



Epoint.az

Electronic payment platform

version 1.0.3

Solution for payment on site

Contents

Epoint payment page working principle.....	3
Formation of API request.....	5
Formation of data and signature.....	5
Sending a request.....	7
Callback function processing.....	9
Checking payment status.....	10
Saving a card to make payments without entering card data.....	13
Executing a payment with a saved card.....	15
Saving the card for making payments without entering card data with the first payment made... 17	
Request for disbursement of funds.....	19
Cancel operations.....	21
Split payment request.....	21
Executing a split payment with a stored card.....	24
Saving the card for making payments without entering card data with the first split payment..	25
Formation of API preauth request.....	28
Apple Pay & Google Pay.....	32
Google Pay (integration for mobile applications).....	34
Wallets.....	34
Invoices.....	38
Heartbeat API.....	41
Bank Response Codes.....	41

Epoint system

provides possibility to connect payment acceptance to your site.

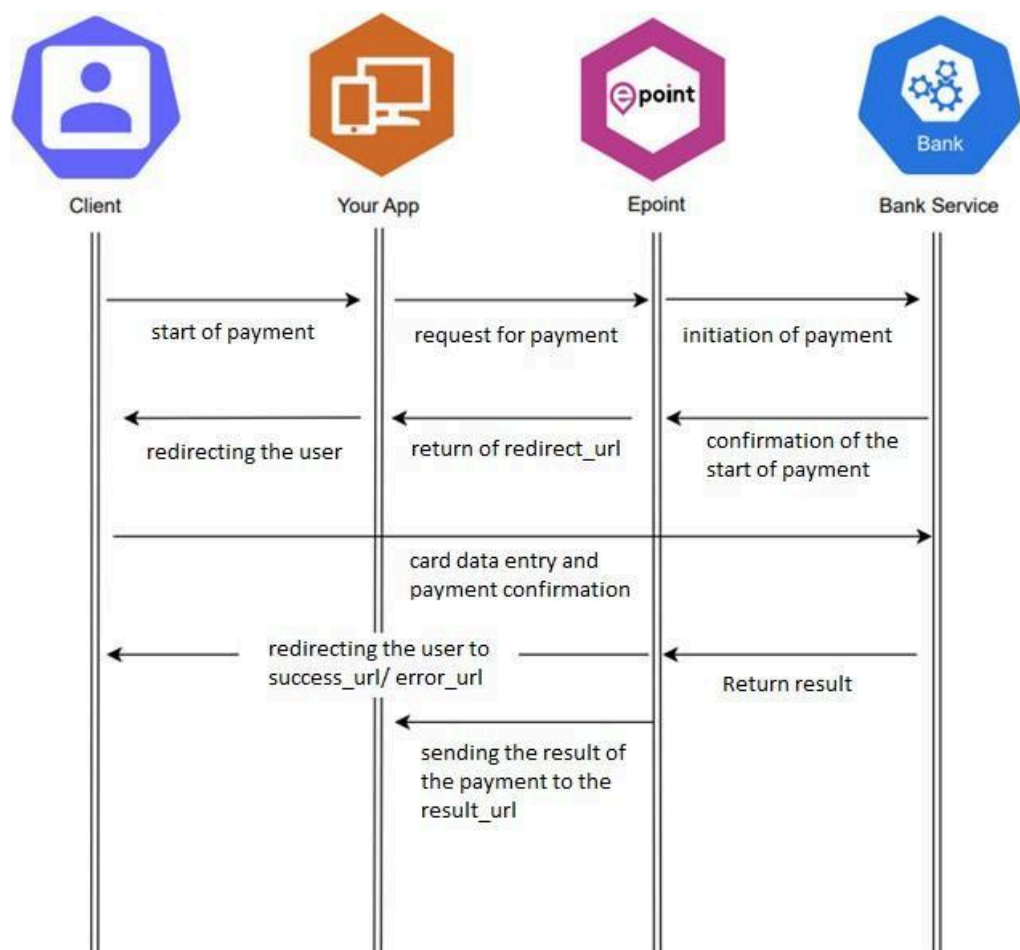
To add a payment button to your application you need to connect the payment service in your personal cabinet on our website.

To set up a merchant in our system you will need to provide us with the following information:

- your website address;
- url of the page of successful payment - `success_url`;
- url of the page for displaying information about unsuccessful payment - `error_url`;
- url to send the result of payment - `result_url`.

After checking this information, you will be given access keys: `public_key` - merchant ID in our system and `private_key` - secret API access key.

