

API платежной формы

Общее описание

Взаимодействие с платежным агрегатором Onpay возможно без перенаправления плательщика на платежную форму.

API поддерживает 2 запроса:

1) **Запрос на получение информации о платежной форме** (курсы обмена, комиссии, списки платежных интерфейсов, получение локализации ключей...) GET запрос отправляется в формате JSON на url идентичный адресу платежной формы -

https://secure.onpay.ru/pay/merchant_login

2) **Запрос на создание платежного ордера**. Ответом является инструкция для дальнейшего перенаправления плательщика для оплаты или показа инструкции. POST запрос отправляется в формате json на url - <https://secure.onpay.ru/pay>

Шаг получения информации является НЕобязательным для создания платежного ордера. Но он необходим для получения свежих данных, в том числе о курсах обмена. Сайт магазина может делать его раз в сутки, либо после получения ошибки об изменении курса обмена, либо при изменении настроек платежных систем в своем кабинете.

Получение информации о платежной форме

Для получения информации следует отправить GET запрос в формате JSON на URL «https://secure.onpay.ru/pay/merchant_login», где merchant_login - идентификатор сайта. В ответ будет получен JSON следующего формата: { «paysystem_interfaces»: {...}, «paysystems»: {...}, «additional_params»: {...}, «phone_codes»: {...}, «locales»: {...} }

в случае ошибки вида: { «errors»: {«key»: [«text1», «text2», ...]}, }

где key - ключ ошибки, значение которого массив текстовых описаний ошибок (может быть одно или несколько).

Пример ответа с ошибкой: { «errors»: { «recipient»: [«Для использования API платежной формы необходимо обратиться в техническую поддержку Onpay для активации данного функционала для магазина»] } }

Пример полного успешного ответа см. в разделе «Пример расчета параметров»

Описание параметров

paysystem_interfaces

Значением является hash, ключами которого являются названия доступных и включенных интерфейсов оплаты (данные значения для использования в запросе на создание ордера как «interface_ticker») Значениями ключей-интерфейсов является также хеш, содержащий 2 ключа: «paysystem» - для определения комиссий и курсов обмена, а также списка дополнительных полей, «logo» - url логотипа для интерфейса.

Пример: «paysystem_interfaces»: { «SBR»: { «paysystem»: «BBR»,
«logo»: «/assets/payment_systems/logo/SBR.png» } }

paysystems

Значением является hash, содержащий следующие ключи: «min» - сумма минимального платежа через данный способ в единицах данной системы «max» - сумма максимального платежа через данный способ в единицах данной системы «currency_code» - 3-х символьный код валюты (для отображения плательщику, если необходимо) «convert_to» - платежная система, в которую будет осуществлена автоконвертация, если не указана валюта в которую должен быть сконвертирован платеж. НЕ во все системы можно зачислять средства, поэтому в качестве значения поля «ticker» в запросе на создание ордера необходимо использовать только значения из данного поля, а не любую из платежных систем. «commissions» - комиссии платежной системы, содержит 3 параметра: pip - процентная комиссия, pif - фиксированная комиссия, mci - минимальная комиссия. «exchange_rates» - курсы обмена, представленные в виде хеша вида: {«USD»:0.01597,«RUR»:1.0}. Такая запись означает, что одна единица в данной платежной системе может быть обменяна на 0.01597 единиц в валюте USD или на 1.0 в валюте RUR.

Пример: «BBR»: { «min»:100.0, «max»:150000.0, «currency_code»:«RUB»,
«convert_to»:«RUR», «commissions»: { «pip»:1.0, «pif»:0.0, «mci»:0.0 },
«exchange_rates»: { «USD»:0.01597, «RUR»:1.0, } }

additional_params

Значением является hash, ключи которого - именование платежной системы

Пример:

«BBR»: { «data»: [{ «name»: «first_name», «regex»: «^[A-ZА-Яа-яёЁ\\s']{2,30}\$», «label»: «pay_form_add_p_label.first_name», «message»: «pay_form_add_p_message.first_name» }, { «name»: «middle_name», «regex»: «^[A-ZА-Яа-яёЁ\\s']{2,30}\$», «label»: «pay_form_add_p_label.middle_name», «message»: «pay_form_add_p_message.middle_name» }, { «name»: «last_name», «regex»: «^[A-ZА-Яа-яёЁ\\s']{2,30}\$», «label»: «pay_form_add_p_label.last_name», «message»: «pay_form_add_p_message.last_name» }, { «name»: «address», «regex»: «^[a-zA-Z\\s'0-9,\\-А-Яа-яёЁ,\\d\\/\\\\\\\\.\\\\,\\»]{4,250}\$», «label»: «pay_form_add_p_label.address», «message»: «pay_form_add_p_message.address» }], «SBR»: [] }

