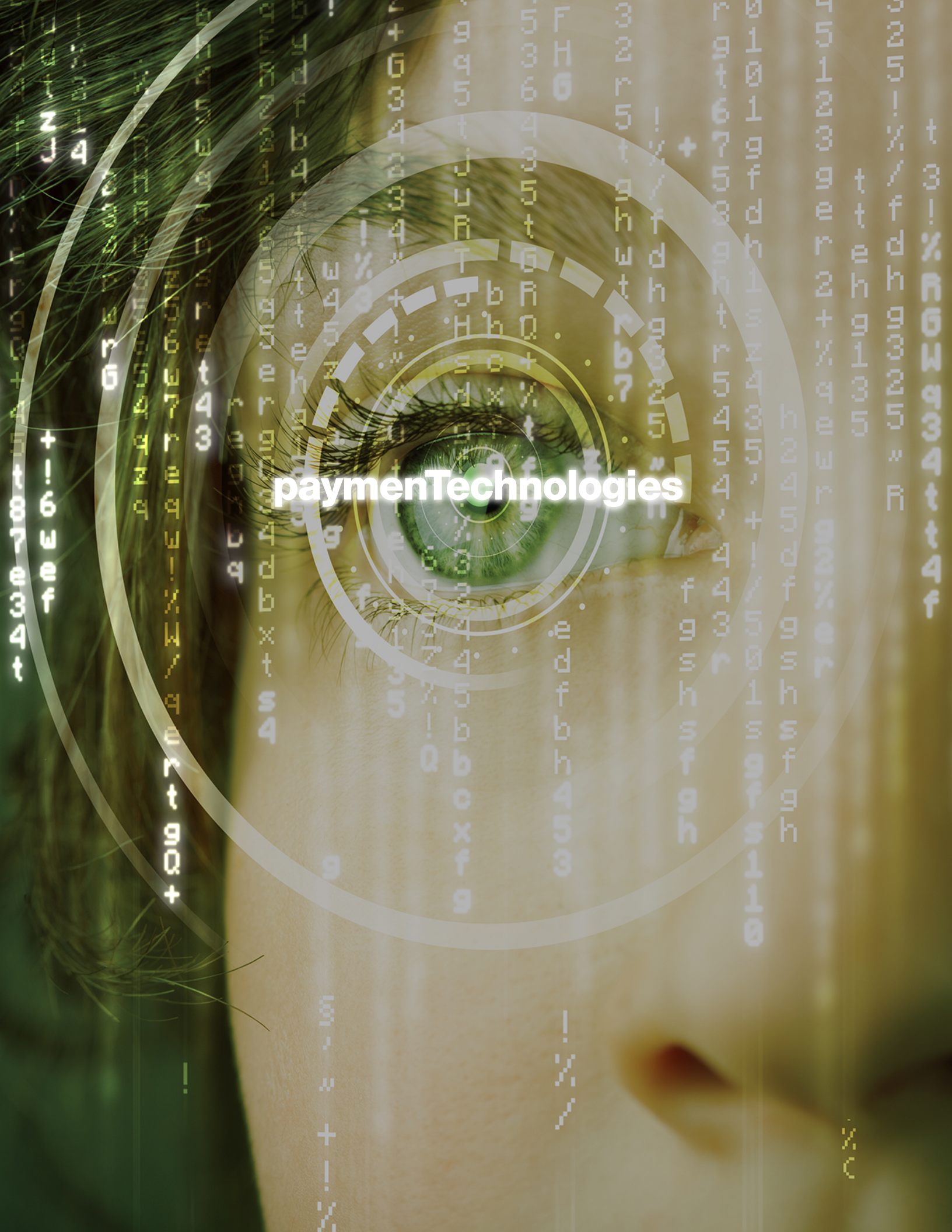
paymenTechnologies

Technical Documentation

Credit Card Interface Specification
Version 1.0.1
Last update: February 20, 2020

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 in your system.

2. OVERVIEW OF INTERFACES FOR PAYMENT TRANSACTIONS

Interface	Description
/authorize_payment	authorization of amount
/refund	credit note for captured amount

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. Please refer to Section 3 for a detailed description of required parameters.