Epoint payment page working principle



1. It is necessary to form a request to the Epoint API according to the technical documentation
2. As a result of executing the request, the client will be redirected to the bank's payment page.
3. The client fills in the card details and confirms the payment.
4. In case of successful payment, the customer will be redirected to success_url or error_url -

otherwise.

5. The result of payment execution with payment details will be sent to the result_url specified by you.

Formation of API request

To call the Epoint API, you must send the data and signature parameters using the POST method to <https://epoint.az/api/1/request> or redirect the user using the POST method to <https://epoint.az/api/1/checkout>, where:

data - json string with API parameters encoded with base64 function,

base64_encode(json_string), signature -

unique signature for each request

base64_encode(sha1(**private_key** + data + **private_key**, 1)),

base64_encode - returns a string encoded in MIME base64 format,

sha1 - returns a hash of a string of 20 characters.

Formation of data and signature

Parameters json_string of the api call:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
amount	Required	Number	The amount of the payment. For example: 100, 20.50
currency	Required	String	Payment currency. Possible values: AZN
language	Required	String	Page display language. Possible values: az, en, ru.
order_id	Required	String	The unique ID of the transaction in your application. Maximum length 255 characters.
description	Optional	String	Payment Description. No more than

			1000 characters.
is_installment	Optional	Number	Parameter defining the payment type. Possible values: 1 (installment payment) or 0 (standard payment).
success_redirect_url	Optional	String	Redirection link in case of successful payment.
error_redirect_url	Optional	String	Redirection link in case of unsuccessful payment
other_attr	Optional	Array	Additional payment options

```
json_string =
{"public_key":"i000000001","amount":"30.75","currency":"AZN","description":"test payment","order_id":"1"}
```

json_string example:

Example data obtained as a result of json_string encoding with the function
base64_encode:

```
data =
```

```
eyJwdWJsaWNfa2V5IjoiaTAwMDAwMDAwMSIsImFtb3VudCI6IjMwLjc1IiwiY3VycmVuY3kiOiJBWk4iLCJkZXNjcmlwdGlvbiI6InRlc3QgcGF5bWVudCIsIm9yZGVyX2lkIjoimsSJ9
```

To form a signature, we need to concatenate private_key + data + private_key. For our case it will have the following form:

```
sgn_string =  
d3hjsl38sd8kdfhbcea0be04eafde9e8e2bad2fb092deyJwdWJsaWNfa2V5IjoiaTAwMDAwMDAwMSIsImFtb3VudCI6IjMwLjc1IiwiY3VycmVuY3kiOiJBWk4iLCJkZXNjcmlwdGlvbiI6InRlc3QgcGF5bWVudCIsIm9yZGVyX2lkIjoimsSJ9d3hjsl38sd8kdfhbcea0be04eafde9e8e2bad2fb092d
```

You have to apply `base64_encode(sha1(sgn_string, 1))` to the received string. The result will be the next string:

```
signature = a76GNudqblZtV8qF199hctA+cG0=
```

Sending a request

A form must be generated to send a request to the Epoint page:

```
<form method="POST" action="https://epoint.az/api/1/checkout" accept-charset="utf-8">  
  <input type="hidden" name="data"  
value="eyJwdWJsaWNfa2V5IjoiaTAwMDAwMDAwMSIsImFtb3VudCI6IjMwLjc1IiwiY3VycmVuY3kiOiJBWk4iLCJkZXNjcmlwdGlvbiI6InRlc3QgcGF5bWVudCIsIm9yZGVyX2lkIjoimsSJ9"/>  
  <input type="hidden" name="signature"  
value="a76GNudqblZtV8qF199hctA+cG0="/>  
  <input type="image" src="//epoint.az/images/wpb_ru.svg"/>  
</form>
```

Or send the received data and signature to url `https://epoint .az/api/1/request`:

```
// PHP code example  
$postfields = http_build_query(array(  
    'data' => $data,  
    'signature' => $signature  
));  
$ch = curl_init();  
curl_setopt($ch, CURLOPT_URL, $url);  
// curl options  
...  
// curl options  
curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields);  
  
$server_output = curl_exec($ch);
```

In this case a json string will be returned with the value of status (success|error), transaction and redirect_url to which user should be redirected to enter card data.

After entering the card data, the user will be redirected to success_redirect_url or

error_redirect_url depending on the payment status. Along with this, a POST request will be sent to result_url with payment details and transaction status.

Callback function processing

After the transaction is processed by Epoint service and the payment status is received from the bank, a POST request with two parameters data and signature will be sent to your server (result_url).