Значением может быть null или hash. Если null - значит для данной платежной системы нет дополнительных полей. Если hash - значит в запросе на создание платежного ордера должны присутствовать перечисленные поля.

Список полей находится по ключу «data», кроме данного ключа могут присутствовать другие, а именно именованная интерфейсов оплаты.

Пример выше расшифровывается как: Для платежной системы BBR 4 дополнительных поля: `first_name`, `middle_name`, `last_name`, `address`. Следовательно для всех интерфейсов данной системы их необходимо передавать, за исключением платежного интерфейса «SBR» (передан отдельным ключом).

Следует отметить, что у ключа интерфейса может быть свой вложенный ключ «data», в этом случае для данного интерфейса используются поля из вложенного hash-а (запись в том же формате)

Значение поля «data» - массив hash-ей, каждый из которых описывает одно дополнительное поле.

hash описания может содержать следующие ключи: `*name*` - название параметра, именно оно должно быть использовано при запросе на создание ордера. `*regex*` - regex для первичной валидации значения на стороне сайта магазина. `*label*` - ключ локализации (перевод брать из файлов локализации, см. в разделе «locales»), который должен быть использован для отображения плательщику названия поля. `*message*` - ключ локализации (перевод брать из файлов локализации, см. в разделе «locales»), который должен быть использован для отображения ошибки плательщику при нарушении regex-а. `*countries*` - данный ключ, присутствует только у полей с `name` равным «user_phone», значение данного поля - список стран, с телефонов которых можно платить данным способом. (Полный список стран см. в разделе «phone_codes»). При выборе плательщиком данного способа список возможных кодов стран должен быть ограничен. `*type*` - данный ключ указывает тип дополнительного поля (если ключ отсутствует, то «input type='text'»). может принимать значение «checkbox», в данном случае поле regex будет равно «true». Это означает что валидное значение параметра в запросе на создание ордера будет строка «true», а плательщику в форме должен отображаться checkbox. `*hint*` - содержит ключ локализации подсказки к полю. Перевод должен быть отображен плательщику рядом с полем. (может быть пустым, в данном случае отображать не следует)

phone_codes

Значением является hash вида: { «ru»:»+7«, «ua»:»+38«, «be»:»+375«, «il»:»+972«, «ab»:»+7840«, «kz»:»+7«, «kg»:»+996«, «cn»:»+86«, «my»:»+60«, «am»:»+374«, «az»:»+994«, «de»:»+49«, «ge»:»+995«, «lt»:»+370«, «lv»:»+371«, «tr»:»+90«, «na»:»+« }

где ключ - двухсимвольное наименование страны, значение - код телефона для данной страны.

Данные должны быть использованы при передаче поля `user_phone` в запросе на создание формы. (для некоторых платежных интерфейсов список должен быть ограничен, если на это указывает параметр «countries» в разделе «additional_params»)

locales

значением является hash вида: {

```
«ru»:«https://secure.onpay.ru/pay\_app/i18n/ru.json»,  
«en»:«https://secure.onpay.ru/pay\_app/i18n/en.json» }
```

где ключ - двухсимвольное обозначение языка перевода, значение - url для получения файла локализации.

Сам файл локализации представляет собой hash многоуровневой вложенности.

