
paymenTechnologies

Sandbox Technical Documentation
Credit Card Interface Specification
Version 1.0.1
Last update: December 02, 2021

1. INTRODUCTION

This document illustrates the paymenTechnologies interface. The interface uses the HTTPS protocol. The paymenTechnologies gateway receives a HTTPS request. The parameters are transmitted in the content using the POST method. Special characters must be URL encoded (e.g. space characters are represented by %20 for example). Only the content, not the complete string, must be encoded otherwise the "&"'s and "="'s would be encoded as well and confuse our gateway. For additional information see http://www.w3c.org.

The paymenTechnologies gateway is easy to implement but you need knowledge of at least one programming language. Pure HTML knowledge is not enough to implement paymenTechnologies your system.

2. THE AUTHORIZE INTERFACE

The Authorize request will send an authorization request to the authorization system, which will verify the credit card data and credit line. If the request is verified, the credit card will be charged (in real-time) immediately.

Interface URL: https://sandbox-api.paymentechnologies.co.uk/v2/authorize

The authorize interface requires the following transaction-specific fields:

Field Name	Type	Length	Format	Mandatory?	Description
authenticate_id	Alphanumeric	32 characters		yes	merchant identification
authenticate_pw	Alphanumeric	32 characters		yes	merchant identification
orderid	alphanumeric	max. 30 characters		yes	orderid in the merchant shop system
transaction_type	1 byte alphabets		A=Authorization	yes	type of transaction
signature	hex		SHA-1 hash value (hexadecimal)	yes	checksum for validation of request
amount	float numeric	8.2 8	12345678.90 12345678	yes	transaction amount
currency	char	3	ISO 4217 i.e. "EUR" or "USD"	yes	
card_info	alphanumeric	Defined by the Credit Card Parameter table below. You must encrypt the credit card information before submitting to paymenTechnologies Gateway. ANNEX C has more information regarding the Credit Card Encryption procedure.			
email	RFC 822	max. 50 characters		yes	e-mail address
street	alphanumeric	max. 100 characters		yes	street
city	alphanumeric	max. 40 characters		yes	city

zip	alphanumeric	max. 10 characters		yes	postal code
state	char	2	customer's 2-character State code	yes	state/province
country	char	3	country (ISO 3166 alpha3)	yes	country
phone	alphanumeric	max. 15 characters		yes	customers phone number
transaction_hash	alphanumeric			yes	verification hash
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	yes	customer IP (IPv4)

Credit Card Parameters:

Field Name	Type	Length	Format	Mandatory?	Description
ccn	numeric	Max. 16 digits		yes	credit or debit card number
exp_month	numeric	2 digits		yes	valid through: month
exp_year	numeric	2 digits		yes	valid through: year
cvc_code	numeric	max. 4 digits		yes	card validation code
firstname	alphanumeric	max. 30 characters		yes	first name
lastname	alphanumeric	max. 30 characters		yes	last name

The fields authenticate_id and authenticate_pw contain the client authentication. It is used to identify the client within the paymenTechnologies gateway. It is created when the client account is created. The authenticate_id and authenticate_pw will be credentialed to the client by the paymenTechnologies administration. If a request contains no or an invalid authenticate_id and authenticate_pw the paymenTechnologies system will instantly reject the request.

For added security and to protect the authenticity of your transaction, you MUST generate and include a transaction_hash. To generate a transaction_hash you must include the following script;

<script type="text/javascript" src="https://sandbox-api.paymentechnologies.co.uk/js?key=**YOUR PUBLIC KEY**&form=**FORM ID**"></script>

immediately after the closing </form> tag of the form that is collecting the card details of the customer's transaction. Also, in the script **YOUR PUBLIC KEY** should be replaced with your real Public Key found in your paymenTechnologies Back Office Panel accessible from your top main menu "PROFILE", then under "Credentials & Terms". The **FROM ID** should be replaced with the actual form id of the form that is collecting the card details of your customer's transaction. This script will generate the transaction_hash and will push that in the form as a hidden field automatically.

