

Preface

About This Guide

The API Developer Reference describes the mobilPay API for generating mobilPay purchases and understanding the mobilPay API response

Intended Audience

This guide is written for developers who are implementing mobilPay's card and bank transfer processing services using the API.

Documentation Feedback

Help us improve this guide by sending feedback to: documentation@mobilPay.ro

mobilPay API Basics

mobilPay offers a set of application programming interfaces (APIs) that give you the means to incorporate mobilPay functionality into your website applications and mobile apps.

This document describes how to make calls to mobilPay API and how to interpret the instant payment notification (IPN) from mobilPay.

mobilPay API Client-Server Architecture

The mobilPay API uses a client-server model in which your website is a client of the mobilPay server.

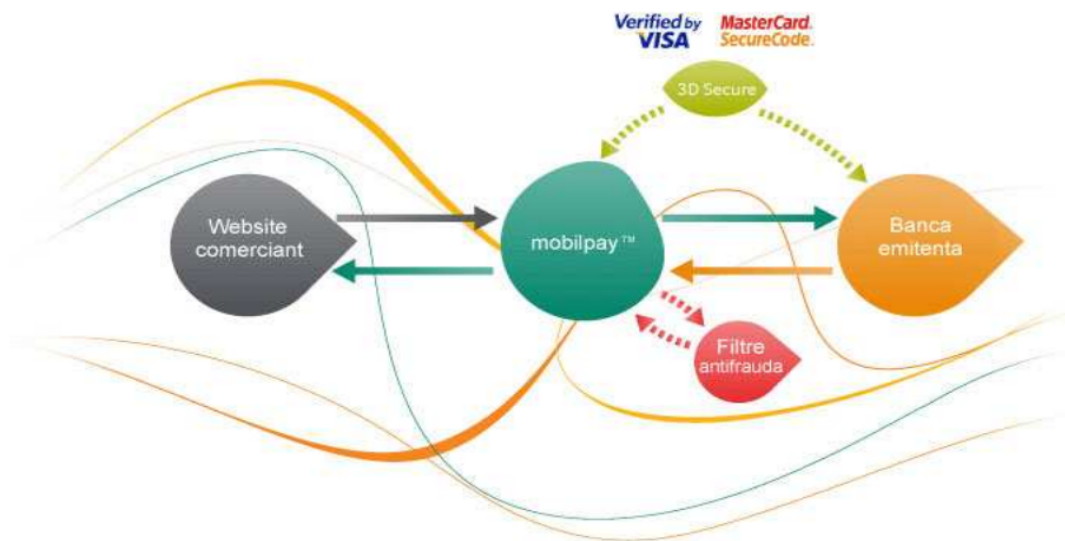
A page on your website initiates an action on a mobilPay API server by sending a request to the server. This request will always be send using the method POST to one of the payment URL (service endpoints) specified. The mobilPay server processes the credit card information presented by the cardholder and responds with an IPN to a callback URL previously specified in the request.

Security

The mobilPay API service is protected to ensure that only authorized mobilPay members use it. There are three levels of security:

1. Request authentication using an API Signature included in the request (Signature field)
2. The data exchanged between the client → mobilPay server and back is encrypted using RSA keys
3. Secure Sockets Layer (SSL) data transport for the request, optional, if available on the merchant side, for the response.

Payment Request/Response Flow



You will redirect the client to mobilPay together with a payment request. This request will have two variables:

env_key – this is the envelope associated with the public key generated upon payment encryption

data – this is the XML structure presented below, signed with the public certificate mobilPay has provided. The certificate is available upon seller account creation in Admin – Seller accounts – Edit – Security settings.

Service Endpoints

You should always start the payment by using POST method for redirecting the client to one of these endpoints:

- standard payment, live mode – <https://secure.mobilpay.ro>
- cvv only payment, live mode – <https://secure.mobilpay.ro/card3>
- token/recurrent payments, live mode – <https://secure.mobilpay.ro/card4>
- standard payment, test mode – <http://sandboxsecure.mobilpay.ro>
- cvv only payment, test mode – <http://sandboxsecure.mobilpay.ro/card3>
- token/recurrent payments, test mode – <http://sandboxsecure.mobilpay.ro/card4>

If you want to display the payment interface in a different language, you should append the language identifier to the specific endpoint, i.e.

<https://secure.mobilpay.ro/en> for English.