Пример:

Если требуется отобразить ключ локализации «user_phone.errors.required» то нужно использовать значение из файла локализации

```
{ «user_phone»:{ «errors»:{ «required»:«This field is required» } } }
```

Пример расчета параметров

Пример полного ответа на запрос получения информации

```
{
```

```
"paysystem_interfaces":{  
  "SBR":{  
    "paysystem":"BBR",  
    "logo":"/assets/payment_systems/logo/SBR.png"  
  },  
  "USD":{  
    "paysystem":"USD",  
    "logo":"/assets/payment_systems/logo/USD.png"  
  },  
  "BBR":{  
    "paysystem":"BBR",  
    "logo":"/assets/payment_systems/logo/BBR.png"  
  }  
},  
"paysystems":{  
  "BBR":{  
    "min":100.0,  
    "max":150000.0,  
    "currency_code":"RUB",  
    "convert_to":"RUR",  
    "commissions":{  
      "pip":1.0,  
      "pif":5.0,  
      "mci":0.0  
    }  
  },  
}
```

```

    "exchange_rates":{
      "USD":0.01597,
      "RUR":1.0,
    }
  },
  "USD":{
    "min":10.0,
    "max":10000.0,
    "currency_code":"USD",
    "convert_to":"USD",
    "commissions":{
      "pip":0.0,
      "pif":0.0,
      "mci":0.0
    },
    "exchange_rates":{
      "RUR":56.980057,
      "USD":1.0
    }
  },
  "RUR":{
    "min":1.01,
    "max":50000.0,
    "currency_code":"RUB",
    "convert_to":"RUR",
    "commissions":{
      "pip":0.0,
      "pif":0.0,
      "mci":0.0
    },
    "exchange_rates":{
      "RUR":1.0,
      "USD"
    }
  }
},
"additional_params":{
  "BBR":{
    "data":[
      {"name":"first_name","regex":"^[A-ZА-Яа-яёЁ\\-\\s']{2,30}$","label":"pay_form_add_p_label.first_name","message":"pay_form_add_p_message.first_name"},
      {"name":"middle_name","regex":"^[A-ZА-Яа-яёЁ\\-\\s']{2,30}$","label":"pay_form_add_p_label.middle_name","message":"pay_form_add_p_message.middle_name"},
      {"name":"last_name","regex":"^[A-ZА-Яа-яёЁ\\-\\s']{2,30}$","label":"pay_form_add_p_label.last_name","message":"pay_form_add_p_message.last_name"},
      {"name":"address","regex":"^[a-zA-Z\\s'0-9,\\-А-Яа-яёЁ,\\d\\/\\\\\\\\.\\\\,\\"]{4,250}$","label":"pay_form_add_p_label.address","message":"pay_form_add_p_message.address"}
    ]
  }
}

```

```

    ],
    "SBR": [ ]
  },
  "USD": {
    "data": [
      { "name": "first_name", "regex": "[A-ZА-Яа-яЁё\\-\\s']{2,30}$", "label": "pay_form_add_p_label.first_name", "message": "pay_form_add_p_message.first_name" },
      { "name": "last_name", "regex": "[A-ZА-Яа-яЁё\\-\\s']{2,30}$", "label": "pay_form_add_p_label.last_name", "message": "pay_form_add_p_message.last_name" },
      { "name": "country", "regex": "[a-zA-Z\\s'\\-А-Яа-яЁё`]{3,70}$", "label": "pay_form_add_p_label.country", "message": "pay_form_add_p_message.country" },
      { "name": "bank_name", "regex": "[A-ZА-ЯЁа-яё\\-\\s']{2,250}$", "label": "pay_form_add_p_label.bank_name", "message": "pay_form_add_p_message.bank_name" }
    ]
  },
  "phone_codes": {
    "ru": "+7",
    "ua": "+38",
    "be": "+375",
    "na": "+"
  },
  "locales": {
    "ru": "https://secure.onpay.ru/pay_app/i18n/ru.json",
    "en": "https://secure.onpay.ru/pay_app/i18n/en.json"
  }
}

```

В данном случае сайт мерчанта должен отобразить плательщику 3 способа оплаты (из раздела «paysystem_interfaces»): «SBR», «BBR» и «USD» с логотипами, ссылки которых указаны в том же разделе. Названия должны быть взяты из нужного файла локализации по ключам «paysystem_interface.SBR.name», «paysystem_interface.BBR.name», «paysystem_interface.USD.name». В качестве доп описания можно использовать ключи вида «paysystem_interface.XXX.description».

Одно из значений «SBR», «BBR», «USD» должно быть использовано в качестве значения параметра «interface_ticker» в запросе на создание ордера.

Из hash-а «paysystems» из полей «convert_to» получаем возможный список значений для поля «ticker» запроса на создание платежной формы. В нашем случае это «RUR» и «USD».

