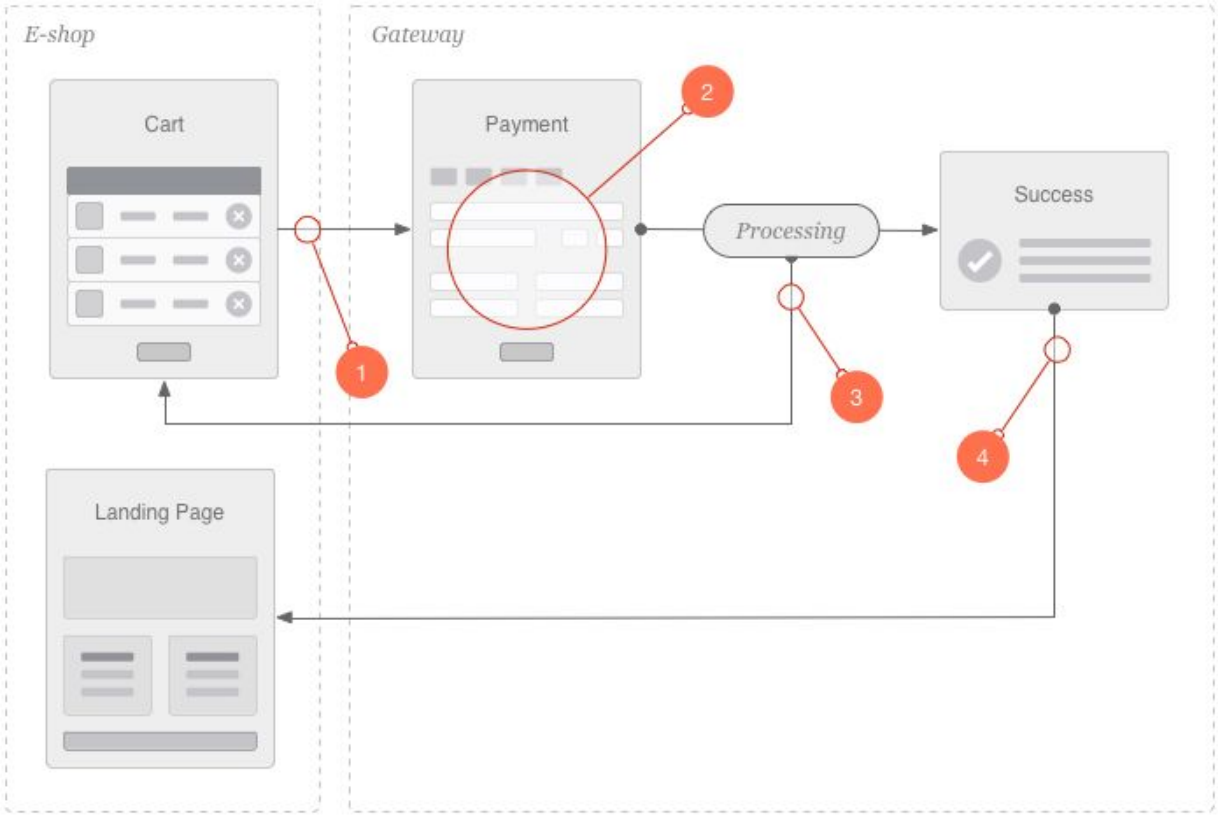

# EveryPay Payment API Documentation

2016-11-25

# Contents

# 1. Payment Process Overview

The process of collecting a payment is displayed on the diagram below:




On high level the steps in the payment collection are as follows:

1. Payment initiation API request is submitted from the merchant's server to EveryPay server.
2. EveryPay Payment Page is displayed to the buyer in one of the following ways:
    a. Embedded on merchant's website (using HTML iFrame)
    b. Buyer is redirected to EveryPay payment page
3. Once the buyer has provided the card details (and performed 3D-Secure bank authentication when required), EveryPay server will process the payment (which includes communication with all required external systems).
4. Once the payment is processed, depending on the payment page solution:
    a. Redirect payment page -- the buyer will be redirected back to the merchant's website
    b. iFrame payment form -- the buyer will see the successful/failed message inside the iFrame

After completing this process, asynchronous automatic callback notification with the transaction status and details is sent from EveryPay server back to the merchant's server. Based on this the merchant server will update the order status depending on the payment result.

# 2. Business Considerations

## 2.1 Charge Payment Type

Charge is an authorisation followed by automatic capture -- request to transfer the funds for the delivery of goods or services automatically, without the need for further manual activities in Merchant Portal:

1) Authorisation is done when the cardholder initiates the payment. It's a process for getting approval to the transaction and booking the required amount on the card (without actually transferring the funds yet).
2) Capture is performed automatically at 0:30 for transactions that were performed on the previous date between 0:00 and 23:59 UTC+2 (or UTC+3 during daylight saving time).
3) The funds are transferred to the merchant's bank account about 12 hours after the capture.