The Refund request can only be followed by a successful authorize request. The Refund request will send a credit note to the authorization system after the customer's credit card has already been charged. The refund amount will be credited to the customer's credit card and the merchant's account will be charged. Please refer to Section 4 for a detailed description of required parameters.

3. THE AUTHORIZE INTERFACE

Interface URL: https://pay.paymentechnologies.co.uk/authorize_payment

The authorize interface requires the following transaction-specific fields:

Field Name	Type	Length	Format	Mandatory?	Description
authenticate_id	Alphanumeric	32 characters		✓	merchant identification
authenticate_pw	Alphanumeric	32 characters		✓	merchant identification
orderid	alphanumeric	max. 30 characters		✓	orderid in the merchant shop system
transaction_type	1 byte alphabets		A=Authorization	✓	type of transaction
signature	hex		SHA-1 hash value (hexadecimal)	✓	checksum for validation of request
amount	float numeric	8.2 8	12345678.90 12345678	✓	transaction amount
currency	char	3	ISO 4217 i.e. “EUR” or “USD”	✓	
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		✓	e-mail address
street	alphanumeric	max. 100 characters		✓	Street
city	alphanumeric	max. 40 characters		✓	City
zip	alphanumeric	max. 10 characters		✓	postal code
state	char	2	customer’s 2-character State code	✓	state/province
country	char	3	country (ISO 3166 alpha3)	✓	Country
phone	alphanumeric	max. 15 characters		✓	customers phone number
transaction_hash	alphanumeric			✓	verification hash
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	✓	customer IP (IPv4)

Credit Card Parameters:

Field Name	Type	Length	Format	Mandatory?	Description
ccn	numeric	Max. 16 digits		✓	credit or debit card number
exp_month	numeric	2 digits		✓	valid through: month
exp_year	numeric	2 digits		✓	valid through: year
cvc_code	numeric	max. 4 digits		✓	card validation code
firstname	alphanumeric	max. 30 characters		✓	first name
lastname	alphanumeric	max. 30 characters		✓	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://pay.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://pay.paymentechnologies.co.uk/js?key=1234abcd&form=payment_form"></script>

PaymenTechnologies API Integration Procedure for PHP:

For PHP enabled servers, the PaymenTechnologies gateway already has a PHP class to make integration faster. To integrate PaymenTechnologies gateway you need to include the supplied ***paymenTechnologies.php*** file to your script/website. Then you must use the following code sample to send payment request(s) to the PaymenTechnologies gateway.

```
// YOUR AUTHENTICATE ID and YOUR AUTHENTICATE PW and YOUR SECRET KEY
//must be replaced with your information.

// update the path of 'paymenTechnologies.php' as per your script/website structure
include_once paymenTechnologies.php';

// at the place of 'YOUR SECRET KEY' put your Secret Key given by PaymenTechnologies
$sec_key = 'YOUR SECRET KEY';

// Now develop the array to send request

$params = array(
        'authenticate_id'=>'YOUR AUTHENTICATE ID',
        'authenticate_pw'=>'YOUR AUTHENTICATE PW',
        'orderid'=>'YOUR ORDER ID',
        'transaction_type'=>'a',
        'amount'=>'10.00',
        'currency'=>'USD',
        'ccn'=>'4111111111111111',
        'exp_month'=>'12',
        'exp_year'=>'22',
        'cvc_code'=>'564',
        'firstname'=>'Jhon',
        'lastname'=>'Smith',
        'email'=>'jhonsmith@gmail.com',
        'street'=>'1600 Amphitheatre Parkway ',
        'city'=>'Mountain View',
        'zip'=>'94043',
        'state'=>'CA',
        'country'=>'USA',
        'phone'=>'YOUR VALIDE MOBILE NUMBER'
        'transaction_hash'=>'SCRIPT GENERATED TRANSACTION HASH'
        );
        // Now Making Final Payment request
        $pay = new paymenTechnologies($sec_key, $params, 'API');
        $response = $pay->payment();
```

```
// The next part is to handle the response from PaymenTechnologies
// You will receive json_encode response and you need to json_decode. Which is already done
// in the following code sample.

$data = json_decode($response);
if ($data->status == 1) {
        // Payment has been made successfully
        // assign response variables to local variables for further use
        $transactionid = $data->transactionid;
        $status = $data->status;
        $errorcode = $data->errorcode;
        $errormessage = $data->errormessage;
        $amount = $data->amount;
        $currency = $data->currency;
        $orderid = $data->orderid;
        $descriptor = $data->descriptor;
}else{
        // Payment failed
        // assign response variables to local variables for further use
        $status = $data->status;
        $errorcode = $data->errorcode;
        $errormessage = $data->errormessage;
}
```