Example:

```
<form name="payment" id="payment_form" method="POST">

[any of your from elements here]

</form>
<script type="text/javascript" src="https://sandbox-
api.paymentechnologies.co.uk/js?key=1234abcd&form=payment_form"></script>
```

Request Example

```
array(
        'authenticate_id'=>'YOUR AUTHENTICATE ID',
        'authenticate_pw'=>'YOUR AUTHENTICATE PW',
        'orderid'=>'YOUR ORDER ID',
        'transaction_type'=>'a',
        'amount'=>'10.00',
        'currency'=>'USD',
        'card_info'=>'YOUR ENCRYPTED CARD DETAILS',
        'email'=>'CUSTOMER EMAIL',
        'street'=>'1600 Amphitheatre Parkway ',
        'city'=>'Mountain View',
        'zip'=>'94043',
        'state'=>'CA',
        'country'=>'USA',
        'phone'=>'YOUR VALIDE MOBILE NUMBER'
        'customerip'=>'NNN.NNN.NNN.NNN'
        'signature'=>'CHECKSUM OF YOUR REQUEST'
        'transaction_hash'=>'SCRIPT GENERATED TRANSACTION HASH'
);
```

Response Parameters

Responses are formatted in JSON and returned in real-time. The response consists of an status code, status and a payload.

The response contains the following fields:

Field Name	Type	Content	Description
code	numeric	code	paymenTechnologies response code
status	string	transaction status	failed, success, error
transaction_id	numeric	transaction_id	paymenTechnologies transaction identification
order_id	alphanumeric	order_id value	order_id in the client system
message	alphanumeric	empty or error message	error messages of the paymenTechnologies gateway or empty
details	alphanumeric	empty or error details	error details of the paymenTechnologies gateway or empty
amount	float/numeric	amount value	requested amount
currency	char	currency value	requested currency
descriptor	alphanumeric	descriptor value	your purchase as it will appear on your statement

Failed Response example


```
{
  "code":430,
  "status":"failed",
  "response":{
    "transaction_id":100000,
    "order_id":"200000",
    "message":"Payment failed",
    "details":"Error",
    "amount":"100.00",
    "currency":"USD",
    "descriptor":""
  }
}
```

Success Response example


```
{
  "code":200,
  "status":"success",
  "response":{
    "transaction_id":100000,
    "order_id":"200000",
    "message":"OK",
    "details":"",
    "amount":"100.00",
    "currency":"USD",
    "descriptor":"Transaction Descriptor"
  }
}
```


IMPORTANT NOTICE:
If you received following response


```
{
  "code":458,
  "status":"redirect-required",
  "message":"An ID verification is required to complete this transaction. Please contact
paymenTechnologies Support for assistance.",
  "redirect_url":"https://sandbox-
gentius.paymentechnologies.co.uk/verification/f30166eafe8fdc680d0ae62742de85c0",
}
```


means that you have to redirect the customer to the URL store in the value related to the key "redirect_url" for an ID verification. For more details please see **THE IDENTITY VERIFICATION PROCESS** section.

3. paymenTechnologies 3D-Secure API

What is 3D-Secure?

3D-Secure is a secure protocol designed to ensure enhanced security and strong authentication for consumers when they use their debit or credit cards for online purchases. It is called, depending on the card type, “MasterCard SecureCode”, “Verified by Visa” and in the case of American Express cards, “Safekey”. It is deployed at the point of transaction, and typically involves the customer being asked by their card issuing bank to enter a passcode or password to prove that they are the legitimate card holder. Card Issuing banks have different methods of generating and delivering these codes, so consumers need to contact their card issuing bank to find out how to register for 3D-Secure and when challenged, enter their passcode/password for their card, and not the passcode/password for their paymenTechnologies Account.

How does 3D-Secure work?