Capture delay functionality can be used if needed to delay the automatic capture of the authorisation for up to 8 days. This provides extra time buffer before fulfilling the customer's order and charging the money from their bank account, which might be required for certain business models, for example:

- to perform additional background checks on the customer (to prevent possible fraud)
- to verify the availability of goods/services

## 2.2 Payment Form Types

There are two different types of payment forms to choose from:

**1) Redirect payment page**
After completing the checkout process on merchant's website the customer will be redirected to EveryPay redirect payment page. After completing the payment process the customer will be redirected back to the merchant's website.

The benefits:

- Simpler integration and less changes required on merchant's website.

The downside:

- Redirecting the customer away from the merchant's website hinders the user experience.

- The redirect payment page used EveryPay visual design which doesn't match the merchant's website design.

**2) Embedded payment form (HTML iFrame)**
The payment form will be embedded directly into the merchant's website checkout page.

The benefits:
- Customer will not be redirected away from the merchant's website.
- The payment form visual design is customizable in EveryPay Merchant Portal and can be configured to closely match the visual design of the merchant's website.

The downside:
- Slightly more complex integration and more changes required on merchant's website
- If the merchant's website is running on HTTP, not seeing the HTTPS encryption sign in web browser can scare off some customers, even though the traffic inside the iFrame is using HTTPS.

**Which one should you use?**
For most merchants the embedded payment form is the recommended option as it helps to improve user experience considerably compared to the redirect payment page. However, if the merchant's website uses HTTP, it might be better to use the redirect payment page as seeing the HTTPS encryption sign (while entering their credit card details) can provide the extra confidence required to prevent them from running away with fears of security risks.

# 3. Technical Aspects

## 3.1 API URL and Authentication

Gateway API endpoint URLs (for HTTP POST form target) in EveryPay environments:
- **Test**: https://igw-demo.every-pay.com/transactions/
- **Production**: https://pay.every-pay.eu/transactions/

API username and secret are different in Production and Test environment. They can be found in corresponding Merchant Portal:
- **Test**: https://mwt-demo.every-pay.com/merchant_settings/general
- **Production**: https://portal.every-pay.eu/merchant_settings/general

## Helper libraries

EveryPay provides the following helper libraries to simplify the integration:
- PHP library -- to simplify the entire integration process (initiating the payment API requests and processing the callback/redirect responses), including HMAC calculation

- JavaScript library -- to simplify iFrame payment page integration

The helper libraries are available in our Github repository:
https://github.com/UnifiedPaymentSolutions/everypay-integration

## 3.2 Nuances

### nonce

All Everypay message responses contain the ‘nonce’ field, that can be used to verify uniqueness of the response message. This approach helps to prevent possible message replay attacks.

Validating the uniqueness of the returned ‘nonce’ value on merchant's side will add an extra layer of security. It requires storing ‘nonce’ values of the received messages and comparing the ‘nonce’ values of the received messages against the stored ‘nonce’ values.

### order_reference

The main purpose of ‘order_reference’ parameter is to match the payment in EveryPay’s system to the correct corresponding order in merchant’s e-shop. ‘order_reference’ must be unique. As an additional benefit, matching the ‘order_reference’ and validating its uniqueness in merchant’s e-shop provides an extra layer of security against tampering attacks.

### HMAC

HMAC (a keyed-hash message authentication code) is used to verify API request:

1) data integrity
2) authentication

EveryPay uses SHA-1 lower hexadecimal HMAC format.

HMAC is constructed by concatenating form parameter keys and values into a string ordered by alphabetic order of the key name and returning its hexdigest with shared secret. Shared secret key can be obtained (and regenerated, when needed for security reasons) in Merchant Portal in the Settings section.

HMAC calculation can be tricky and it’s important to pay extra attention to which API parameters and how should be included in HMAC calculations. For example:

- The hmac_fields field is used in all stages of the payment (initiation, callback, redirect,

cancellation) to assist with providing required information about which fields are included in HMAC calculation.
- If not told otherwise (for a specific API parameter, e.g. locale) all API request parameters must be included in HMAC calculation.
- HMAC string contents should not include code escapes -- neither for payment initiation API request or the callback/redirect response (e.g. the JSON contents of 'processing_errors' and 'processing_warnings').

The most common mistakes with HMAC calculation are:
- not including all required parameters and their values in calculation
- using code escapes

HMAC string example:

```
account_id=EUR3D1&amount=10&api_username=abcd1234abcd1234&callback_url=http://www.google.ee/
?q=callback&customer_url=http://www.google.ee/?q=redirect&hmac_fields=account_id,amount,api_
username,billing_address,billing_city,billing_country,billing_postcode,callback_url,customer
_url,delivery_address,delivery_city,delivery_country,delivery_postcode,email,hmac_fields,non
ce,order_reference,timestamp,transaction_type,user_ip&nonce=d6e33441e0171249e71992946896f754
&order_reference=98c9fa2e52f0679610935497ff4da714&timestamp=1397038382&transaction_type=char
ge&user_ip=82.131.119.82
```

# 4. One-Off Payment

EveryPay Payment API expects the API request in HTTP POST format.