Payment Request Structure

The following annotated description of the XML request structure shows the elements required by the mobilPay API.

```
<?xml version="1.0" encoding="utf-8"?>
<order type="card" id="string64" timestamp="YYYYmmddHHiiss">
  <signature>XXXX-XXXX-XXXX-XXXX-XXXX</signature>
  <invoice currency="RON" amount="XX.YY" installments="R1,R2"
    selected_installments="R2" customer_type="2"
    customer_id="internal_id" token_id="token_identifier"
    pan_masked="NNNN">
    <details>Payment Details</details>
    <contact_info>
      <billing type="company|person">
        <first_name>first_name</first_name>
        <last_name>last_name</last_name>
        <email>email_address</email>
        <address>address</address>
        <mobile_phone>mobile_phone</mobile_phone>
      </billing>
    </contact_info>
  </invoice>
  <params>
    <param>
      <name>param1Name</name>
      <value>param1Value</value>
    </param>
  </params>
  <url>
    <confirm>http://www.your\_website.com/confirm</confirm>
    <return>http://www.your\_website.com/return</return>
  </url>
</order>
```

Request Parameters

order type – states the type of transaction that is to be initiated. Must be card;

order id – this is an internal identifier of your order. It should not have more than 64 characters (string) and should be unique for a seller account. Unless you specifically want to make a payment request for the same order, this attribute should be refreshed on each payment request. You will use this identifier when you receive the payment response;

timestamp – this is the timestamp of your server formatted as YYYYMMDDhhmmss

(i.e. 20130925020304 is 2013, September 25th, 02:03:04)

signature – unique key assigned to your seller account for the payment process. Can be obtained from mobilPay's merchant console and has to look like XXXX-XXXX-XXXX-XXXX;

invoice – the details of the payment about to be initiated;

currency – the currency in which the payment will be processed. Should be the ISO code of the currency (i.e. RON for Romanian Leu). You can also set whichever currency you want, but if mobilPay's legal department hasn't cleared you for that currency, the amount will be converted to RON and displayed in dual currency, with RON first;

amount – the amount to be processed. A minimum of 0.10 and a maximum of 99999 units are permitted;

installments – (optional) this is the installments number available for the payment. This is an integer and needs to be previously set in the console by a mobilPay representative;

selected_installments – (optional) this is the number of installments that the client has selected on your web application. It should be an integer within the values specified in the installments attribute;

customer_type – this specifies the customer type within mobilPay. It is optional for standard payments, but mandatory and always has the value “2” for CVV only payments;

customer_id – this is the internal identifier of your customer. It is mandatory for CVV only payments, and should be unique.

token_id – this is the token you have previously received from mobilPay upon a first payment request. It helps identify the customer and the payment instrument;

pan_masked – the first 1 and last 4 digits of the credit card associated with the token_id above. You may use this to present the client with the card he is about to use, in a friendly manner;

details – the details of the payment as they will appear in the mobilPay secure payment page;

contact_info – information regarding the payer. The data here is optional, but if you provide it in the request, the customer will be presented with a more fluent payment experience, where the second step (asking for customer data) will no longer be present;

billing type – the type of customer. It can be either person or company;

first_name – the first name of the customer;

last_name – the last name of the customer;

email – email address of the customer;

mobile_phone – phone number of the customer;

address – address of the customer;

params – you may send an array of custom parameters, with as much data as needed in order to have a large enough number of details regarding the payer and/or the product being paid for;

url – this element specifies where mobilPay will communicate the payment result

confirm – a URL in your web application that will be called whenever the status of a payment changes or a manual IPN is being sent. This is a transparent asynchronous call, however, the first call is always synchronous;

return – a URL in your web application where the client will be redirected to once the payment is complete. Not to be confused with a success or cancel URL, the information displayed here is dynamic, based on the information previously sent to confirm URL.

Payment Response Structure

Upon every change in the status of a payment, mobilPay will make a POST to the URL you have set as **confirm**. mobilPay will construct the parameters in the same way you have done when making the payment request. Data will be encrypted using a X509 public certificate and you will use the private key provided by mobilPay to decrypt it.

You will receive all the parameters you have sent, unchanged, and mobilPay will add another element, called **mobilpay**, to the response XML.