The response contains the following fields:

Field Name	Type	Content	Description
transactionid	numeric	transactionid	PaymenTechnologies transaction identification for authorize, rebill 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
ordered	alphanumeric	orderid value	orderid in the client system
descriptor	alphanumeric	descriptor value	your purchase as it will appear on your statement

Example: Authorize Response

Request:
URL: https://pay.paymentechnologies.co.uk/authorize_payment

Response:
transactionid=500145927&status=1&errorcode=&errormessage=&amount=17.75&currency=USD&orderid=1505811500&descriptor=descriptor

IMPORTANT CUSTOMER NOTIFICATION NOTE:
The following notice in its entirety must appear on your final payment approval page. We also strongly suggest you include it on your payment and order confirmation emails as well. This will help you avoid chargebacks.

IMPORTANT: This purchase will appear as "**descriptor**" on your credit card statement or online transaction detail.

4. THE REFUND INTERFACE

Interface URL: https://pay.paymentechnologies.co.uk/refund

The Refund interface requires the following fields:

Field Name	Type	Length	Format	Mandatory?	Description
authenticate_id	alphanumeric	32 characters		✓	merchant identification
authenticate_pw	alphanumeric	32 characters		✓	merchant identification
transaction_type	1 byte alphabets		R=Refund	✓	type of transaction
signature	hex		SHA-1 hash value (hexadecimal)	✓	checksum for validation of request
transaction_id	numeric	max. 50 digits		✓	identification of an PaymenTechnologies transaction
amount	float/numeric	8.2 / 8	12345678.90 12345678	✓	amount
currency	char	3	ISO 4217	✓	Currency
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	✓	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
transactionid	numeric	transactionid	PaymenTechnologies transaction
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	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://pay.paymentechnologies.co.uk/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:
transactionid=2603271&status=1&errorcode=&errormessage=&amount=1.00&currency=USD

5. 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 PaymenTechnologiesis 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’s 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.

5.1. OVERVIEW OF INTERFACES FOR 3D-Secure TRANSACTIONS

Interface	Description
/authorize3dsv2	authorization of amount

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.

5.2. THE 3D-Secure AUTHORIZE INTERFACE

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://pay.paymentechnologies.co.uk/authorize3dsv2

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

Field Name	Type	Length	Format	Mandatory?	Description
authenticate_id	alphanumeric	32 characters		✓	merchant identification
authenticate_pw	alphanumeric	32 characters		✓	merchant identification
orderid	alphanumeric	max. 30 characters		✓	orderid in the merchant shop system
transaction_type	alphabets	1 byte	A=Authorization	✓	type of transaction
signature	hex		SHA-1 hash value (hexadecimal)	✓	checksum for validation of request
amount	float numeric	8.2 8	12345678.90 12345678	✓	transaction amount
currency	char	3	ISO 4217 i.e. “EUR” or “USD”	✓	
firstname	alphanumeric	max. 30 characters		✓	first name
lastname	alphanumeric	max. 30 characters		✓	last name
email	RFC 822	max. 50 characters		✓	e-mail address
street	alphanumeric	max. 100 characters		✓	street
city	alphanumeric	max. 40 characters		✓	city
zip	alphanumeric	max. 10 characters		✓	postal code
state	char	2	2 character State code	✓	state/province
country	char	3	country (ISO 3166 alpha3)	✓	country
phone	alphanumeric	max. 15 characters		✓	Customers phone number
dob	string	yyyy-mm-dd		✓	Date of birth of client attempting purchase
success_url	string	https://yoursite.com/success		✓	Redirect to success page after payment is successfull. Url must be encoded (UrlEncode)
fail_url	string	https://yoursite.com/failed		✓	Redirect to fail page after payment is failed. Url must be encoded (UrlEncode)

notify_url	string	https://yoursite.com/notify_url		✓	Receive point on your website for the payment information (UrlEncode)
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	✓	customer IP (IPv4)
transaction_hash	alphanumeric			✓	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://pay.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://pay.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:

Field Name	Type	Content	Description
status	numeric	status	1 = no error, 0 = error
transactionid	numeric	transactionid	PaymenTechnologies transaction identification for authorize and refund
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	orderid in the merchant system
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.***

IMPORTANT 3D-Secure IN-LINE (PAGE) NOTIFICATION NOTE:
The following notice in its entirety must appear on the same page of your 3D-Secure form.

IMPORTANT:
This 3D-Secure transaction will appear as "**descriptor**" on your credit card statement or online transaction detail.

IMPORTANT CUSTOMER NOTIFICATION NOTE:
The following notice in its entirety must appear on your final payment approval page. We also strongly suggest you include it on your payment and order confirmation emails as well. This will help you avoid chargebacks.

IMPORTANT: This purchase will appear as "**descriptor**" on your credit card statement or online transaction detail.

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
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	orderid in the merchant system
descriptor	alphanumeric	descriptor value	your purchase will appear as on your statement

Your Notification Page Response:

```
Array
(
   [transactionid] => 500109511
   [status] => 1
   [errorcode] =>
   [errormessage] =>
   [amount] => 1
   [currency] => USD
   [orderid] => 1497496523
   [descriptor] => descriptor18773937303
)
```

3D-Secure Specific CALCULATION OF CHECKSUMS – SIGNATURE

The signature parameter is a required automated calculation in your integration for every /authorize3ds 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.

Please see the following pseudo code example code:

```
// NOTE THIS CODE WILL NOT COMPILE, IT IS PSEUDO CODE
// WE HAVE PURPOSELY NOT OPTIMISED THIS CODE, SO IT IS AS SELF EXPLANATORY AS POSSIBLE

$signature = "";


// GENERATE THE SIGNATURE

// 1) Sort the parameters alphabetically (by key value)

key_sort($post);

// 2) Use a foreach to loop through the POST array.
foreach( $clean_post as $key => $val )
{
     // 3) Concatenate each value. Do not include the signature parameter.
     if( $key != "signature" )
     {
          $signature .= $val;
     }
}
//4)Append the secret.

$signature .= "YourSecretNumber";

// 5) Calculate SHA-1 checksum in lowercase characters.

$signature = lower_case( sha1( $signature ) );
```

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.
- Check the script calculating the signature. Please make the following test to make sure that your script is correctly working.

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.

Example: 3D-Secure Signature Calculation

Request parameters sorted in alphabetical order:

amount=17.17
authenticate_id=authenticate_id
authenticate_pw=authenticate_pw
city=Mountain View
country=USA
currency=USD
customerip=127.0.0.1
dob=1982-01-22
email=jhonsmith@gmail.com
fail_url=https%253A%252F%252Fyoursite.com%252Ffail.php
firstname=Jhon
lastname=Smith
notify_url=https%253A%252F%252Fyoursite.com%252Fnotify.php
orderid=1544678929
phone=16502530000
state=CA
street=1600 Amphitheatre Parkway
success_url=https%253A%252F%252Fyoursite.com%252Fsuccess.php
transaction_hash=582e8r35ed29dgbd8274kfcas56a3b2d
transaction_type=a
zip=94043

Input string for the SHA-1 function (not URL encoded except success, fail & notify URL Your Secret at the end of the string):

17.17authenticate_idauthenticate_pwMountain ViewUSAUSD127.0.0.11982-01-22jhonsmith@gmail.comhttps%3A%2F%2Fyoursite.com%2Ffail.phpJhonSmithhttps%3A%2F%2Fyoursite.com%2Fnotify.php1544678929165025300000CA1600 Amphitheatre Parkwayhttps%3A%2F%2Fyoursite.com%2Fsuccess.php582e8r35ed29dgbd8274kfcas56a3b2da94043YourSecretNumber