Description of the optional field markings:
O - field is optional
OF - field is optional, but including it improves fraud detection (therefore it's highly recommended to provide this information whenever possible)

## Parameters

| Field name | Optional | Description |
|---|---|---|
| api_username | | Merchant API username. The value can be found in Merchant Portal in the Settings section. |
| account_id | | Processing account to be used for the transaction. Processing account defines the transaction processing configuration, including the currency to be used, pricelist, 3D Secure settings, payment recurrence, clearing settings, etc. |

| | | |
|---|---|---|
| | | Example values could be: ‘EUR3D1, ‘USD3D2’.<br><br>The field value can be found in Merchant Portal in the Settings section. |
| nonce | | Random unique value to prevent replay attacks |
| timestamp | | Time of creating the transaction. Expressed as seconds from January 1, 1970 UTC |
| callback_url | | Once EveryPay gateway has processed the transaction, processing result data is posted to this URL. |
| customer_url | | When the buyer clicks on the Back button, he will be redirected to this URL. |
| email | | E-mail address (buyer) |
| amount | | Payment amount |
| order_reference | | Order reference, must be unique for every payment attempt |
| user_ip | | IP-address (buyer) |
| billing_address | OF | Billing address |
| billing_country | OF | Billing country, in two-character ISO 3166[1] format |
| billing_city | OF | Billing city |
| billing_postcode | OF | Billing postcode |
| delivery_address | O | Delivery address |
| delivery_country | O | Delivery country, in two-character ISO 3166 format |
| delivery_city | O | Delivery city |
| delivery_postcode | O | Delivery postcode |
| hmac | | HMAC is constructed by concatenating form parameter keys and values into a string ordered by alphabetic order of the key name and returning its hexdigest with shared secret. Shared secret key can be viewed and changed in Merchant Portal in the Settings section. |
| hmac_fields | | It is a string which contains all fields (keys) that are going to be used in hmac calculation, separated by comma. “hmac_fields” should contain the key “hmac_fields” and |

[1] https://en.wikipedia.org/wiki/ISO_3166

| | | |
|---|---|---|
| | | should not contain the key “hmac”.<br><br>E.g:<br>"account_id,amount,api_username,callback_url,customer_url,hmac_fields,nonce,order_reference,timestamp,transaction_type,user_ip" |
| transaction_type | | Use fixed value ‘charge’ |
| locale | O | Sets the locale and the language of the EveryPay Payment Page user interface displayed to the customer.<br>Defaults to ‘en’.<br>Accepted values are:<br>● ‘en’ - English<br>● ‘et’ - Estonian<br>● ‘fi’ - Finnish<br>● ‘de’ - German<br>● ‘lv’ - Latvian<br>● ‘lt’ - Lithuanian<br>● ‘ru’ - Russian<br>● ‘es’ - Spanish<br>● ‘sv’ - Swedish<br>Please contact the Everypay support if you need other supported locales.<br>***Note:*** This field must not be included for HMAC calculation. |
| request_cc_token | O | Accepted values are:<br>● ‘1’ - cc_token will be returned in callback<br>● ‘0’ (default) - cc_token will not be returned in callback |
| cc_token | O | Enables payment with stored card token. If a valid cc_token is provided, the buyer will not be asked to fill in card details. Can be used in two ways:<br>1) separately - regular token payment without any user input (using saved card number and expiration date)<br>2) together with token_security field (to use added payer/card authentication measures) |
| token_security | O | To be used only together with the cc_token parameter. Enables token payments (with saved card data) with added payer/card authentication measures. Possible values are:<br>● ‘none’ - regular token payment without any user input (using saved card number and expiration |

| | | |
|---|---|---|
| | | date)<br>● 'cvc' - payer must provide CVC code<br>● '3ds' - payer must perform 3DS authentication<br>● 'cvc_3ds' - payer must provide CVC code and perform 3DS authentication |
| skin_name | O | Skin name for the embedded payment form (HTML iFrame).<br><br>This parameter should only be used if the required changes are made on merchant's website to embed EveryPay iFrame in the checkout process. |