The following annotated description of the XML response structure shows the elements sent by the mobilPay API.

```
<?xml version="1.0" encoding="utf-8"?>
<order type="card" id="string64" timestamp="YYYYMMDDHHMMSS">
  {your_request_XML}
  <mobilpay timestamp="YYYYMMDDHHMMSS" crc="XXXXX">
    <action>action_type</action>
    <customer type="person|company">
      <first_name>first_name</first_name>
      <last_name>last_name</last_name>
      <address>address</address>
      <email>email_address</email>
      <mobile_phone>phone_no</mobile_phone>
    </customer>
    <purchase>mobilPay_purchase_no</purchase>
    <original_amount>XX.XX</original_amount>
    <processed_amount>NN.NN</processed_amount>
    <pan_masked>X****YYYY</pan_masked>
    <payment_instrument_id>ZZZZZZZ</payment_instrument_id>
    <token_id>token_identifier</token_id>
    <token_expiration_date>YYYY-MM-DD
    HH:MM:SS</token_expiration_date>
    <error code="N">error_message</error>
  </mobilpay>
</order>
```


Response Parameters

mobilpay – this is mobilPay's response, appended to your unchanged request;

timestamp – mobilPay's internal timestamp, format YYYYMMDDHHSS;

crc – mobilPay internal identifier check;

action – the action attempted by mobilPay. Possible actions are “new, paid_pending, confirmed_pending, paid, confirmed, credit, canceled”. This is not the status of the transaction, as all actions can either fail or succeed;

customer type – the type of paying customer. This is the data provided to mobilPay by the customer. Can be either person or company.

first_name – the customer's first name, as inserted in the payment page;

last_name – the customer's last name, as inserted in the payment page;

address – the customer's address, as inserted in the payment page;

email – the customer's email address, as inserted in the payment page;

mobile_phone – the customer's phone, as inserted in the payment page;

purchase – mobilPay internal identifier. This is unique for the entire mobilPay platform;

original_amount – the original amount processed;

processed_amount – the processed amount at the moment of the response. It can be lower than the original amount, ie for capturing a smaller amount or for a partial credit;

pan_masked – first 1 and last 4 digits of the credit card used. Store this and use it for presenting the customer with a friendly way of identifying the payment instrument;

payment_instrument_id – this is an internal mobilPay identifier of the payment instrument, and allows you to check for its uniqueness;

token_id – this is the token associated with the payment instrument. You can use it to initiate recurring payments;

token_expiration_date – the expiration date of the token. It is usually set to the expiration date of the card;

error code – the error code states whether the action has been successful or not. A 0 (zero) value states that the action has succeeded. A different value means it has not;

error message – the error message associated to the error code. This is generally a message that can be presented to the user in order to help him understand why a transaction has been rejected, or if it has been approved.

Merchant's Response

For each call to your confirm URL, you will need to send a response in XML format back to mobilPay, in order to help us understand whether you have successfully recorded the response or not. For debugging purposes, you may view your response in mobilPay console (Order – Details – Merchant Communication Log)

The following annotated description of the XML response structure shows the elements sent by you to the mobilPay API.

```
<?xml version="1.0" encoding="utf-8" ?>
<crc error_type="1|2" error_code="{numeric}">{message}</crc>
```

The attributes of the **crc** element are only sent if you had any problem recording the IPN, and have the following meaning

error_type – based on this mobilPay will activate a resend IPN mechanism or not. If its value is 1, it means you encountered a temporary error. Set it to 2 if you encountered a permanent error;

error_code – this is your internal error code, helping you to view the error generated by your web application;

message – if you encountered an error while processing the IPN, this should be your error message, helping you find the error. If no error occurred, you should set this to the **crc** value received in the IPN or something else that you may find suitable

Error Code Values

- 0 – approved
- 16 – card has a risk (i.e. stolen card)
- 17 – card number is incorrect
- 18 – closed card
- 19 – card is expired
- 20 – insufficient funds
- 21 – cVV2 code incorrect
- 22 – issuer is unavailable
- 32 – amount is incorrect
- 33 – currency is incorrect
- 34 – transaction not permitted to cardholder
- 35 – transaction declined
- 36 – transaction rejected by antifraud filters
- 37 – transaction declined (breaking the law)
- 38 – transaction declined
- 48 – invalid request
- 49 – duplicate PREAUTH
- 50 – duplicate AUTH
- 51 – you can only CANCEL a preauth order
- 52 – you can only CONFIRM a preauth order
- 53 – you can only CREDIT a confirmed order
- 54 – credit amount is higher than auth amount
- 55 – capture amount is higher than preauth amount
- 56 – duplicate request
- 99 – generic error