To authenticate a request from the Epoint server, you must:

1. Generate a signature on your server side using the data received in the response from Epoint and your private_key.

```
signature = base64_encode(sha1(private_key + data + private_key, 1))
```

2. The received signature should be compared with the one received from Epoint, if the signatures match, then you have received a genuine response from the Epoint server unmodified by a third party.

To decode the data value you must execute:

```
result = json_decode(base64_decode(data))
```

Use the Payment Status API function to get the status of a transaction, which can be done at any time.

Payment result parameters:

Parameter	Description
order_id	The unique ID of the transaction in your application.
status	Operation result success or failed
code	Bank response code
message	Payment execution status message
transaction	Epoint service transaction
bank_transaction	Bank payment transaction
bank_response	Bank's response, with the result of payment processing
operation_code	001 - card

	registration 100 - user payment
rrn	Retrieval Reference Number - a unique transaction identifier. Present only for a successful transaction
card_name	User name specified on the payment page
card_mask	User card mask in the format: 123456*****1234
amount	Payment amount
other_attr	Additional parameters

Checking payment status

To invoke Epoint payment status check, you need to pass the data and signature parameters by POST method to <https://epoint.az/api/1/get-status>, where:

data - json string with APIs parameters encoded by
function base64, base64_encode(json_string),
signature - unique signature for each request
base64_encode(sha1(**private_key** + data + **private_key**, 1).
json_string example:

```
json_string = {"public_key":"i000000001","transaction":"tw0000000101"}
```

Example data obtained as a result of encoding json_string with base64_encode function:

```
data = eyJwdWJsaWNfa2V5IjoiaTAwMDAwMDAwMSIsIm9yZGVyX2lkIjoxNX0=
```

To form a signature, we need to concatenate private_key + data + private_key. For our case it will have the following form:

```
sgn_string =  
d3hjsl38sd8kdfhbcea0be04eafde9e8e2bad2fb092deyJwdWJsaWNfa2V5IjoiaTAwMDAwMDAwM  
SIsIm9yZGVyX2lkIjoxNX0=d3hjsl38sd8kdfhbcea0be04eafde9e8e2bad2fb092d
```

Base64_encode(sha1(sgn_string, 1)) should be applied to the received string. The result will be the following string:

```
signature = bH9cG854p/wHLf5j6pp6LBI+wBs=
```

```
// PHP code example  
$postfields = http_build_query(array(  
    'data' => $data,  
    'signature' => $signature  
));  
$ch = curl_init();  
curl_setopt($ch, CURLOPT_URL, $url);  
// curl options  
...  
// curl options  
curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields);  
$server output = curl_exec($ch);
```

Параметры ответа:

Parameter	Description
-----------	-------------

status	Payment status
code	Bank response code
message	Payment execution status message
transaction	Epoint service transaction
bank_transaction	Bank payment transaction
bank_response	Bank's response, with the result of payment processing

operation_code	001 - card registration 100 - user payment
rrn	Retrieval Reference Number - a unique transaction identifier. Present only for a successful transaction
card_name	Username specified on the payment page
card_mask	User card mask in 123456*****1234 format
amount	Payment amount
other_attr	Additional parameters

Payment Statuses:

- new - payment is registered in the Epoint system ;
- success - successful payment;
- returned - the payment has been refunded;
- error - an error occurred during payment.
- server_error - status check execution error.

Saving a card to make payments without entering card data

To call Epoint API you need to send data and signature parameters by POST method to <https://epoint.az/api/1/card-registration>.

The json_string parameters of the api call:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
language	Required	String	Page display language. Possible values: az, en, ru.
refund	Optional	Number	Card type: 0 - debit card; 1 - payout card.
description	Optional	String	Payment Description. No more than 1000 characters.
success_redirect_url	Optional	String	Redirection link in case of successful payment.
error_redirect_url	Optional	String	Redirection link in case of unsuccessful payment

The request will return a json string with the status value (success|error) and redirect_url to which the user should be redirected to enter card data and card_id - a unique card identifier that will be used to make payments.

The client fills in the card information and confirms the payment.

In case of successful payment, the client will be redirected to success_url or error_url - otherwise (success_redirect_url and error_redirect_url, if specified).

After the transaction is processed by Epoint service and the payment status is received from the bank, a POST request with two parameters data and signature will be sent to the result_url specified by you.

To verify the authenticity of the request from Epoint server, you need to generate a signature on your server side, using the data and your private_key received in the response from Epoint.

```
signature = base64_encode(sha1(private_key + data + private_key,1))
```

The received signature should be compared with the one received from Epoint, if the signatures match, then you have received a genuine response from the Epoint server unmodified by a third

party.