## HTML form example

```
<form action="https://igw-demo.every-pay.com/transactions/" method="post">
  <input name="hmac" value="75ed21e06d7e3ed26d1eb8b3fab24bdf3d73df20">
  <input name="hmac_fields"
value="account_id,amount,api_username,billing_address,billing_city,billing_country,billing_po
stcode,callback_url,customer_url,delivery_address,delivery_city,delivery_country,delivery_pos
tcode,email,hmac_fields,nonce,order_reference,timestamp,transaction_type,user_ip">
  <input name="transaction_type" value="charge">
  <input name="locale" value="en">
  <input name="amount" value="10.0">
  <input name="api_username" value="1adcffbf5e2df582">
  <input name="account_id" value="EUR1">
  <input name="billing_address" value="Tamme 2">
  <input name="billing_city" value="Tallinn">
  <input name="billing_country" value="EE">
  <input name="billing_postcode" value="12345">
  <input name="callback_url" value="http://www.google.ee/?q=callback">
  <input name="customer_url" value="http://www.google.ee/?q=redirect">
  <input name="delivery_address" value="Kuuse 3">
  <input name="delivery_city" value="Tallinn">
  <input name="delivery_country" value="EE">
  <input name="delivery_postcode" value="12345">
  <input name="email" value="email@example.org">
  <input name="nonce" value="d6e33441e0171249e71992946896f754">
  <input name="order_reference" value="98c9fa2e52f0679610935497ff4da714">
  <input name="timestamp" value="1397038382">
  <input name="user_ip" value="82.131.119.82">
  <input class="btn btn-danger" type="submit" value="Proceed to Payment">
</form>
```

# 5. Redirect Payment Page

## 5.1 Redirect Customer Back to Merchant

Once the buyer has completed the payment process, she will be redirected back to the merchant's website -- to the URL specified in customer_url field. The redirect includes HTTP PUT message with payment result and other related information.

## HTTP PUT Parameters

| Field name | Optional | Description |
|---|---|---|
| api_username | | Merchant API username |
| account_id | | Processing account ID that was used to process the transaction |
| nonce | | Random unique value to prevent replay attacks. Validating the uniqueness of the returned 'nonce' value on merchant's side will add an extra layer of security. |
| timestamp | | Seconds from January 1, 1970 UTC |
| amount | | Transaction amount |
| order_reference | | Order reference |
| payment_reference | | Payment reference ID |
| payment_state | | Current status of the payment which reflects the definitive payment outcome based on which the order status should be updated in merchant system.<br><br>Possible scenarios:<br>● 'settled' or 'authorised' (successful payments)<br>● 'cancelled' or 'waiting_for_3ds_response' (cancelled/incomplete payments)<br>● 'failed' (failed payments) |
| hmac | | Calculated HMAC for request authenticity verification. Use only the fields listed in hmac_fields for HMAC calculation, all the other fields should be ignored. |
| hmac_fields | | It is a string which contains all fields (keys) that are going to be used in hmac calculation, separated by comma. "hmac_fields" contains a key "hmac_fields" and does not |

| | | |
|---|---|---|
| | | contain a key “hmac”.<br><br>E.g:<br>"account_id,amount,api_username,hmac_fields,nonce,order_reference,payment_reference,payment_state,timestamp,transaction_result" |
| transaction_result | | Should only be considered as additional information about different phases of the payment.<br><br>Possible values<br>● ‘completed’ (successful payments)<br>● ‘cancelled’ (cancelled/incomplete payments)<br>● ‘failed’ (failed payments)<br><br>The definitive outcome of the payment attempt is reflected by **payment_state** parameter based on which the order status should be updated in merchant system. |
| cc_token | O | cc_token will be returned only if it was requested in the Payment Initiation. |
| cc_last_four_digits | | Last four digits of the card number. |
| cc_month | | Card expiration month (as entered by the cardholder in payment form), in mm format (e.g. 1 or 12) |
| cc_year | | Card expiration year (as entered by the cardholder in payment form), in YYYY format (e.g. 2017) |
| cc_holder_name | | Cardholder name (as entered by the cardholder in payment form) |
| cc_type | | Card type. Possible values are ‘visa’ and ‘master_card’. |

It’s important to notice that in case of using the redirect payment page, the merchant server will receive two messages about the payment:

- asynchronous server-to-server callback (HTTP POST) from EveryPay server to merchant server which contains full details about the payment
- synchronous client-to-client redirect message (HTTP PUT) from EveryPay payment page to merchant’s shop (only relevant for redirect, not iFrame payment page solution) which contains a subset of the callback message details (as we don’t want the buyer to see some of the payment details which are only meant for merchant’s use)

The contents of the two messages overlap partially. As browser redirect usually takes more time the callback message is usually delivered before the redirect message. However, it can also

happen the other way around. If the additional parameters in callback message will be used for something, the order records on merchant side should be updated accordingly if the callback message is delivered later than the redirect message.

## 5.2 Payment Cancellation

When using redirect payment solution, it’s possible for the to cancel the payment by clicking BACK button on the redirect payment page. After clicking the button, the buyer will be redirected back to the merchant’s website (to the URL specified in API request customer_url parameter).