SHA-1 hash value of the string above:

a5ebf747b359e34f11f5a36ccaf341fcdc1ab57d

Final request with signature:

amount=17.17&authenticate_id=authenticate_id&authenticate_pw=authenticate_pw&city=Mountain+View&country=USA&currency=USD&customerip=127.0.0.1&dob=1982-01-22&email=jhonsmith%40gmail.com&fail_url=https%253A%252F%252Fyoursite.com%252Ffail.php&firstname=Jhon&lastname=Smith&notify_url=https%253A%252F%252Fyoursite.com%252Fnotify.php&orderid=1544678929&phone=16502530000&state=CA&street=1600+Amphitheatre+Parkway&success_url=https%253A%252F%252Fyoursite.com%252Fsuccess.php&transaction_hash=582e8r35ed29dgbd8274kfcas56a3b2d&transaction_type=a&zip=94043&signature=a5ebf747b359e34f11f5a36ccaf341fcdc1ab57d

6. PaymenTechnologies 3DS-Verified API

What is 3DS-Verified?

3DS-Verified 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.

6.1. OVERVIEW OF INTERFACES FOR 3DS-Verified TRANSACTIONS

The Authorize request will send an authorization request to the authorization system, which will verify the cardholder information. If the request is verified, the cardholder will be redirected to a page where card holder need to put transaction OTP. The cardholder will then be redirected to the Success/Fail URL you submitted in the original request. Your Notify URL will receive the PaymenTechnologies response you need to update your transaction.

6.2. THE 3DS-Verified AUTHORIZE INTERFACE

Transactions URL: https://pay.paymentechnologies.co.uk/authorize3dsv_payment

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

Field Name	Type	Length	Format	Mandatory?	Description
authenticate_id	alphanumeric	32 characters		✓	merchant identification
authenticate_pw	alphanumeric	32 characters		✓	merchant identification
orderid	alphanumeric	max. 30 characters		✓	orderid in the merchant shop system
transaction_type	1 byte alphabets		A=Authorization	✓	type of transaction
signature	hex		SHA-1 hash value (hexadecimal)	✓	checksum for validation of request
amount	float numeric	8.2 8	12345678.90 12345678	✓	transaction amount
currency	char	3	ISO 4217 i.e. “EUR” or “USD”	✓	
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		✓	e-mail address
street	alphanumeric	max. 100 characters		✓	street
city	alphanumeric	max. 40 characters		✓	city
zip	alphanumeric	max. 10 characters		✓	postal code
state	char	2	customer’s 2-character State code	✓	state/province
country	char	3	country (ISO 3166 alpha3)	✓	country
transaction_hash	alphanumeric			✓	Verification hash

phone	alphanumeric	max. 15 characters		✓	Customers phone number
dob	string	yyyy-mm-dd		✓	Date of birth of client attempting purchase
success_url	string	https://yoursite.com/success		✓	Redirect to success page after payment is successful. Url must be encoded (UrlEncode)
fail_url	string	https://yoursite.com/failed		✓	Redirect to fail page after payment is failed. Url must be encoded (UrlEncode)
notify_url	string	https://yoursite.com/notify_url		✓	Receive point on your website for the payment information (UrlEncode)

Credit Card Parameters:

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

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.

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://pay.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://pay.paymentechnologies.co.uk/js?key=1234abcd&form=payment_form "></script>

PaymenTechnologies 3DS-Verified Integration Procedure:

For PHP enabled servers, the PaymenTechnologies gateway already has a PHP class to make integration faster. To integrate PaymenTechnologies gateway you need to include the supplied ***paymenTechnologies.php*** file to your script/website. Then you must use the following code sample to send payment request(s) to the PaymenTechnologies gateway.

```
// YOUR AUTHENTICATE ID and YOUR AUTHENTICATE PW and YOUR SECRET KEY
//must be replaced with your information.

// update the path of 'paymenTechnologies.php' as per your script/website structure
include_once 'paymenTechnologies.php';

// at the place of 'YOUR SECRET KEY' put your Secret Key given by PaymenTechnologies
$sec_key = 'YOUR SECRET KEY';

// Now develop the array to send request

$params = array(
        'authenticate_id'=>'YOUR AUTHENTICATE ID',
        'authenticate_pw'=>'YOUR AUTHENTICATE PW',
        'orderid'=>'YOUR ORDER ID',
        'transaction_type'=>'a',
        'amount'=>'10.00',
        'currency'=>'USD',
        'ccn'=>'4111111111111111',
        'exp_month'=>'12',
        'exp_year'=>'17',
        'cvc_code'=>'564',
        'firstname'=>'Jhon',
        'lastname'=>'Smith',
        'email'=>'jhonsmith@gmail.com',
        'street'=>'1600 Amphitheatre Parkway ',
        'city'=>'Mountain View',
        'zip'=>'94043',
        'state'=>'CA',
        'country'=>'USA',
        'phone'=>'YOUR VALIDE MOBILE NUMBER',
        'transaction_hash'=>'SCRIPT GENERATED TRANSACTION HASH',
        'dob'=>'1982-01-22',
        'success_url'=>'https://yoursite.com/success.php',
        'fail_url'=>'https://yoursite.com/failed.php',
        'notify_url'=>'https://yoursite.com/notify.php',
        );
```

```
// Now Making Final Payment request
$pay = new paymenTechnologies($sec_key, $params, '3DSV');
$response = $pay->payment();

// The next part is to handles the response from PaymenTechnologies
// If the request is valid you will receive json_encode response and you need to json_decode.
// The response contains redirect URL where you have to redirect the cardholder. Which is
// already done in the following code sample.

$data = json_decode($response);
if ($data->status == 1) {
        // assign response variables to local variables for further use
        $status = $data->status;
        $redirect_url = $data->redirect_url;
        header('Location: '.$data->redirect_url);
}else{
        // something goes wrong
        $status = $data->status;
}
```

The cardholder is redirected to the transaction OTP page to complete the 3DSv authentication process. Merchant notify_url will receive the following notification response after 3DSv 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=1480939661)

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	orderid in the client system
descriptor	alphanumeric	descriptor value	your purchase will appear as on your statement

IMPORTANT 3DS-Verified IN-LINE (PAGE) NOTIFICATION NOTE:
The following notice in its entirety must appear on the same page of your 3DS-Verified form.

IMPORTANT:
This 3DS-Verified transaction will appear as "**descriptor**" on your credit card statement or online transaction detail.

IMPORTANT CUSTOMER NOTIFICATION NOTE:
The following notice in its entirety must appear on your final payment approval page. We also strongly suggest you include it on your payment and order confirmation emails as well. This will help you avoid chargebacks.

IMPORTANT: This purchase will appear as "**descriptor**" on your credit card statement or online transaction detail.

ANNEX A: CALCULATION OF CHECKSUMS – SIGNATURE

The signature parameter is a required automated calculation in your integration for every /pay 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.

Please see the following pseudo code example code:

```
// NOTE THIS CODE WILL NOT COMPILE, IT IS PSEUDO CODE
// WE HAVE PURPOSELY NOT OPTIMISED THIS CODE, SO IT IS AS SELF EXPLANATORY AS POSSIBLE

$signature = "";


// GENERATE THE SIGNATURE

// 1) Sort the parameters alphabetically (by key value)

key_sort($post);

// 2) Use a foreach to loop through the POST array.
foreach( $clean_post as $key => $val )
{
      // 3) Concatenate each value. Do not include the signature parameter.
      if( $key != "signature" )
      {
            $signature .= $val;
      }
}
//4)Append the secret.

$signature .= "YourSecretNumber";

// 5) Calculate SHA-1 checksum in lowercase characters.

$signature = lower_case( sha1( $signature ) );
```

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.
- Check the script calculating the signature. Please make the following test to make sure that your script is correctly working.

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

PaymenTechnologies Error Codes

Code	Error Message
399	Missing first/last name
400	Bad Request
401	Unauthorized
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
502	Bad Gateway

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