3D-Secure authentication is the secure and direct interaction between the card issuing bank and consumer, in which paymenTechnologies unable to ‘view’ the cardholders banking details. paymenTechnologies generates a secure session between the card issuing bank and the cardholder to verify that the consumer is the owner of the card that they are trying to add to their wallet. For Sellers that have deployed the paymenTechnologies branded checkout, there is nothing more to do - we deploy 3D-Secure when it is necessary to comply with the regulations. Once the consumer’s payment card is added to the wallet, there will be very few instances when the level of risk in the transaction is sufficiently high for us to require this higher level of verification. Exceptions will be when we believe that the risk in the transaction can be mitigated using 3D-Secure, and rather than declining the payment, we will process such transactions through the 3D-Secure systems to request that the card issuing bank authenticate the consumer is the real cardholder. One of the benefits of using paymenTechnologies checkout is that we can mostly differentiate good from bad transactions, and invoke the use of 3D-Secure when it is necessary, and minimize consumer disruption from over-use of the system. We are confident that this process should increase your business with more ‘good’ approved transactions.

3.1. THE 3D-Secure AUTHORIZE INTERFACE

The Authorize request will send an authorization request to the authorization system, which will verify the cardholder information. If the request is verified, request returns a redirect_url which the cardholder will be redirect to, once the transaction completes the cardholder will be redirected to the success_url or fail_url merchant submitted in the original request. Merchant notify_url will receive the paymenTechnologies response to update merchant transaction.

All requests to the transaction platform must be made using POST requests over HTTPS in UTF-8.

The initial request returns a redirect_url which the cardholder will be redirect to, once the transaction completes the cardholder will be redirected to the merchant supplied success_url or fail_url and an asynchronous request will be sent to the notify_url.

Transactions URL: https://sandbox-api.paymentechnologies.co.uk/v2/authorize-3dsv

The 3Ds authorize interface requires the following transaction-specific fields:

Field Name	Type	Length	Format	Mandatory?	Description
authenticate_id	alphanumeric	32 char		yes	merchant identification
authenticate_pw	alphanumeric	32 char		yes	merchant identification
orderid	alphanumeric	max. 30 char		yes	orderid in the merchant shop system
transaction_type	alphabets	1 byte	A=Authorization	yes	type of transaction
signature	hex		SHA-1 hash value (hexadecimal)	yes	checksum for validation of request
amount	float numeric	8.2 8	12345678.90 12345678	yes	transaction amount
currency	char	3	ISO 4217 i.e. “EUR” or “USD”	yes	
card_info	alphanumeric	Defined by the Credit Card Parameter table below. You must encrypt the credit card information before submitting to paymenTechnologies Gateway. ANNEX C has more information regarding the Credit Card Encryption procedure.			
email	RFC 822	max. 50 char		yes	e-mail address
street	alphanumeric			yes	street
city	alphanumeric	max. 40 char		yes	city
zip	alphanumeric	max. 10 char		yes	postal code
state	char	2	2 char state code	yes	state/province
country	char	3	country (ISO 3166 alpha3)	yes	country
phone	alphanumeric	max. 15 characters		yes	Customers phone number
dob	string	yyyy-mm-dd		yes	Date of birth of client attempting purchase
success_url	string	https://yoursite.com/success		yes	Redirect to success page after payment is successfull. Url must be encoded (UrlEncode)
fail_url	string	https://yoursite.com/failed		yes	Redirect to fail page after payment is failed. Url must be encoded (UrlEncode)
notify_url	string	https://yoursite.com/notify_url		yes	Receive point on your website for the payment information (UrlEncode)
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	yes	customer IP (IPv4)
transaction_hash	alphanumeric			yes	Verification hash

The field authenticate_id and authenticate_pw contains the merchant authentication. It is used to identify the merchant within the paymenTechnologies gateway. It is created when the merchant is created. The authenticate_id, authenticate_pw will be communicated to the merchant by the paymenTechnologies

administration. If a request contains no or an invalid authenticate_id, authenticate_pw the paymenTechnologies system will instantly reject the request.