Если плательщик выберет вариант «BBR» или «USD» то он также должен заполнить соответствующие дополнительные поля, для способа «SBR» их нет.

Если магазину необходимо получить 100 USD (в запросе на создание платежного ордера поля ticker=«USD», receive_amount=100.0), то он должен рассчитать сумму, которую должен заплатить плательщик через интерфейс оплаты.

Пример расчета для интерфейса оплаты SBR

1) SBR принадлежит платежной системе BBR, значит для расчета суммы берем курсы обмена из этой платежной системы

```
«exchange_rates»:{ «USD»:0.01597, «RUR»:1.0, }
```

2) Считаем сумму в BBR, которая должна поступить в систему Онпай для обмена на 100 USD
сумма = $100 / 0.01597 = 6261.740763932373$

3) Считаем сумму в BBR, которая должна поступить от плательщика в платежную систему, чтобы в систему Онпай поступило 6261.740763932373 BBR Используем комиссии платежной системы BBR { «pif»:1.0, «pif»:5.0, «mci»:0.0 }

сумма = $(6261.740763932373 + pif) / (1 - (pif / 100))$ (обратный расчет комиссии) =
 $(6261.740763932373 + 5.0) / (1 - (1.0 / 100)) = 6330.041175689265 = 6330.04$ (математическое округление до 2-ого знака)

4) Проверяем минимальную комиссию комиссия = $6330.04 - 6261.75 = 68.31$ если комиссия меньше чем mci, то делаем пересчет суммы как: сумма до обмена + mci ($6261.75 + mci$) в нашем случае это не требуется, так как рассчитанная комиссия больше чем значение минимальной.

5) Проверяем сумму на ограничения платежного интерфейса Берем ограничения от платежной системы BBR { «min»:100.0, «max»:150000.0 }

значение 6330.04 находится в данном диапазоне, следовательно плательщик сможет оплатить через данный способ (в случае превышения лимита НЕ стоит предлагать плательщику платежный интерфейс. У разных интерфейсов разные ограничения)

6) Результирующая сумма к оплате Для получения магазином 100 USD плательщик может заплатить 6330.04 BBR

7) Формируем данные для запроса на создание платежного ордера { «ticker»:«USD», «interface_ticker»:«SBR», «pay_amount»:6330.04, «receive_amount»:100.0 }

8) Отображаем плательщику форму для ввода дополнительных полей В случае SBR - таковых нет. в случае BBR: { «first_name»:«Test», «middle_name»:«Testovich», «last_name»:«Testov», «address»:«Address» }

Другие обязательные поля смотреть в разделе *Создание ордера*

Создание ордера

Запрос

Обязательные параметры

Для создания ордера необходимо отправить POST запрос на адрес <https://secure.onpay.ru/pay> в

формате JSON со следующими обязательными параметрами:

Название	Тип	Значение и описание
user_email	string	Email плательщика
pay_for	string	название платежа
ticker	string	Название валюты ордера (в которой будет начисление магазину)
interface_ticker	string	Название способа оплата (через который плательщик будет платить)
recipient	string	идентификатор магазина
pay_mode	string	тип платежа, fix, free или inv
pay_amount	float	сумма, которую должен заплатить плательщик способом interface_ticker (может быть пустым, в этом случае будет рассчитано автоматически, в зависимости от receive_amount)
receive_amount	float	сумма, которую должен будет получить магазин в валюте ticker
дополнительные поля	string	для некоторых способов оплаты существуют дополнительные необходимые данные(их список можно получить при помощи запроса получения информации)

Необязательные параметры

Название	Тип	Значение и описание
user_phone	hash	телефон плательщика (для некоторых интерфейсов оплаты это поле обязательно, их список можно получить при помощи запроса получения информации)
note	string	дополнительное описание платежа
url_success_enc	string	закодированный в BASE64 url, на который будет перенаправлен плательщик после успешного платежа
url_fail_enc	string	закодированный в BASE64 url, на который будет перенаправлен плательщик после неуспешного платежа
additional_params	hash	см. описание API 2.0

Пример запроса

Пример запроса на создание ордера через системы BBR, которая имеет 4 дополнительных поля(first_name, last_name, middle_name, address):

```
{
  "user_email": "test@email.com",
  "pay_for": "test_pay_for",
  "receive_amount": 100,
  "ticker": "RUR",
  "interface_ticker": "BBR",
  "recipient": "onpay",
  "pay_mode": "free",
  "pay_amount": 101.01,
  "first_name": "test",
  "last_name": "testov",
  "middle_name": "testovich",
  "address": "address",
```



```
"note": "test_note",  
"user_phone": {  
  "code": "+7",  
  "number": "9001234567"  
}
```

```
}
```

Маски для первичной валидации и сообщения об ошибках для дополнительных полей можно получить при помощи запроса информации.

