iCanPay

Technical Documentation

Credit Card Interface Specification
Version 3.1.3
Last update: July 30, 2018

1. INTRODUCTION

This document illustrates the iCanPay interface. The interface uses the HTTPS protocol. The iCanPay 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 iCanPay gateway is easy to implement but you need knowledge of at least one programming language. Pure HTML knowledge is not enough to implement iCanPay in your system.

2. OVERVIEW OF INTERFACES FOR PAYMENT TRANSACTIONS

Interface	Description
/pay/refund	credit note for captured amount
/pay/rebill	rebilling of a success transaction

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.

The Rebill 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 immediately. Please refer to Section 5 for a detailed description of required parameters.

3. THE AUTHORIZE INTERFACE

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
amount	float numeric	8.2 8	12345678.90 12345678	✓	transaction amount
currency	char	3	ISO 4217 i.e. “EUR” or “USD”	✓	
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
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

The fields authenticate_id and authenticate_pw contain the client authentication. It is used to identify the client within the iCanPay gateway. It is created when the client account is created. The authenticate_id and authenticate_pw will be credentialed to the client by the iCanPay administration. If a request contains no or an invalid authenticate_id and authenticate_pw the iCanPay 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.icanpay.cn.com/pay/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 iCanPay 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.icanpay.cn.com/pay/js?key=1234abcd&form=payment_form"></script>
```

iCanPay API Integration Procedure:

To integrate iCanPay gateway you need to include the supplied ***iCanPay.php*** file to your script/website. Then you must use the following code sample to send payment request(s) to the iCanPay Gateway.

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

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

// at the place of 'YOUR SECRET KEY' put your Secret Key given by iCanPay
$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'
        );

        // Now Making Final Payment request

        $pay = new iCanPay($sec_key, $params, 'API');

        $response = $pay->payment();
```

```
// The next part is to handle the response from iCanPay
// 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	iCanPay 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 iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay 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

Success Response:


```
{
        "transactionid":"500145927",
        "status":"1",
        "errorcode":"",
        "errormessage":"",
        "amount":"17.75",
        "currency":"USD",
        "orderid":"1505811500",
        "descriptor":"descriptor"
}
```


Failed Response:


```
{
        "status":"0",
        "errorcode":"401",
        "errormessage":"Unauthorized"
}
```


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.icanpay.cn.com/pay/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 iCanPay 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 iCanPay 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 iCanPay 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	iCanPay transaction
status	numeric	Status	1 = no error, 0 = error
errorcode	numeric	empty or error code	error code of the iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay 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.icanpay.cn.com/pay/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. THE REBILL INTERFACE

Interface URL: https://pay.icanpay.cn.com/pay/rebill

The Rebill 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
orderid	alphanumeric	max. 30 characters		✓	orderid in the merchant shop system
transactionid	numeric	max. 50 digits		✓	identification of an iCanPay transaction
transaction_type	2 bytes alphabets		RB=Rebill	✓	type of transaction
amount	float/numeric	8.2 / 8	12345678.90 12345678	✓	amount
currency	char	3	ISO 4217	✓	currency
signature	hex		SHA-1 hash value (hexadecimal)	✓	checksum for validation of request
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	✓	customer IP (IPv4)

The signature is a checksum which helps iCanPay 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 iCanPay 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 rebill response contains the following fields:

Field Name	Type	Content	Description
transactionid	numeric	transactionid	iCanPay 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 iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay 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

Example: Rebill Request

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

URL: https://pay.icanpay.cn.com/pay/rebill

POST data:
amount=52&authenticate_id=8d15a6699085555543b8bbf4442be83b&authenticate_pw=227f76d965554116747juh466598f44f3¤cy=USD&customerip=127.0.0.1&orderid=1496124951&transaction_type=RB&transactionid=&signature=99fb4e6f84e2a41271ec6c8398660f673f9fcd55

Response:
transactionid=1496125130&status=1&errorcode=&errormessage=&amount=52.00&currency=USD&orderid=1496124951&descriptor=some text

6. iCanPay 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 iCanPay Account.

How does 3D-Secure work?

3D-Secure authentication is the secure and direct interaction between the card issuing bank and consumer, in which iCanPay is unable to ‘view’ the cardholders banking details. iCanPay 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 iCanPay 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 iCanPay’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.

6.1. OVERVIEW OF INTERFACES FOR 3D-Secure TRANSACTIONS

Interface	Description
/pay/authorize3ds	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 iCanPay response to update merchant transaction.

6.2. 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.icanpay.cn.com/pay/authorize3ds

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	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”	✓	
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
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
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)

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

The signature is a checksum which helps iCanPay 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 iCanPay 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	iCanPay transaction identification for authorize and refund
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.

Example: Authorize Request

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

URL: https://pay.icanpay.cn.com/pay/authorize3ds

POST data:
amount=10&authenticate_id=authenticate_id&authenticate_pw=authenticate_pw&ccn=4242424242424242&city=Mountain+View&country=USA&currency =USD&customerip=127.0.0.1&cvc_code=657&dob=1982-01-22&email=jhonsmith.t%40gmail.com&exp_month=12&exp_year=17&fail_url=https%253A%252F%252Fyoursite.com%252Ffail.php&firstname=Jhon&lastname=Smith&notify_url =https%253A%252F%252Fyoursite.com%252Fnotify.php&orderid=1497496523&phone=16502530000&state=CA&street=1600+Amphitheatre+Parkway&success_url=https%253A%252F%252Fyoursite.com%252Fsuccess.php&transaction_type=a&zip=94043&signature=521a2b35851a46bbec99bb7cd2a924d50e65b522

Response:
status=1&transactionid=159753456258&amount=1.00&currency=USD&orderid=1497496523&redirect_url=http://3dsecure.com/5566589658785956852245

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	iCanPay transaction identification for authorize and refund
status	numeric	status	1 = no error, 0 = error
errorcode	numeric	empty or error code	error code of the iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay 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:

amount=10&authenticate_id=authenticate_id&authenticate_pw=authenticate_pw&ccn=4242424242424242&city=Mountain+View&country=USA&currency =USD&customerip=127.0.0.1&cvc_code=657&dob=1982-01-22&email=jhonsmith.t%40gmail.com&exp_month=12&exp_year=17&fail_url=https%253A%252F%252Fyoursite.com%252Ffail.php&firstname=Jhon&lastname=Smith&notify_url=https%253A%252F%252Fyoursite.com%252Fnotify.php&orderid=1497496523&phone=16502530000&state=CA&street=1600+Amphitheatre+Parkway&success_url=https%253A%252F%252Fyoursite.com%252Fsuccess.php&transaction_type=a&zip=94043&signature=521a2b35851a46bbec99bb7cd2a924d50e65b522

Parameters sorted in alphabetical order:

amount=20
authenticate_id=authenticate_id
authenticate_pw=authenticate_pw
ccn= 4242424242424242
city=Mountain View
country=USA
currency=USD
customerip=127.0.0.1
cvc_code=657
dob=1982-01-22
email=jhonsmith@gmail.com
exp_month=12
exp_year=17
fail_url= https://yoursite.com/fail.php
firstname=Jhon
lastname=Smith
notify_url=https://yoursite.com/notify.php
orderid= 1497496523
phone=16502530000
state=CA
street=1600 Amphitheatre Parkway
success_url= https://yoursite.com/success
transaction_type=a
zip=94043

Input string for the SHA-1 function (not URL encoded except notify, success & fail url, Your Secret at the end of the string):
10authenticate_idauthenticate_pw4242424242424242Mountain ViewUSAUSD127.0.0.16571982-01-22jhonsmith.t@gmail.com1217https%3A%2F%2Fyoursite.com%2Ffail.phpJhonSmithhttps%3A%2F%2Fyoursite.com%2Fnotify.php14974965231650253000 0CA1600 Amphitheatre Parkwayhttps%3A%2F%2Fyoursite.com%2Fsuccess.phpa94043YourSecretNumber

SHA-1 hash value of the string above:

521a2b35851a46bbec99bb7cd2a924d50e65b522

7. iCanPay Incognito API

iCanPay incognito allows you to integrate your iCanPay payments with your website's back-end operations, so you get immediate notification and authentication of the iCanPay payments you receive.

How does Incognito work?
You create an Instant Payment Notification (IPN) listener page on your website and then specify the URL of the listener page in their iCanPay account profile. iCanPay then sends notifications of all transaction-related events to that URL. When customers pay for goods or services, iCanPay sends a secure FORM POST containing payment information (IPN messages) to the URL. The IPN listener detects and processes IPN messages using your backend processes.

7.1. OVERVIEW OF INTERFACES FOR Incognito TRANSACTIONS

Interface	Description
/payment	authorization of amount

The Authorize request will send an authorization request to the authorization system, which will verify the order information. If the request is verified, the customer will be redirected to a page where customer need to login and place the order and then the customer will be redirected to the ReturnUrl you submitted in the orginal request. You must define your IPN URL into your iCanPay account to receive the iCanPay response you need to update your transaction.

7.2 THE Incognito AUTHORIZE INTERFACE

Submit parameters via a server-to-server HTTPS GET / POST to URL:
https://incognito.icanpay.cn.com/payment

The Incognito authorize interface requires the following transaction-specific fields:

Field Name	Type	Length	Format	Mandatory?	Description
IMID	alphanumeric	32 characters		✓	incognito identification
item	string			✓	Item details
orderid	alphanumeric	max. 30 characters		✓	orderid in the merchant shop system
quantity	numeric			✓	order quantity
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"	✓	
email	RFC 822	max. 50 characters		✓	e-mail address
success_url	string	https://yoursite.com/success		✓	Redirect to success page after payment is success / failed
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	✓	customer IP (IPv4)

The field IMID contains the client authentication. It is used to identify the client within the iCanPay gateway. It is created when the client is created. The IMID will be communicated to the client by the iCanPay administration. If a request contains no or an invalid IMID the iCanPay system will instantly reject the request.

The signature is a checksum which helps iCanPay 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 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 iCanPay administration and must be kept secret. Please refer to the Incognito 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 response contains the following fields:

Field Name	Type	Content	Description
transactionid	numeric	transactionid	iCanPay transaction identification for authorize and refund
status	numeric	status	1 = no error, 0 = error
errorcode	numeric	empty or error code	error code of the iCanPay gateway or empty
errormessage	alphanumeric	empty or errormessage	error messages of the iCanPay 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 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.

Example: Authorize Request

Request:
Submit parameters via a server-to-server HTTPS GET / POST to URL:
https://incognito.icanpay.cn.com/payment

Your Final array with signature may look like:

```
Array
(
   [orderid] => 1481702811
   [item] =>27-inch iMac with Retina 5K display
   [quantity] => 1
   [amount] => 1400
   [currency] => USD
   [success_url] => https://yoursite.com/success
   [IMID] => 57777ec4b7773
   [email] => jhonsmith@gmail.com
   [customerip] => 127.0.0.1
   [signature] => a4e12c2871e13f0a7df765008eec78d55829bee4
)
```

And finally redirect your request to our server.

Example of final redirection:
https://incognito.icanpay.cn.com/payment?orderid=1481702811&item=27-inch+iMac+with+Retina+5K+display&quantity=1&amount=1400& currency =USD&success_url=https%3A%2F%2Fyoursite.com%2Fsuccess&IMID=57777ec4b7773&email=jhonsmith%40gmail.com&customerip=127.0.0.1&signature=a4e12c2871e13f0a7df765008eec78d55829bee4

Your IPN Notification Page Response:
transactionid=6906281&status=1&errorcode=&errormessage=&amount=1400.00&currency=USD&orderid= 1481702811&descriptor=some text

Incognito Specific CALCULATION OF CHECKSUMS – SIGNATURE

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

Signature calculation

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

Signature parameter composition: IMID.orderid.currency.amount.secret

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:

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 check that required 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. Please remember that the gateway does not accept requests with empty or malformed signatures. Empty or malformed signatures will be declined.

Example: Incognito Signature Calculation
Input string for the SHA-1 function (not URL encoded, Your Secret at the end of the string):
57777ec4b77731481702811USD140056600d5d7797f2.74293177

SHA-1 hash value of the string above:
a4e12c2871e13f0a7df765008eec78d55829bee4

8. iCanPay 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.

8.1. 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 his/her name and draw his/her signature. The cardholder will then be redirected to the Success/Fail URL you submitted in the original request. Your Notify URL will receive the iCanPay response you need to update your transaction.

8.2. THE 3DS-Verified AUTHORIZE INTERFACE

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
amount	float numeric	8.2 8	12345678.90 12345678	✓	transaction amount
currency	char	3	ISO 4217 i.e. “EUR” or “USD”	✓	
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
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)

The field authenticate_id and authenticate_pw contain the client authentication. It is used to identify the client within the iCanPay gateway. It is created when the client is created. The authenticate_id, authenticate_pw will be communicated to the client by the iCanPay administration. If a request contains no or an invalid authenticate_id, authenticate_pw the iCanPay 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.icanpay.cn.com/pay/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 iCanPay 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.icanpay.cn.com/pay/js?key=1234abcd&form=payment_form "></script>
```

iCanPay 3DS-Verified Integration Procedure:

To integrate iCanPay gateway you need to include the supplied ***iCanPay.php*** file to your script/website. Then you must use the following code sample to send payment request(s) to the iCanPay Gateway.

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

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

// at the place of 'YOUR SECRET KEY' put your Secret Key given by iCanPay
$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
$pay = new iCanPay($sec_key, $params, '3DSV');
$response = $pay->payment();

// The next part is to handles the response from iCanPay
// 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 signature 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	iCanPay transaction identification for authorize and refund
status	numeric	status	1 = no error, 0 = error
errorcode	numeric	empty or error code	error code of the iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay 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.

9. iCanPay Pre-Auth API

The pre-authorization 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	2 byte alphabets		PA=Pre-Authorization	✓	type of transaction
amount	float numeric	8.2 8	12345678.90 12345678	✓	transaction amount
currency	char	3	ISO 4217 i.e. “EUR” or “USD”	✓	
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
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

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

iCanPay Pre-Authorization Integration Procedure:

To integrate iCanPay gateway you need to include the supplied ***iCanPay.php*** file to your script/website. Then you must use the following code sample to send payment request(s) to the iCanPay Gateway.

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

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

// at the place of 'YOUR SECRET KEY' put your Secret Key given by iCanPay
$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'
        );

        // Now Making Final Payment request

        $pay = new iCanPay($sec_key, $params, ' PreAuth');

        $response = $pay->payment();
```

```
// The next part is to handles the response from iCanPay
// 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) {

        // Pre-authorization 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;
        $status_msg = $data->status_msg;
        $amount = $data->amount;
        $currency = $data->currency;
        $orderid = $data->orderid;
        $descriptor = $data->descriptor;
}else{

        // Pre-authorization 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	iCanPay transaction identification for pre-authorize
status	numeric	status	1 = no error, 0 = error
errorcode	numeric	empty or error code	error code of the iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay gateway or empty
status_msg	alphanumeric	status message	status messages of the iCanPay gateway if pre-auth created
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 will appear as on your statement

Example: Pre-Authorization Response

Success Response:


```
{
        "transactionid":"500145927",
        "status":"1",
        "errorcode":"",
        "errormessage":"",
        "status_msg":"approved",
        "amount":"17.75",
        "currency":"USD",
        "orderid":"1505811500",
        "descriptor":"descriptor"
}
```


Failed Response:


```
{
        "status":"0",
        "errorcode":"401",
        "errormessage":"Unauthorized"
}
```


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.

9.1 iCanPay Pre-Auth Capture API

Allows the application to complete the sale on a previously authorized (pre-auth) transaction.

Interface URL: https://pay.icanpay.cn.com/pay/pre_authorize/capture

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

The capture 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
transactionid	numeric	max. 50 digits		✓	identification of an iCanPay pre-auth transaction
signature	hex		SHA-1 hash value (hexadecimal)	✓	checksum for validation of request
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	✓	customer IP (IPv4)

The signature is a checksum which helps iCanPay 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 iCanPay 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 capture response contains the following fields:

Field Name	Type	Content	Description
transactionid	numeric	transactionid	iCanPay 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 iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay gateway or empty
status_msg	alphanumeric	status message	status messages of the iCanPay gateway if pre-auth captured success
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

Example: Pre-Auth Capture Request & Response

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

URL: https://pay.icanpay.cn.com/pay/pre_authorize/capture

POST data:
authenticate_id=8d15a6699085555543b8bbf4442be83b&authenticate_pw=227f76d965554116747juh466598f44f&customerip=127.0.0.1&transactionid=1496125130&signature=1d6a88d20703588ff9d47ecabe2060fca362d1c2

Success Response:
{"transactionid":"1496125130","status":"1","errorcode":"","errormessage":"","status_msg":"approved","amount":"17.75","currency":"USD","orderid":"1505811500","descriptor":"descriptor"}

Failed Response:
{"transactionid":"","status":"0","errorcode":"","errormessage":"Payment failed","amount":"17.75","currency":"USD","orderid":"1505811500","descriptor":"descriptor"}

9.2 iCanPay Pre-Auth Void API

Allows the application to void a previously authorized (pre-auth) transaction.

