UNICODE);
$result = $kassa->sendDocument($document);

// ответ:
// Array ( [status] => QUEUED [fnState] => ASSOCIATED [fiscalInfo] => )
// QUEUED - значит встало в очередь
```

В зависимости от значения установленного параметра **monetasdk_kassa_type** чек будет передан в Модуль.кассу, в АТОЛ онлайн, либо в интеграционный сервис **kassa.payanyway.ru**

Глава
6

Список методов

Методы SDK

showChoosePaymentSystemForm($redirectUrl = null, $paySystemTypes = array())

showPaymentFrom($orderId, $amount, $currency = 'RUB', $description = null, $isIframe = false, $paymentSystem = null, $isRegular = false, $additionalData = null, $method = 'POST')

showAccountBalance($accountId)

showOperationInfo($operationId)

showCreateUserForm($redirectUrl = null)

showAccountHistoryForm($accountId)

processCleanChoosenPaymentSystem()

processPutSecretToAccountProfile($secret)

processEncryptPayPassword($payPassword)

processDecryptPayPassword($payPassword)

processInputData($definedEventType = null)

processRecurentPaymentNotificationCronTask()

processRecurentPaymentTransferCronTask()

processPayRecurrent($operationId, $description = null)

Chapter 7

Полезные ссылки

Topics:

SDK на различных языках

- *C#*
- *PHP*
- *Python3*
- *Ruby*
- *Java*

Документация по MONETA.Assistant

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

Описание методов Merchant API

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