To decode the data value you need to do:

```
result = json_decode(base64_decode(data))
```

Response Parameters:

Parameter	Description
status	Operation result success or failed
code	000 - successful operation, 500 is an error.
message	Operation progress status message
card_id	The unique card identifier that will be used to make the payment
bank_transaction	Bank payment transaction
bank_response	Bank's response, with the result of payment processing
operation_code	001 - card registration 100 - user payment
rrn	Retrieval Reference Number - a unique transaction identifier. Present only for a successful transaction
card_name	Username specified on the payment page
card_mask	User card mask in 123456*****1234 format

Executing a payment with a saved card

To make a payment with a stored card you need to send data and signature parameters by POST method to <https://epoint.az/api/1/execute-pay>.

Parameters json_string of API call:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
language	Required	String	Page display language. Possible values: az, en, ru.
card_id	Required	String	The card ID obtained by saving the card.
order_id	Required	String	Unique transaction ID in your application. The maximum length is 255 characters.
amount	Required	Number	Amount of payment. For example: 100, 20.50
currency	Required	String	Payment currency. Possible values: AZN
description	Optional	String	Description of the payment. Not more than 1000 characters.

After processing the transaction by the Epoint service and receiving the payment status from the bank, a response will be returned with the following parameters:

Parameter	Description
status	Result of a success or failed operation
transaction	Epoint Transaction ID.
bank_transaction	Bank payment transaction

bank_response	Bank response, with the result of payment processing
rrn	Retrieval Reference Number - Unique identifier of the transaction. Present only for successful transaction
card_name	User name specified on the payment page
card_mask	User card mask in 123456*****1234 format
amount	Amount of payment
message	Error message.

Saving the card for making payments without entering card data with the first payment made

If you use this type of payment, you will be paid for the specified amount along with the card registration. To call the Epoint API, you need to pass the data and signature POST parameters by the method to the address <https://epoint.az/api/1/card-registration-with-pay>

Parameters for the json_string call of the API:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
language	Required	String	Page display language. Possible values: az, en, ru.
order_id	Required	String	Unique transaction ID in your application. The maximum length is 255 characters.
amount	Required	Number	Amount of payment. For example: 100, 20.50
currency	Required	String	Payment currency. Possible values: AZN
description	Optional	String	Description of the payment. Not more than 1000 characters.
success_redirect_url	Optional	String	Redirection link in case of successful

			payment.
error_redirect_url	Optional	String	Redirection link in case of failed payment

As a result of the request, a json string with the status (success' error) value, transaction and redirect_url to which the user needs to be redirected to enter card data, will be returned, and card_id is the unique identifier of the card that will need to be used to make payments.

The customer fills in the card details and confirms payment. If the payment is successful, the client will be redirected to the success_url or error_url - otherwise (success_redirect_url and error_redirect_url if specified).

After processing the transaction by the Epoint service and receiving the payment status from the bank, a POST request with two data and signature parameters will be sent to result_url indicated by you.

To authenticate the request from the Epoint server, you must generate a signature on the side of your server using the Epoint data received in the response and your private_key.

```
signature = base64_encode(sha1(private_key + data + private_key,1))
```

The received signature must be compared with the one received from Epoint, if the signatures match, then you received a genuine response from the Epoint server unchanged by a third party.

To decode the data value, perform:

```
result = json_decode(base64_decode(data))
```

Response Options:

Parameter	Description
status	Result of a success or failed operation
code	000 - successful operation.
card_id	Unique identifier of the card you want to use to make the payment
order_id	Unique identifier of the payment in your application.
transaction	Epoint Transaction ID.
bank_transaction	Bank payment transaction
bank_response	Bank response, with the result of payment processing
operation_code	200 - card registration with the first payment
rrn	Retrieval Reference Number - Unique identifier of the transaction. Present only for successful transaction
card_mask	User card mask in 123456*****1234 format
card_name	Name of the cardholder.
amount	Amount of payment
other_attr	Advanced Options

Request for disbursement of funds

To check the payment of funds, you must send a POST request to the <https://epoint.az/api/1/refund-request> address with the data and signature parameters.

API call json_string parameters:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i0000000001
language	Required	String	Page display language. Possible values: az, en, ru.
card_id	Required	String	The card ID received by the card saving method.

order_id	Required	String	Unique transaction ID in your application. The maximum length is 255 characters.
amount	Required	Number	Amount of payment. For example: 100, 20.50
currency	Required	String	Payment currency. Possible values: AZN
description	Optional	String	Description of the payment. Not more than 1000 characters.