### Parameters

The redirect includes HTTP PUT request with the following parameters:

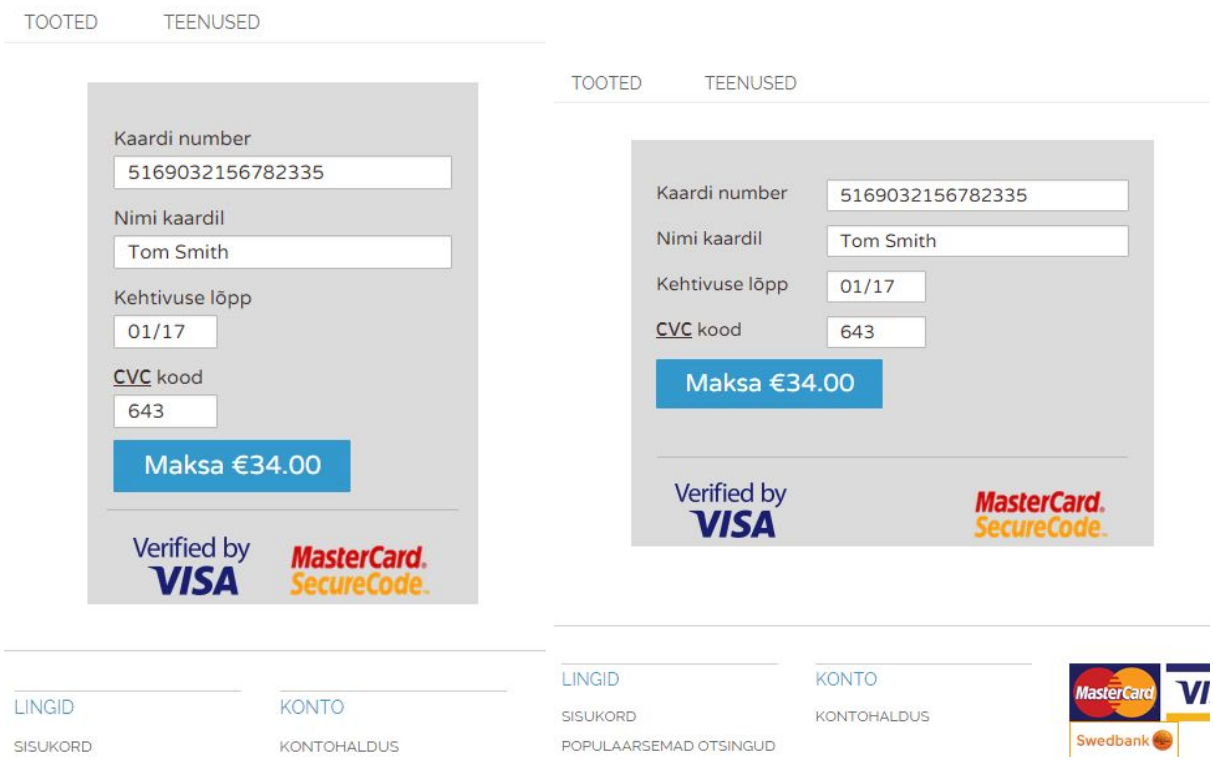
| Field name | Description |
|---|---|
| nonce | Random unique value to prevent replay attacks. Validating the uniqueness of the returned 'nonce' value on merchant's side will add an extra layer of security. |
| timestamp | Seconds from January 1, 1970 UTC |
| order_reference | Order reference |
| payment_state | Current status of the payment which reflects the definitive payment outcome based on which the order status should be updated in merchant system.<br><br>In case of payment cancellation the value is ‘cancelled’. |
| hmac | Calculated HMAC for request authenticity verification. Use only the fields listed in hmac_fields for HMAC calculation, all the other fields should be ignored. |
| hmac_fields | It is a string which contains all fields (keys) that are going to be used in hmac calculation, separated by comma. “hmac_fields” contains a key “hmac_fields” and does not contain a key “hmac”.<br><br>E.g:<br>"api_username,hmac_fields,nonce,order_reference,payment_state,timestamp,transaction_result" |
| transaction_result | In case of a cancellation the value is ‘cancelled’. |
| api_username | Merchant API username |

# 6. Embedded Payment Form (iFrame)

If you don't want buyers to redirect away from your site during the checkout process, use the embedded payment form solution by integrating EveryPay iFrame payment form in payment flow. The design of the iFrame payment form contents can be fully customised to match the shop's look-and-feel. The customization can be done in Merchant Portal "iFrame skins" section under Settings:

- **Test**: https://mwt-demo.every-pay.com/merchant_settings/skins
- **Production:** https://portal.every-pay.eu/merchant_settings/skins

The contents of the iFrame are responsive and displayed according to the iFrame size. When iFrame width is below 400px, the field labels are displayed above (not in front of) the fields. This allows for implementing responsive design for the iFrame size in shop.