For added security and to protect the authenticity of your transaction, you MUST generate and include a transaction_hash. To generate a transaction_hash you must include the following script;

<script type="text/javascript" src="https://sandbox-api.paymentechnologies.co.uk/js?key=**YOUR PUBLIC KEY**&form=**FORM ID**"></script>

immediately after the closing </form> tag of the form that is collecting the card details of the customer's transaction. Also, in the script **YOUR PUBLIC KEY** should be replaced with your real Public Key found in your paymenTechnologies Back Office Panel accessible from your top main menu "PROFILE", then under "Credentials & Terms". The **FROM ID** should be replaced with the actual form id of the form that is collecting the card details of your customer's transaction. This script will generate the transaction_hash and will push that in the form as a hidden field automatically.

Example:

<form name="payment" id="payment_form" method="POST">

[any of your from elements here]

</form>
<script type="text/javascript" src="https://sandbox-api.paymentechnologies.co.uk/js?key=1234abcd&form=payment_form"></script>

The signature is a checksum which helps paymenTechnologies to ensure the authenticity of the request, e.g. that it was actually sent by the merchant and was not tampered on the way from the merchant to our gateway. It contains a 40 characters long hexadecimal value. The value is computed from transaction-specific parameters and a "secret", according to the secure encryption algorithm. SHA-1. The "secret" will be given to you via email by the paymenTechnologies administration and must be kept secret. Please refer to the 3D-Secure Specific CALCULATION OF CHECKSUMS – SIGNATURE section of this document for an in-depth explanation of how to calculate the signature. The signature field is mandatory and must be transmitted to the system with each request.

The initial response contains the following fields:
Responses are formatted in JSON and returned in real-time. The response consists of an status code, status and a payload.

The initial response contains the following fields:

Field Name	Type	Content	Description
code	numeric	code	paymenTechnologies response code
status	string	status	redirect-required
transaction_id	numeric	transaction_id	paymenTechnologies transaction identification
order_id	alphanumeric	order_id value	order_id in the client system
message	alphanumeric	empty/error message	error messages of the paymenTechnologies gateway or empty
details	alphanumeric	empty/detailed message	detailed message of the paymenTechnologies gateway or empty
amount	float/numeric	amount value	requested amount
currency	char	currency value	requested currency
descriptor	alphanumeric	descriptor	your purchase as it will appear on your statement
redirect_url	string (255)	redirect_url	The cardholder must be redirected to the redirect_url in order to complete the 3DS authentication process. ***Only available for success request. Redirect must be done via the client's browser, retrieving the result via file_get_contents or curl won't work.***

Initial response example


```
{
  "code":200,
  "status":"redirect-required",
  "response":{
    "transaction_id":100000,
    "order_id":"200000",
    "message":"Redirect required to complete the transaction",
    "details":"",
    "amount":"100.00",
    "currency":"USD",
    "descriptor":"",
    "redirect_url":"https://sandbox-api.paymentechnologies.co.uk/v2/authorize-3dsv/payment/aaeea733e0b05d08a649edf572234ffd"
  }
}
```


The merchant has redirected the cardholder to the redirect_url in order to complete the 3DS authentication process. Merchant notify_url will receive the following notification after the 3DS verification process has been completed & cardholder was redirected to the merchant's supplied success_url or fail_url with merchant orderid. (i.e, https://yoursite.com/success.php?oid=1497496523)

The notification response contains the following fields:

Field Name	Type	Content	Description
code	numeric	code	paymenTechnologies response code
status	string	status	success or failed
transaction_id	numeric	transaction_id	paymenTechnologies transaction identification
amount	float/numeric	amount value	requested amount
currency	char	currency value	requested currency
orderid	alphanumeric	orderid value	orderid in the client system
descriptor	alphanumeric	descriptor	your purchase as it will appear on your statement
message	alphanumeric	empty/error message	error messages of the paymenTechnologies gateway or empty

Your Notification Page Response:

```
Array
(
    [code] => 200
    [status] => success
    [transaction_id] =>  100000
    [amount] => 100.00
    [currency] => USD
    [orderid] => 200000
    [descriptor] => descriptor
    [message] =>
)
```

4. THE REFUND INTERFACE

The Refund operation, as its name states, consists of returning the money already collected by a Payment transaction.