After processing the transaction by the Epoint service and receiving the payment status from the bank, a response will be returned with the following parameters:

Parameter	Description
status	Result of a success or failed operation
transaction	Epoint Transaction ID.
bank_transaction	Bank payment transaction
bank_response	Bank response, with the result of payment processing
rrn	Retrieval Reference Number - Unique identifier of the transaction. Present only for successful transaction
card_mask	User card mask in 123456*****1234 format
card_name	Name of the cardholder.
amount	Amount of payment
message	Error message.

Cancel operations

To cancel the operation, you must send a POST request to the <https://epoint.az/api/1/reverse> address with the data and signature parameters.

API call json_string parameters:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
language	Required	String	Page display language. Possible values: az, en, ru.
transaction	Required	String	Epoint Transaction ID.
amount	Optional	Number	Amount of payment. For example: 100, 20.50. You can specify a partial refund.
currency	Required	String	Payment currency. Possible values: AZN

After processing the transaction by the Epoint service and receiving the execution status from the bank, a response will be returned with the following parameters:

Parameter	Description
status	Result of a success or failed operation
message	Error message.

Split payment request

To create a split payment, you need to transfer the data and signature POST parameters by the method to the address <https://epoint.az/api/1/split-request>

Parameters for the json_string call of the API:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
amount	Required	Number	Amount of payment. For example, 100, 20.50
split_user	Required	String	The ID of the second user in the Epoint system.
split_amount	Required	Number	Payment amount for the second user. For

			example, 100, 20.50
currency	Required	String	Payment currency. Possible values: AZN
language	Required	String	Page display language. Possible values: az, en, ru.
order_id	Required	String	Unique transaction ID in your application. The maximum length is 255 characters.
description	Optional	String	Description of the payment. Not more than 1000 characters.
success_redirect_url	Optional	String	Redirection link in case of successful payment.
error_redirect_url	Optional	String	Redirection link in case of failed payment
other_attr	Optional	Array	Additional payment options

In this case, a json string will be returned with the status (success' error) value, transaction and redirect_url to which the user must be redirected to enter card data. After processing the transaction with the Epoint service and receiving payment status from the bank, a POST request with two data and signature parameters will be sent to your server (result_url).

To authenticate a request from an Epoint server, you must:

1. Generate a signature on the side of your server using the Epoint data received in the response and your private_key.

```
signature = base64_encode(sha1(private_key + data + private_key, 1))
```

2. The received signature must be compared with the one received from Epoint, if the signatures match, then you received a genuine response from the Epoint server unchanged by a third party.

To decode the data value, perform:

```
result = json_decode(base64_decode(data))
```

Please note that the amount paid for the second merchant will only appear on their payment list.

Payment Result Options:

Parameter	Description
order_id	Unique transaction ID in your application.
status	Result of a success or failed operation
code	Bank Response Code
message	Payment Status Message
transaction	Epoint Service Transaction
bank_transaction	Bank payment transaction
bank_response	Bank response, with the result of payment processing
operation_code	001 - card registration 100 - user payment
rrn	Retrieval Reference Number - Unique identifier of the transaction. Present only for successful transaction
card_name	User name specified on the payment page
card_mask	User card mask in 123456*****1234 format
amount	Amount of payment
split_amount	Payment amount for the second user.
other_attr	Additional parameters

Executing a split payment with a stored card

To pay with a saved card, you must send the data and signature POST parameters by the method to the address <https://epoint.az/api/1/split-execute-pay>.

API call json_string parameters:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001API call json_string parameters:
language	Required	String	Page display language. Possible values: az, en, ru.
card_id	Required	String	The card ID received by the card saving method.
order_id	Required	String	Unique transaction ID in your application. The maximum length is 255 characters.
amount	Required	Number	Amount of payment. For example: 100, 20.50
split_user	Required	String	The ID of the second user in the Epoint system.
split_amount	Required	Number	Payment amount for the second user. For example, 100, 20.50
currency	Required	String	Payment currency. Possible values: AZN

description	Optional	String	Description of the payment. Not more than 1000 characters.
-------------	----------	--------	--

After processing the transaction by the Epoint service and receiving the payment status from the bank, a response will be returned with the following parameters:

Parameter	Description
status	Result of a success or failed operation
transaction	Epoint Transaction ID.
bank_transaction	Bank payment transaction
bank_response	Bank response, with the result of payment processing
rrn	Retrieval Reference Number - Unique identifier of the transaction. Present only for successful transaction
card_mask	User card mask in 123456*****1234 format
card_name	User name specified on the payment page
amount	Amount of payment
message	Error message.
split_amount	Payment amount for the second user.