With iFrame payment page (unlike with redirect payment page) the buyer is not automatically redirected to customer_url after completing the payment process. The iFrame will contain the successful or unsuccessful payment page. Redirection inside iFrame is not done because otherwise merchant website would be displayed inside the iFrame. To redirect the buyer to a

different page after completing the payment process, the merchant website can rely on the iFrame-to-parent post messages which contain information about payment result.

EveryPay provides the required tools for customizing the design of the payment form inside the iFrame and ensures that the functional aspects work properly. The merchant should make sure that the iFrame itself fits nicely with the shop layout and design.

## Parameters

If parameter skin_name is included in the API request, the payment form will be returned with iFrame compatible design.

| Field name | Description |
|---|---|
| skin_name | Use ‘default’ if you want to use default skin design. |

## iFrame-to-parent communication

Using iFrames in a cross-domain environment has certain security restrictions. There are few scenarios where these security restrictions set significant limitations to web developers and make it difficult to more achieve decent end-user-experience. Because of this, EveryPay has developed a JavaScript helper library to assist merchants with the integration: https://github.com/UnifiedPaymentSolutions/everypay-integration/tree/master/javascript

The JavaScript has two main purposes:

1) receiving payment status messages from iFrame
2) resizing 3D-Secure authentication page

### Payment status message

Receiving a message about the payment status (success or failure) directly from the iFrame, right after the buyer has completed the payment, allows the merchant website to display relevant message to the buyer immediately after the payment process inside the iFrame has been completed. Otherwise it would be necessary to “poll” for the arrival of the callback message from EveryPay server to merchant server.

Due to standard cross-domain iFrame-to-parent security restrictions the JavaScript includes required code to overcome these restrictions. The iFrame sends postMessage to the domain specified in the customer_url API request parameter.

The message from iFrame to parent includes two types of fields:

1) field ‘transaction_result’ (‘completed’ or ‘failed’, depending on the result of the payment attempt)

2) success/error messages (contains the same texts that the buyer would see inside the iFrame)

Some message fields are not always present.

| Field name | Optional | Description |
|---|---|---|
| transaction_result | | Possible values are ‘completed’ and ‘failed’ |
| message_title | | E.g. "Thank you! Payment successful." or "Sorry, payment was unsuccessful." |
| message_error | O | Information about the cause of payment failure, if the exact reason is known and can be revealed to the buyer. |
| message_action | O | Suggestions about what the buyer should do to perform a successful payment. |
| message_contact | O | Instructions about who the customer should contact if the problem persists -- the merchant or the card issuing bank. |

When a payment is initiated with a card token (without the need for buyer to enter card details), thanks to these messages, it’s not even always necessary to display the full iFrame itself as no user interface is required -- the iFrame can be included on the page as a hidden element and will only be necessary to receive the (error) messages to display to the buyer using the shop’s visual design styles (font, color etc).

### Resizing 3D-Secure page

MasterCard, Visa and other card schemes have established certain standards to banks about the size of the 3D-Secure authentication page -- roughly 400x400px. However, unfortunately not all banks follow this standard. Therefore it’s better to automatically resize the iFrame to ensure that even bigger 3D-Secure pages will fit nicely without the need for scrolling.

## iFrame implementation example

```
<iframe id="iframe-payment-container" name="iframe-payment-container", width="400",
height="400"></iframe>

<form action="https://igw-demo.every-pay.com/transactions" id="iframe_form" method="post"
style="display: none" target="iframe-payment-container">
  <input name="hmac" value="75ed21e06d7e3ed26d1eb8b3fab24bdf3d73df20">
  <input name="hmac_fields"
value="account_id,amount,api_username,callback_url,customer_url,locale,nonce,order_reference
,skin_name,timestamp,transaction_type,user_ip">
  <input name="transaction_type" value="charge">
  <input name="locale" value="en">
  <input name="amount" value="1.0">
```

```
  <input name="api_username" value="b3616e26a91d3cb4">
  <input name="account_id" value="EUR1">
  <input name="callback_url" value="http://www.google.ee/?q=callback">
  <input name="customer_url" value="http://www.google.ee/?q=redirect">
  <input name="nonce" value="30d7810d31dbb77d4300fd3f6a59ff11">
  <input name="order_reference" value="98c9fa2e52f0679610935497ff4da714">
  <input name="timestamp" value="1437488204">
  <input name="user_ip" value="82.131.119.82">
  <input name="skin_name" value="default">
</form>
<script>
  window.onload = function() { document.getElementById("iframe_form").submit(); }
</script>
```

# 7. Payment With Saved Card

To facilitate the payment process for buyer, it's possible to save the card token on merchant server and allow future payments to be made automatically using the card token, without the need for the buyer to enter the full set of card details in the payment form. The token refers to the unique combination of card number and expiration date, stored securely in EveryPay server.

For added security it's possible to ask the buyer to enter also CVC code and/or perform 3DS authentication.

The token is only usable under a single merchant and can't be converted back to card number and expiration date outside EveryPay servers, eliminating the risk of card fraud.