The amount collected can be refunded in total. You cannot refund amounts bigger than the one sent in the original transaction.

Interface URL: https://sandbox-api.paymentechnologies.co.uk/v2/refund

The Refund interface requires the following fields:

Field Name	Type	Length	Format	Mandatory?	Description
authenticate_id	alphanumeric	32 characters		yes	merchant identification
authenticate_pw	alphanumeric	32 characters		yes	merchant identification
signature	hex		SHA-1 hash value (hexadecimal)	yes	checksum for validation of request
transactionid	numeric			yes	identification of an paymenTechnologies transaction
amount	float/numeric	8.2 / 8	123.90 123	yes	amount
currency	char	3	ISO 4217	yes	Currency
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	yes	customer IP (IPv4)

The signature is a checksum which helps paymenTechnologies to ensure the authenticity of the request, e.g. that it was actually sent by the client and was not tampered on the way from the client to our gateway. It contains a 40 character long hexadecimal value. The value is computed from transaction-specific parameters and a "secret", according to the secure encryption algorithm, SHA-1. The "secret" will be given to you via email by the paymenTechnologies administration and must be kept secret. Please refer to Annex A for an in-depth explanation of how to calculate the signature. The signature field is mandatory and must be transmitted to the system with each request.

The response contains the following fields:

Field Name	Type	Content	Description
code	numeric	code	paymenTechnologies response code
status	string	status	success or failed
refundid	numeric	transaction_id	paymenTechnologies refund identification
errormessage	alphanumeric	empty or error message	error messages of the PaymenTechnologies gateway or empty
amount	float/numeric	amount value	requested amount
currency	char	currency value	requested currency
orderid	alphanumeric	orderid value	orderid in the client system

Example: Refund Request

Request:
All requests to the transaction platform must be made using POST requests over HTTPS in UTF-8.

URL: https://sandbox-api.paymentechnologies.co.uk/v2/refund

POST data:

amount=1.00&authenticate_id=authenticate_id&authenticate_pw=authenticate_pw&currency=USD&customerip=127.1.1.1&transaction_id=6906281&transaction_type=R&signature=67c9855e1f9403czc8a35r1d59f2d6c20c50c73d

Response:


```
{
  "code":200,
  "status":"success",
  "response":{
    "refundid":100000,
    "message":"Refund successfully",
    "amount":"1.00",
    "currency":"USD",
    "transactionid":"200001"
  }
}
```

5. THE HOSTED PAYMENT

All requests to the transaction platform must be made using POST requests over HTTPS in UTF-8.

POST URL: https://sandbox-api.paymentechnologies.co.uk/v2/hosted-request

The hosted payment interface requires the following transaction-specific fields:

Field Name	Type	Length	Format	Mandatory?	Description
authenticate_id	Alphanumeric	32 characters		yes	merchant identification
authenticate_pw	Alphanumeric	32 characters		yes	merchant identification
orderid	alphanumeric	max. 30 characters		yes	unique order number
amount	float numeric	8.2 8	12345678.90 12345678	yes	transaction amount
currency	char	3	ISO 4217 i.e.“EUR” or “USD”	yes	transaction currency
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	yes	customer IP (IPv4)
hash	String	Hash is a crucial parameter – used specifically to avoid any tampering during the transaction. For hash calculation See ANNEX D.			

The field authenticate_id and authenticate_pw contain the client authentication. It is used to identify the client within the paymenTechnologies gateway. It is created when the client is created. The authenticate_id, authenticate_pw will be communicated to the client by the paymenTechnologies administration. If a request contains no or an invalid authenticate_id, authenticate_pw the paymenTechnologies system will instantly reject the request.

Initial Success Response

The paymenTechnologies API will return the response in JSON format. As shown below.

```
{
  "code":200,
  "status":"redirect-required",
```

```
  "message":"",
  "redirect_url":"https://sandbox.paymentechnologies.co.uk/finish-
pay/1e3c8308304a8921871efb23ee991916",
}
```


Initial Failed Response


```
{
  "status":"error",
  "code":503,
  "message":"Service Unavailable"
  "details":"Service Unavailable. Please try again"
}
```


IMPORTANT NOTICE: if you get the code = 200 and status=redirect-required and redirect_url means that you have to redirect the customer to the URL store in the value related to the key "redirect_url".