Saving the card for making payments without entering card data with the first split payment

If you use this type of payment, you will be paid for the specified amount along with the card registration. To call the Epoint API, you need to pass the data and signature POST parameters by the method to the address
<https://epoint.az/api/1/split-card-registration-with-pay>

Parameters for the json_string call of the API:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the

			created merchant. For example: i000000001
language	Required	String	Page display language. Possible values: az, en, ru.
order_id	Required	String	Unique transaction ID in your application. The maximum length is 255 characters.
amount	Required	Number	Amount of payment. For example: 100, 20.50
split_user	Required	String	The ID of the second user in the Epoint system.
split_amount	Required	Number	Payment amount for the second user. For example, 100, 20.50
currency	Required	String	Payment currency. Possible values: AZN
description	Optional	String	Description of the payment. Not more than 1000 characters.
success_redirect_url	Optional	String	Redirection link in case of successful payment.
error_redirect_url	Optional	String	Redirection link in case of failed payment

As a result of the request, a json string with the status (success' error) value, transaction and redirect_url to which the user needs to be redirected to enter card data, will be returned, and card_id

is the unique identifier of the card that will need to be used to make payments.

The customer fills in the card details and confirms payment.

If the payment is successful, the client will be redirected to the `success_url` or `error_url` - otherwise (`success_redirect_url` and `error_redirect_url` if specified).

After processing the transaction by the Epoint service and receiving the payment status from the bank, a POST request with two data and signature parameters will be sent to `result_url` indicated by you.

To authenticate the request from the Epoint server, you must generate a signature on the side of your server using the Epoint data received in the response and your `private_key`.

```
signature = base64_encode(sha1(private_key + data + private_key,1))
```

The received signature must be compared with the one received from Epoint, if the signatures match, then you received a genuine response from the Epoint server unchanged by a third party.

To decode the data value, perform:

```
result = json_decode(base64_decode(data))
```

Response Options:

Parameter	Description
status	Result of a success or failed operation
code	000 - successful operation.
card_id	Unique identifier of the card you want to use to make the payment
order_id	Unique identifier of the payment in your application.
transaction	Epoint Transaction ID.
bank_transaction	Bank payment transaction
bank_response	Bank response, with the result of payment processing

operation_code	200 - card registration with the first payment
rrn	Retrieval Reference Number - Unique identifier of the transaction. Present only for successful transaction
card_mask	User card mask in 123456*****1234 format
card_name	Name of the cardholder.
amount	Payment amount
split_amount	Payment amount for the second user.
other_attr	Advanced Options

Formation of API preauth request

To call the Epoint API, you must send the data and signature parameters using the POST method to <https://epoint.az/api/1/pre-auth-request> where:

data - json string with API parameters encoded with base64 function,

base64_encode(json_string),

signature - unique signature for each request

base64_encode(sha1(**private_key** + data + **private_key**, 1)),

base64_encode - returns a string encoded in MIME base64 format,

sha1 - returns a hash of a string of 20 characters.

Formation of data and signature

Parameters json_string of the api call:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
amount	Required	Number	The amount of the payment. For example: 100, 20.50
currency	Required	String	Payment currency. Possible values: AZN
language	Required	String	Page display language. Possible values: az, en, ru.
order_id	Required	String	The unique ID of the transaction in your application. Maximum length 255

			characters.
description	Optional	String	Payment Description. No more than 1000 characters.
success_redirect_url	Optional	String	Redirection link in case of successful payment.
error_redirect_url	Optional	String	Redirection link in case of unsuccessful payment
other_attr	Optional	Array	Additional payment options

json_string example:

```
json_string =
{"public_key":"i000000001","amount":"30.75","currency":"AZN","description":"test payment","order_id":"1"}
```

Example data obtained as a result of json_string encoding with the function
base64_encode:

```
data =
```

```
eyJwdWJsaWNfa2V5IjoiaTAwMDAwMDAwMSIsImFtb3VudCI6IjMwLjc1IiwiaY3VycmVuY3kiOiJBWk4iLCJkZXNjcmlwdGlvbiI6InRlc3QgcGF5bWVudCIsIm9yZGVyX2lkIjoiaMSJ9
```

To form a signature, we need to concatenate private_key + data + private_key. For our case it will have the following form:

```
sgn_string =  
d3hjsl38sd8kdfhbcea0be04eafde9e8e2bad2fb092deyJwdWJsaWNfa2V5IjoiaTAwMDAwMDAwMSIsImFtb3VudCI6IjMwLjc1IiwiaY3VycmVuY3kiOiJBWk4iLCJkZXNjcmlwdGlvbiI6InRlc3QgcGF5bWVudCIsIm9yZGVyX2lkIjoiaMSJ9d3hjsl38sd8kdfhbcea0be04eafde9e8e2bad2fb092d
```