**IMPORTANT!** The merchant is responsible for complying with any regulations that might be involved with storing the card token. According to the information available to EveryPay, there's a European Union regulation that requires the merchant to ask for their customer's permission before storing the card token for future payments. The permission must be asked as opt-in (not opt-out) -- the customer must deliberately express her consent with this, e.g. by checking a checkbox (the checkbox must not be checked by default). The merchant is responsible for retaining a copy of that permission.

There are two ways to obtain the card token - either via 0-amount tokenisation or in response to the first payment.

## 7.1 0-amount tokenisation

0-amount tokenisation (a.k.a account check) means performing a 0-amount authorisation with the card to verify its validity. The token will be returned in the callback message for successful authorisations. The request is almost identical with one-off payment request. The only difference is 'transaction_type' parameter value.

| Field name | Description |
|---|---|
| transaction_type | Value ‘tokenisation’ |

## 7.2 Get token with first payment

To request the token to be returned in the callback message, request_cc_token parameter must be included in the one-off payment request. The token will be returned in the callback message for successful authorisations.

| Field name | Description |
|---|---|
| request_cc_token | Accepted values are:<br>● ‘1’ - cc_token will be returned in callback<br>● ‘0’ (default) - cc_token will not be returned in callback |

### Callback parameters

If the payment was processed successfully, the callback message includes the card token in cc_token parameter. Together with the last four digits of the card number, the token can be easily stored and referred to on merchant side.

| Field name | Description |
|---|---|
| cc_token | cc_token will be returned only if it was requested in the Payment Initiation. |

## Payment with token

When the API request contains the optional cc_token parameter with a valid token value, the payment is processed as token payment. Optionally also token_security parameter can be included.

| Field name | Description |
|---|---|
| cc_token | Enables payment with stored card token. Can be used in two ways:<br>1) separately - regular token payment without any user input (using saved card number and expiration date)<br>2) together with token_security field (to use added payer/card authentication measures) |

| | |
|---|---|
| token_security | To be used only together with the cc_token parameter. Enables token payments (with saved card data) with added payer/card authentication measures. Possible values are:<br>● 'none' - regular token payment without any user input (using saved card number and expiration date)<br>● 'cvc' - payer must provide CVC code<br>● '3ds' - payer must perform 3DS authentication<br>● 'cvc_3ds' - payer must provide CVC code and perform 3DS authentication |

# 8. Callback to Merchant

Once the payment attempt is completed (successfully or not), EveryPay system will send a callback message (HTTP form POST) to the URL defined in the payment initiation request callback_url field. If the first attempt fails (i.e. URL doesn't return HTTP 2xx or 3xx response code), EveryPay server will attempt to resend the callback several times over the following 72 hours until it succeeds (or fails permanently).
When Merchant checks hmac on callback, fields used for calculating the hmac must be taken dynamically, so any added field would not break system and give wrong hmac as result.

## Parameters

| Field name | Optional | Description |
|---|---|---|
| api_username | | Merchant API username |
| account_id | | Processing account ID that was used to process the transaction |
| nonce | | Random unique value to prevent replay attacks. Validating the uniqueness of the returned 'nonce' value on merchant's side will add an extra layer of security. |
| timestamp | | Seconds from January 1, 1970 UTC |
| amount | | Transaction amount, dot as decimal separator |
| order_reference | | Order reference |
| payment_reference | | Payment reference ID |
| payment_state | | Current status of the payment which reflects the definitive payment outcome based on which the order status should be updated in merchant system. |

| | | |
|---|---|---|
| | | Possible scenarios:<br>● ‘settled’ or ‘authorised’ (successful payments)<br>● ‘cancelled’ or ‘waiting_for_3ds_response’ (cancelled/incomplete payments)<br>● ‘failed’ (failed payments) |
| processing_errors | | Array of error hashes in JSON format.<br>***Note:*** Don’t use code escapes for HMAC calculation. |
| processing_warnings | | Hash of fraud check warnings in JSON format.<br>***Note:*** Don’t use code escapes for HMAC calculation. |
| hmac | | Calculated HMAC for request authenticity verification over the fields in the response message. Use only the fields listed in hmac_fields for HMAC calculation, all the other fields should be ignored. The HMAC must be generated based on the exact returned fields, i.e. without code escapes for processing_errors and processing_warnings fields. |
| hmac_fields | | It is a string which contains all fields (keys) that are going to be used in hmac calculation, separated by comma. “hmac_fields” contains a key “hmac_fields” and does not contain a key “hmac”.<br><br>E.g:<br>"account_id,amount,api_username,hmac_fields,nonce,order_reference,payment_reference,payment_state,timestamp,transaction_result,user_ip" |
| transaction_result | | Should only be considered as additional information about different phases of the payment.<br><br>Possible values<br>● ‘completed’<br>● ‘cancelled’<br>● ‘failed’<br><br>The definitive outcome of the payment attempt is reflected by **payment_state** parameter based on which the order status should be updated in merchant system. |
| cc_token | O | cc_token will be returned only if it was requested in the Payment Initiation. |
| cc_last_four_digits | | Last four digits of the card number. |