After you redirect the customer to the URL provided in the redirect_url key and the customer finishes the transaction, we are going to display a success or failed message with a redirect option (depending on the transaction status) to the customer.

Be in mind, you will receive the following notification response with the transaction result once the transaction is successful. The notification will be a request POST in form-data format.

For setting up your Success, Failed and Notify URL please contact paymenTechnologies Support for assistance.

Notification Response


```
Array
(
   [transactionid] => 100000
   [status] => 1
   [errorcode] =>
   [errormessage] =>
   [amount] => 12.50
   [currency] => USD
   [orderid] => 200000
   [descriptor] => transaction descriptor
)
```

The notification response contains the following fields:

Field Name	Type	Content	Description
transactionid	numeric	transactionid	paymenTechnologies transaction identification for authorize and refund
status	numeric	status	1 = no error, 0 = error
errorcode	numeric	empty or error code	error code of the paymenTechnologies gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the paymenTechnologies gateway or empty
amount	float/numeric	amount value	requested amount
currency	char	currency value	requested currency
orderid	alphanumeric	orderid value	your orderid
descriptor	alphanumeric	descriptor value	your purchase will appear as on your statement

For your reference, please find sample code below which shows the basic set of parameters being posted.

Basic Request Params

Array
(
[authenticate_id] => 1234567c8aef906300afd0001cb112bb
[authenticate_pw] => 0cf12345678d38e1513dcc7e31234fdd
[orderid] => 1622102403
[amount] => 12.50
[currency] => USD
[customerip] => 127.0.0.1
)

For the above request the hash string will look like this:
1234567c8aef906300afd0001cb112bb|0cf12345678d38e1513dcc7e31234fdd|1622102403|12.50|USD|5e88e00ebb8d0.00099005

and the calculated hash value will be :
071d84dad64031085e06ef4a8e7c3949aa1399ce7ef4857472a7673f1d8e4a466bfc9b7c5e0fd586ca40c45acf389511f293f9ff084245dbeeb3f1e3c948dfac

Final Request with Hash Params

Array
(
[authenticate_id] => 1234567c8aef906300afd0001cb112bb
[authenticate_pw] => 0cf12345678d38e1513dcc7e31234fdd
[orderid] => 1622102403
[amount] => 12.50
[currency] => USD
[customerip] => 127.0.0.1
[hash] =>
071d84dad64031085e06ef4a8e7c3949aa1399ce7ef4857472a7673f1d8e4a466bfc9b7c5e0fd586ca40c45acf389511f293f9ff084245dbeeb3f1e3c948dfac
)

Sample cURL PHP Example

```
$data_stream = http_build_query(Final Request with Hash Params);
$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_stream);
curl_setopt($ch, CURLOPT_URL, POST URL);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
```

THE IDENTITY VERIFICATION PROCESS

For the security of your account we need to verify your customer identity. This automated process should take about 30 seconds to complete. An Identity Verification is required to complete the transaction if your account is set for Identity Verification or your CHARGEBACK ratio is over 0.375%.

Please communicate with your customer and have the following information ready:

- Government issued photo identification. (ie: Passport, Driver's License)
- Only JPG, JPEG formats are accepted
- The file size may not exceed 4MB.

After you redirect the customer to the URL provided in the redirect_url key and the customer finishes the ID verification process, we are going to display a success or failed message with a redirect option (depending on the transaction status) to the customer.

Be in mind, you will receive the following notification response with the transaction result once the transaction process. The notification will be a request POST in form-data format.

For setting up your Success, Failed and Notify URL please contact paymenTechnologies Support for assistance.

Be in mind, you will receive the following notification response with the transaction result once the Identity Verification completes and transaction gets processed. The notification will be a request POST in form-data format.