You have to apply `base64_encode(sha1(sgn_string, 1))` to the received string. The result will be the next string:

```
signature = a76GNudqblZtV8qF199hctA+cG0=
```

Sending a preauth request

Send the received data and signature to url <https://epoint.az/api/1/pre-auth-request>:

```
// PHP code example  
$postfields = http_build_query(array(  
    'data' => $data,  
    'signature' => $signature  
));  
$ch = curl_init();  
curl_setopt($ch, CURLOPT_URL, $url);  
// curl options  
...  
// curl options  
curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields);  
$server output = curl_exec($ch);
```

In this case a json string will be returned with the value of status (success|error), transaction and redirect_url to which user should be redirected to enter card data.

After entering the card data, the user will be redirected to success_redirect_url or

error_redirect_url depending on the payment status. Along with this, a POST request will be sent to result_url with payment details and transaction status.

Complete preauth request

After the transaction is processed by Epoint service and the payment status is received from the bank, you should complete this payment otherwise it will not be added to your Epoint balance. Until this, it will be shown on your pending balance in the business panel.

Everything is the same when you want to complete the preauth request. Just body data will be different like in below.

Parameters json_string of the api call:

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
amount	Required	Number	The amount of the payment. For example: 100, 20.50
transaction	Required	String	Transaction id you got from epoint (see Callback function processing section) (looks like: te001111111)

To complete preauth request use endpoint below:

<https://epoint.az/api/1/pre-auth-complete>

Apple Pay & Google Pay

Google Pay: For web integration.

Apple Pay: For both web and app integration.

Create Widget Url (**POST** Request)

You need to pass the data and signature to the address <https://epoint.az/api/1/token/widget>.

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
amount	Required	Number	Amount of payment. For example: 100, 20.50
order_id	Required	String	Unique transaction ID in your application. The maximum length is 255 characters.
description	Required	String	Description of the payment. Not more than 1000 characters.

```
data = base64_encode(fields_in_json)
```

Fields in json

```
{
  "public_key": "your_public_key_on_epoint",
  "amount": 2.50,
  "order_id": "order id generated by your system",
  "description": "Test payment"
}
```

```
signature_string = private_key + data + private_key
```



```
signature = base64_encode(sha1(signature_string))
```

Signature string:

```
signature_string =  
d3hjsl38sd8kdfhbcea0be04eafde9e8e2bad2fb092deyJwdWJsaWNfa2V5ljoiaTAwMDAwMDA  
wMSIsImFtb3VudCI6IjMwLjc1IiwiaY3VycmVuY3kiOiJBWk4iLCJkZXNjcmlwdGlvbil6InRlc3QgcG  
F5bWVudCI6Im9yZGVyX2lkIjoiaMSJ9d3hjsl38sd8kdfhbcea0be04eafde9e8e2bad2fb092d
```

Example in php (laravel):

```
$payload = [  
    'public_key' => 'public_key',  
    'amount' => 2.50,  
    'order_id' => 'order id generated by your system',  
    'description' => 'Test payment',  
];  
  
$data = base64_encode(json_encode($payload));  
$private_key = 'your_private_key';  
$signature = base64_encode(sha1($private_key . $data . $private_key, 1));  
  
$request = Http::get("https://epoint.az/api/1/token/widget", [  
    'data' => $data,  
    'signature' => $signature  
]);  
  
$response = $request->json();
```

Response:

```
{  
    "status": "success",  
    "widget_url": "https://epointv1.test/api/1/token/widget/000001"
```

```
}
```

When payment is finished inside of the iframe or webview you can listen to the iframe message. You can see an example in the javascript code sample below:

```
window.addEventListener('message', function(event) {  
    console.log(event.data); // {status: 'success', payment: {...}}  
});
```

Google Pay (integration for mobile applications)

To integrate Google Pay into your mobile application, you must implement a native integration.

Integration Steps:

1. Implement native integration following the official Google Pay documentation.
<https://developers.google.com/pay/api/android/overview>
2. Provide us with screenshots showing:
 - The placement of the Google Pay button in your app interface.
 - The appearance of the Google Pay button.
3. After reviewing the submitted materials, we will provide you with a Merchant ID to complete the integration.