Значениями поля «user_phone.code» может быть одно из значений кодов стран из запроса на получение информации. Для некоторых платежных интерфейсов список кодов ограничивается (см. в разделе *additional_params*)

Ответ

Формат и общее описание

Сервер Onpay отвечает в формате JSON { «redirect_to»: {}, «po_psi_data_request»: {}, «errors»: {} }

Где, одно из полей, errors, redirect_to или po_psi_data_request НЕ пустое (НЕ пустыми могут быть несколько полей).

Алгоритм обработки ответа: Если НЕ пустое «errors»

сайт магазина должен отобразить ошибки плателъщику и переформировать данные для отправки (пример: в случае отсутствия обязательного поля - заполнить данное поле). (подробнее читать в разделе "errors")

Иначе

Если НЕ пустое "redirect_to"

сайт магазина должен перенаправить плателъщика на url, указанный в этом параметре. (подробнее читать в разделе "redirect_to")

Иначе

сайт магазина, должен сформировать форму из данных, содержащихся в поле "po_psi_data_request", и перенаправить пользователя сабмитом данной формы. (подробнее читать в разделе "po_psi_data_request")

errors

в качестве значения errors выступает hash.

Примеры:

```
''"errors": {
```

```

    "receive_amount": [
        "Ошибка в курсе обмена. Возможно курс обмена изменился.
Пожалуйста, попробуйте продолжить операцию еще раз. <a
href='http://wiki.onpay.ru/doku.php?id=oshibki#ошибки_платежной_формы'
target='_blank'>Подробнее</a>"
    ]
}''

```

```

''"errors": {
    "first_name": [
        "pay_form_add_p_message.first_name"
    ],
    "last_name": [
        "pay_form_add_p_message.last_name"
    ]
}''

```

Ключ ошибки - это поле, значение которого неверно. Значение - массив текстов ошибок. Иногда текст ошибки представлен в виде ключа для локализации, пример: «pay_form_add_p_message.first_name». В таких случаях перевод нужно брать из файлов локализации, url которых можно получить при помощи запроса получения информации.

Если присутствует ключ ошибки «system», значит допущена фатальная ошибка при запросе(примеры: неверный формат, пустой запрос)

redirect_to

в качестве значения redirect_to выступает hash вида { «url»:
«<https://secure.onpay.ru/1234567890/instruction>», »...«: »...« }

сайт магазина должен осуществить перенаправление плательщика GET запросом на URL, указанный как значение ключа «url». В ответе могут присутствовать другие ключи, они являются служебными и должны игнорироваться сайтом магазина.

po_psi_data_request

в качестве значения po_psi_data_request выступает hash вида:

```

"po_psi_data_request": {
    "route": {
        "action": "https://card.onpay.ru/",
        "method": "POST"
    },
    "data": {
        "store_name": "onpay",
        "email": "artem.chinkov@kodep.ru",
        "order_id": "123",
        "sum": 103.09,
    }
}

```

```
}  
}
```

Данная запись означает, что пользователь должен быть перенаправлен на url
«<https://card.onpay.ru/>» POST запросом со следующими полями в теле

```
"store_name"="onpay"  
"email"="test@email.com"  
"order_id"="123"  
"sum"=103.09
```

При этом сайт магазина должен передавать значения полей в неизменяемом виде(соблюдая типы данных и не применяя дополнительных функций(округление, изменение регистра)) hash кроме ключей route и data может содержать дополнительные ключи, сайт магазина должен игнорировать их.

From:

<http://wiki.onpay.ru/> - Onpay.ru Wiki

Permanent link:

<http://wiki.onpay.ru/doku.php?id=api-for-pay-form&rev=1445248451>

Last update: **2015/10/19 10:54**