The notification response contains the following fields:

Field Name	Type	Content	Description
transactionid	numeric	transactionid	paymenTechnologies transaction identification for authorize and refund
status	numeric	status	1 = no error, 0 = error
errorcode	numeric	empty or error code	error code of the paymenTechnologies gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the paymenTechnologies gateway or empty
amount	float/numeric	amount value	requested amount
currency	char	currency value	requested currency
orderid	alphanumeric	orderid value	your orderid
descriptor	alphanumeric	descriptor value	your purchase will appear as on your statement

3D-Secure Specific CALCULATION OF CHECKSUMS – SIGNATURE

The signature parameter is a required automated calculation in your integration for every 3DS request, this ensures the origin of the request is actually coming from your integration and has not been tampered with.

IMPORTANT: Parameters must not be URL encoded except notify, success & failed URL before signature calculation.

TIP: Ideally the parameters would be available in an array/hash to make manipulation easier, and reduce code errors/repetition.

For each request, please follow these steps to build a signature string in your code:

1. Parameters must not be URL encoded except notify_url, success_url & fail_url
2. Sort parameters, by parameter name alphabetically. This is easily achieved if your parameters are stored in an array/hash or similar
3. Append/Concatenate/Implode, the parameter values together, according to the alphabetical sequence of parameter names
4. Append your secret to the end of the concatenated string
5. Calculate a SHA-1 hex value of the string. This hash value must be in lowercase letters

The “secret” is known only by you and the payment gateway. It must be exchanged by email.

If you have problems to calculate the correct signature, please check the following:

- The signature parameter must be in hexadecimal format.
- The hexadecimal string must be written in lower-case letters.
- Please make sure that the parameters are not URL encoded except notify, success & failed URL before signature calculation.
- Please check that all parameter values are included in the signature calculation.
- The secret must be appended to the SHA-1 function input string.

CRITICALLY IMPORTANT NOTE:
The calculation of the signature value must be done dynamically for every single request which you send from your system to the payment gateway. If the signature is not calculated properly per request, the gateway will respond with error code 418. Please remember that the gateway does not accept requests with empty or malformed signatures. Empty or malformed signatures will be declined.

ANNEX A: CALCULATION OF CHECKSUMS – SIGNATURE

The signature parameter is a required automated calculation in your integration for every request, this ensures the origin of the request is actually coming from your integration and has not been tampered with.

IMPORTANT: Parameters must not be URL encoded before signature calculation.

TIP: Ideally the parameters would be available in an array/hash to make manipulation easier, and reduce code errors/repetition.

For each request, please follow these steps to build a signature string in your code:

1. Parameters must not be URL encoded
2. Sort parameters, by parameter name alphabetically. This is easily achieved if your parameters are stored in an array/hash or something similar
3. Append/Concatenate/Implode, the parameter values together, according to the alphabetical sequence of parameter names
4. Append your secret to the end of the concatenated string
5. Calculate a SHA-1 hex value of the string. This hash value must be in lowercase letters

The “secret” is known only by you and the payment gateway. It must be exchanged by email.

If you have problems to calculate the correct signature, please check the following:

- The signature parameter must be in hexadecimal format.
- The hexadecimal string must be written in lower-case letters.
- Please make sure that the parameters are not URL encoded before signature calculation.
- Please check that all parameter values are included in the signature calculation.
- The secret must be appended to the SHA-1 function input string.

CRITICALLY IMPORTANT NOTE:
The calculation of the signature value must be done dynamically for every single request which you send from your system to the payment gateway. If the signature is not calculated properly per request, the gateway will respond with error code 418. Please remember that the gateway does not accept requests with empty or malformed signatures. Empty or malformed signatures will be declined.