Interface URL: https://pay.icanpay.cn.com/pay/pre_authorize/void

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

The void 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
transactionid	numeric	max. 50 digits		✓	identification of an iCanPay pre-auth transaction
signature	hex		SHA-1 hash value (hexadecimal)	✓	checksum for validation of request
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	✓	customer IP (IPv4)

The signature is a checksum which helps iCanPay 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 iCanPay 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 void response contains the following fields:

Field Name	Type	Content	Description
transactionid	numeric	transactionid	iCanPay 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 iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay gateway or empty
status_msg	alphanumeric	status message	status messages of the iCanPay gateway if void success
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

Example: Pre-Auth Void Request & Response

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

URL: https://pay.icanpay.cn.com/pay/pre_authorize/void

POST data:
authenticate_id=8d15a6699085555543b8bbf4442be83b&authenticate_pw=227f76d965554116747juh466598f44f&customerip=127.0.0.1&transactionid=1496125130&signature=1d6a88d20703588ff9d47ecabe2060fca362d1c2

Success Response:
{"transactionid":"1496125130","status":"1","errorcode":"","errormessage":"","status_msg":"approved","amount":"17.75","currency":"USD","orderid":"1505811500","descriptor":"descriptor"}

Failed Response:
{"transactionid":"","status":"0","errorcode":"","errormessage":"Payment failed","amount":"17.75","currency":"USD","orderid":"1505811500","descriptor":"descriptor"}

9.3 iCanPay Pre-Auth Refund API

Allows the application to refund a previously authorized (pre-auth) captured transaction.

Interface URL: https://pay.icanpay.cn.com/pay/pre_authorize/refund

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

The refund 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
transactionid	numeric	max. 50 digits		✓	identification of an iCanPay pre-auth transaction
signature	hex		SHA-1 hash value (hexadecimal)	✓	checksum for validation of request
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	✓	customer IP (IPv4)

The signature is a checksum which helps iCanPay 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 iCanPay 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 refund response contains the following fields:

Field Name	Type	Content	Description
transactionid	numeric	transactionid	iCanPay 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 iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay gateway or empty
status_msg	alphanumeric	status message	status messages of the iCanPay gateway if refund success
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

Example: Pre-Auth Refund Request & Response

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

URL: https://pay.icanpay.cn.com/pay/pre_authorize/refund

POST data:
authenticate_id=8d15a6699085555543b8bbf4442be83b&authenticate_pw=227f76d965554116747juh466598f44f&customerip=127.0.0.1&transactionid=1496125130&signature=1d6a88d20703588ff9d47ecabe2060fca362d1c2

Success Response:
{"transactionid":"1496125130","status":"1","errorcode":"","errormessage":"","status_msg":"approved","amount":"17.75","currency":"USD","orderid":"1505811500","descriptor":"descriptor"}

Failed Response:
{"transactionid":"","status":"0","errorcode":"","errormessage":"Payment failed","amount":"17.75","currency":"USD","orderid":"1505811500","descriptor":"descriptor"}

10. iCanPay eCheck API

An eCheck is an electronic payment funded by the buyer's bank account. eChecks clear in 2 business days and require the same information as an ACH transaction to process. Capturing your sales data into your CRM? We can work with that! Our postback URL capabilities allow you to receive up-to-date information on all your sales.