| | | |
|---|---|---|
| cc_month | | Card expiration month (as entered by the cardholder in payment form), in mm format (e.g. 1 or 12) |
| cc_year | | Card expiration year (as entered by the cardholder in payment form), in YYYY format (e.g. 2017) |
| cc_holder_name | | Cardholder name (as entered by the cardholder in payment form) |
| cc_type | | Card type. Possible values are ‘visa’ and ‘master_card’. |
| cc_issuer_country | | Card issuing bank country, in alpha-2 format (e.g. ‘EE’). |
| cc_issuer | | Card issuing organization |
| stan | O | Payment STAN reference number, in numeric format. Returned only in case the payment was sent for processing to card processor and didn’t fail already on EveryPay level. |
| state_3ds | | Payment 3D-Secure authentication status. Possible values are:<br>● ‘3ds’ - authentication completed successfully<br>● ‘attempted’ - authentication attempted but not completed<br>● ‘failed’ - authentication failed<br>● ‘no3ds’ - no 3DS coverage for payment<br>● ‘unknown’ - 3DS status unknown |
| fraud_score | | Payment fraud score, calculated by EveryPay fraud prevention engine, in numeric format. |

# 9. Integration Testing

## 9.1 Test cards

The following test cards can be used to perform successful test payments:

| Card type | Card number | Expiration date | CVC code | Cardholder name |
|---|---|---|---|---|
| MasterCard | 5168832136360487 | 10/18 | 119 | (any name) |
| Visa | 4761739001010119 | 12/25 | (any code) | (any name) |

The 3DS authentication simulator (Poseidon bank) password is ‘secret’.

To test failed payments, the easiest ways are to enter incorrect expiration date or incorrect 3DS password.

## 9.2 Testing checklist

As the entire payment workflow has several steps , it's necessary to verify that all steps and possible scenarios are working flawlessly.

The testing should include the following scenarios:

1) Sending a valid (format wise) API request for payment initiation
2) Correct processing of the different payment result response messages from EveryPay system:
    a) Asynchronous server-to-server callback message
    b) The redirect HTTP PUT message (only relevant for redirect payment page solution)
    c) The iFrame-to-parent post message (only relevant for iFrame payment page solution)
3) Updating the order status correctly in merchant system based on the payment response(s) received from EveryPay system
4) Displaying the correct page and contents to the buyer in merchant system for different scenarios
5) Compare the payment result in EveryPay merchant Portal with the order status in merchant system to ensure that the merchant systems order status reflects the expected result.
6) All of the steps above (where relevant) should be verified with different possible payment scenarios:
    a) Successful payment (payment_state='settled' or 'authorised')
    b) Failed payment (payment_state='failed')
    c) Cancelled payment (payment_state='cancelled' or 'waiting_for_3ds_response') -- only relevant for redirect payment page, when buyer clicks the BACK button on payment page
    d) Abandoned payment -- nothing is returned from EveryPay system because the buyer didn't finish the payment process (e.g. closed the browser window).

# 10. Changelog

| Date | Change |
|---|---|
| 2016-10-26 | Added<br>● cc_issuer to callback fields.<br>● Added guides to callback hmac scheck |
| 2016-07-16 | Changed:<br>New testcard for MasterCard |
| 2016-05-16 | Added:<br>● HMAC string example<br>● “0-amount tokenisation” section<br>● new ‘locale’ values - de, es, sv |
| 2016-04-15 | Clarified payment_state and transaction_result usage |
| 2016-03-21 | Added:<br>● test cards<br>● new ‘locale’ values - fi, lv, lt |
| 2016-02-12 | Removed transaction_type ‘authorisation’ |
| 2016-01-21 | Added token_security parameter |
| 2016-01-15 | Content reordering and restructuring<br>More thorough explanations added to several payment process aspects |
| 2015-12-17 | Added new parameters to payment responses - callback and return redirect:<br>● cc_month<br>● cc_year<br>● cc_holder_name<br>● cc_type<br>● cc_issuer_country<br>● stan<br>● state_3ds<br>● fraud_score<br>Amendments to “iFrame” section. |
| 2015-10-02 | Added “Payment with card token” section |
| 2015-09-23 | Added “iFrame to parent communication” section and link to JavaScript library |
| 2015-07-24 | Added hmac_fields field |

| | |
|---|---|
| 2015-07-21 | Added iFrame integrated payment form feature |
| 2015-07-10 | Added token payment parameters:<br>● cc_token<br>● request_cc_token<br>● cc_last_four_digits |
| 2015-06-01 | Initial document |