ANNEX B: External Links

ISO list for country codes (Alpha 3)
- ✓ http://unstats.un.org/unsd/methods/m49/m49alpha.htm

ISO list for state codes (only for USA, Canada and Australia)
- ✓ http://www.sdms.org/statelist.asp

ISO list for currency codes
- ✓ http://www.xe.com/iso4217.php

SHA-1 Generator
- ✓ http://www.tech-faq.com/sha-1-generator
- ✓ http://www.sha1generator.de/

ANNEX C: Encrypt Credit Card Information as required by paymenTechnologies Gateway

Credit card information [Credit Card Number, CVC, Expiry Date, First Name, Last Name] must be encrypted before submitting data to paymenTechnologies Gateway. Merchant must use AES symmetric algorithm, block length 256 bits, CBC mode to encrypt credit card information. Merchant server support PHP < 7.0 with mcrypt library installed can also use RIJNDAEL-128 with CBC Mode to encrypt credit card information.

Steps to encrypt Credit Card Information,

- ✓ Encryption Key: You needs to create encryption key from SECRET KEY provided by paymenTechnologies Gateway. To make an encryption key, follow these steps:
  - Remove all non-alphanumeric characters from SECRET KEY.
  - Get first 16 characters (character position 0 to 16th) from filtered out SECRET KEY.
- ✓ Create IV
- ✓ Make card information a sting like
  ccn||4321450000000000__expire||05/25__cvc||111__firstname||Jhon__lastname||Smith

- ✓ Encrypt card information
- ✓ Make final string by concat encrypted data, **::** and IV like
  ENCRYPTED_DATA '**::**' IV
- ✓ BASE64 ENCODE the final string and add it as value for key card_info in POST Data

ANNEX D: Hashkey build

For hash calculation, you need to generate a string using certain parameters and apply the sha512 algorithm on this string. Please note that you have to use pipe (|) character in between these parameters as mentioned below. The parameter order is mentioned below:
sha512(authenticate_id|authenticate_pw|orderid|amount|currency|SECRET KEY)

All these parameters (and their descriptions) have already been mentioned earlier. Here, SECRET KEY (to be provided by paymenTechnologies)

sha512(authenticate_id|authenticate_pw|orderid|amount|currency|SECRET KEY)

For PHP example code to generate Hash:
$output = hash ("sha512", $text);

Test Data:

Transactions can be submitted using the following information:

Visa: 4001919257537193

MasterCard: 5425233430109903

Both you can set any expiry and cvv.

Triggering Errors in Sandbox Mode:

To simulate Declined (Card declined by issuer), pass card other then 4001919257537193

, 5425233430109903

To simulate a Payment failed, pass 999 in the cvv field.

paymenTechnologies Error Codes

Code	Error Message
399	Missing first/last name
400	Bad Request
401	Unauthorized
405	Method Not Allowed
406	Not Acceptable
418	Wrong signature calculation
419	Wrong currency
421	Invalid card
422	Wrong card
423	Unknown error
424	Wrong country
425	Invalid postcode/ZIP
426	Expiry month must be valid
427	Expiry year must be valid
428	Expiry Date must be valid
429	No gateway found for payment
430	Payment failed
431	Trying to refund more amount than captured
432	Invalid transaction id
433	Refund failed
434	Wrong state
435	Url must be encoded(urlencode)
436	Notification url not configured properly
437	Request must be HTTPS GET
438	Trying to process unapproved transaction
439	Wrong transactionid
440	Wrong transaction type

441	Invalid amount
442	Encryption / Decryption Failed
443	Successful Transaction exists with same order id
444	Missing cvc_code/Invalid
445	The transaction is not coming from an approved IP address
446	Unauthorized Domain
447	No Transaction Capability
448	Missing Transaction Hash
449	Card is blocked or Transaction Laundering (TL) Cards
450	We currently do not accept cards from your country.
451	The transaction attempted exceeds the Day per Card limit that has been established for your account.
452	The transaction attempted exceeds the Weekly per Card limit that has been established for your account.
453	The transaction attempted exceeds the Monthly per Card limit that has been established for your account.
454	The transaction attempted exceeds the total transaction amount allowed for this card for this month.
455	The transaction attempted exceeds the daily transaction amount allowed for this card.
456	The transaction attempted exceeds the maximum transaction amount allowed for this card.
459	Hash mismatch.
500	Internal Server Error.
502	Bad Gateway
505	HTTP Version Not Supported

**Please contact for the error that is not described on above since it may be a system error.