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

Transactions URL: https://pay.icanpay.cn.com/pay/authorize_echeck

The eCheck 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		E=eCheck	✓	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"	✓	
routing_number	string	9-digit number		✓	Routing Number of customer. Must be 9-digit number it could start with any number include 0 e.g. 012345678
account_number	string	20 digits		✓	Account number of customer's bank. e.g. 00015948263
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	customer's 2 character State code	✓	state/province
country	char	3	country (ISO 3166 alpha3)	✓	country
phone	alphanumeric	max. 15 characters		✓	Customers phone number
postback_url	string	https://yoursite.com/postback_url		✓	Receive point on your website for the payment status information (UrlEncode)
customerip	alphanumeric	max. 15 characters	NNN.NNN.NNN.NNN	✓	customer IP (IPv4)

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

The signature is a checksum which helps iCanPay 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 iCanPay 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 initial response contains the following fields:

Field Name	Type	Content	Description
transactionid	numeric	transactionid	iCanPay transaction identification for authorize and refund
status	numeric	status	1 = no error, 0 = error
errorcode	numeric	empty or error code	error code of the iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay 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

Example: eChck Authorize Request & Response

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

Transactions URL: https://pay.icanpay.cn.com/pay/authorize_echeck

POST data:
account_number=1111&amount=15.50&authenticate_id=8d15a6699085555543b8bbf4442be83b&authenticate_pw=227f76d965554116747juh466598f44f&city=Mountain+View&country=USA&currency=USD&customerip=127.0.0.1&email=jhonsmitht%40gmail.com&firstname=Jhon&lastname=Smith&orderid=1523008040&phone=9998881234&postback_url=https%253A%252F%252Fyoursite.com%252Fpostback_url&routing_number=012345678&state=CA&street=1600+Amphitheatre+Parkway&transaction_type=e&zip=94043&signature=7d056acbcd212fbf536ccb1d29daebc414cfae62

Success Response:
transactionid=1496125130&status=1&errorcode=&errormessage=&amount=15.50&currency=USD&orderid=1523008040

Failed Response:
transactionid=&status=0&errorcode=430&errormessage=Payment Failed&amount=15.50&currency=USD&orderid=1523008040

The parameter POSTBACKURL allows you to receive real time status updates on your deposits. Status updates are made via a POST request to your postback URL. The postback endpoint must follow the following conditions:

1. Postback URL must be less than 200 characters.
2. Must be publicly accessible endpoint without authentication, such as Basic Auth.
3. HTTPS protocol is supported, but not required. However, if enabled, the certificate must be valid.

The postback response contains the following fields:

Field Name	Type	Content	Description
transactionid	numeric	transactionid	iCanPay transaction identification for authorize and refund
status	string	status	deposited=success, other then anything failed
errorcode	numeric	empty or error code	error code of the iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay 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

Your Notification Page Response:

```
Array
(
    [transactionid] => 1496125130
    [status] => deposited
    [errorcode] =>
    [errormessage] =>
    [amount] => 15.50
    [currency] => USD
    [orderid] => 1523008040
)
```

11. iCanPay qCARD API

The qCARD 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
amount	float numeric	8.2 8	12345678.90 12345678	✓	transaction amount
currency	char	3	ISO 4217 i.e. “EUR” or “USD”	✓	
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
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)

The field authenticate_id and authenticate_pw contain the client authentication. It is used to identify the client within the iCanPay gateway. It is created when the client is created. The authenticate_id, authenticate_pw will be communicated to the client by the iCanPay administration. If a request contains no or an invalid authenticate_id, authenticate_pw the iCanPay 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.icanpay.cn.com/pay/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 iCanPay 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.icanpay.cn.com/pay/js?key=1234abcd&form=payment_form"></script>
```

iCanPay qCARD Integration Procedure:

To integrate iCanPay gateway you need to include the supplied ***iCanPay.php*** file to your script/website. Then you must use the following code sample to send payment request(s) to the iCanPay Gateway.

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

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

// at the place of 'YOUR SECRET KEY' put your Secret Key given by iCanPay
$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
$pay = new iCanPay($sec_key, $params, ' BCP');
$response = $pay->payment();

// The next part is to handles the response from iCanPay
// 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;
        $errorcode = $data->errorcode;
        $errormessage = $data->errormessage;
}
```

The cardholder is redirected to the signature page to complete the qCARD authentication process. Merchant notify_url will receive the following notification response after qCARD 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	iCanPay transaction identification for authorize and refund
status	numeric	status	1 = no error, 0 = error
errorcode	numeric	empty or error code	error code of the iCanPay gateway or empty
errormessage	alphanumeric	empty or error message	error messages of the iCanPay 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 qCARD IN-LINE (PAGE) NOTIFICATION NOTE:
The following notice in its entirety must appear on the same page of your qCARD form.

IMPORTANT:
This qCARD 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/

iCanPay 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.