Wallets

The Wallet API provides the following endpoints:

1. <https://epoint.az/api/1/wallet/status> - to retrieve the list of wallets
2. <https://epoint.az/api/1/wallet/payment> - to create a payment using a wallet

To retrieve the list of wallets, the public_key parameter must be sent to <https://epoint.az/api/1/wallet/status> using the POST method.

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001

Request submission:

```
const fetchWallets = async () => {
  const url = "https://epoint.az/api/1/wallet/status";
  const body = {
    public_key: "your_public_key_on_epoint"
  };

  try {
    const response = await fetch(url, {
      method: "POST",
      headers: {
        "Content-Type": "application/json"
      },
      body: JSON.stringify(body)
    });

    const wallets = await response.json();
    console.log("Wallets:", wallets);

    // Dynamic button creation
    renderWalletButtons(wallets);
  } catch (error) {
    console.error("An error occurred:", error);
  }
};
```

Dynamic button creation and logic:

```

const renderWalletButtons = (wallets) => {
  const container = document.getElementById("wallet-buttons");
  container.innerHTML = ""; // Clear existing buttons

  for (const [id, name] of Object.entries(wallets)) {
    const button = document.createElement("button");
    button.innerText = name;
    button.onclick = () => startPayment(id);
    container.appendChild(button);
  }
};

const startPayment = (walletId) => {
  alert(`Selected wallet ID: ${walletId}`);
  // Additional operations to initiate the payment can be performed here
};

```

Predefined Structure in HTML:

```
<div id="wallet-buttons"></div>
```

To create a payment using a wallet, the data and signature POST parameters must be sent to <https://epoint.az/api/1/wallet/payment>.

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
wallet_id	Required	String	Selected wallet ID.
amount	Required	Number	Amount of payment. For example: 100, 20.50
currency	Required	String	Payment currency. Possible values: AZN
order_id	Required	String	Unique transaction ID in your application. The maximum length is 255 characters.
description	Optional	String	Description of the payment. Not more than 1000 characters.
language	Required	String	Page display language. Possible values: az, en, ru.

Request data preparation:

```
const createPaymentData = (walletId, amount, orderId, description) => {
  const data = {
    public_key: "your_public_key_on_epoint",
    wallet_id: walletId,
    amount: amount,
    currency: "AZN",
    order_id: orderId,
    description: description,
    language: "en"
  };

  return btoa(JSON.stringify(data));
  // Base64 encode the JSON
};

const createSignature = (privateKey, data) => {
  const signatureString = `${privateKey}${data}${privateKey}`;
  const hash = CryptoJS.SHA1(signatureString);
  // SHA1 calculation (using CryptoJS library)
  return btoa(hash.toString(CryptoJS.enc.Latin1));
  // Base64 encode the hash
};
```

Request Submission:

```

const sendPayment = async (walletId, amount, orderId, description) => {
  const url = "https://epoint.az/api/1/wallet/payment";
  const privateKey = "your_private_key_on_epoint";
  const data = createPaymentData(walletId, amount, orderId, description);
  const signature = createSignature(privateKey, data);

  const body = {
    data: data,
    signature: signature
  };

  try {
    const response = await fetch(url, {
      method: "POST",
      headers: {
        "Content-Type": "application/json"
      },
      body: JSON.stringify(body)
    });

    const result = await response.json();
    console.log("Ödənişin cavabı:", result);
  } catch (error) {
    console.error("Xəta baş verdi:", error);
  }
};

```

Invoices

To call the Invoice API, you need to pass the data and signature parameters using the POST method to the address <https://epoint.az/api/1>

The Invoice API has the following list of endpoints:

3. /invoices/create – creating an invoice
4. /invoices/update – updating invoice information
5. /invoices/view – viewing invoice information
6. /invoices/list – viewing information on all invoices
7. /invoices/send-sms – sending an SMS to the invoice recipient
8. /invoices/send-email - sending an email to the invoice recipient

Creating and updating an invoice has an additional optional parameter `invoice_images[]`, which contains images in jpg, png, jpeg, svg, bmp formats.

/invoices/create

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
sum	Required	Number	The amount of the payment. For example: 100, 20.50
display	Required	Number	Display invoice. Possible values: 1 or 0
save_as_template	Required	Number	Save invoice as template. Possible values: 1 or 0.
status_installment	Optional	Number	Enables installment payments. Possible values: 1 or 0.
name	Optional	String	Name
description	Optional	String	Payment Description. No more than 1000 characters.
phone	Optional	String	Phone.
email	Optional	String	Email.
inn	Optional	String	TIN.
contract_number	Optional	String	Contract number.
merchant_order_id	Optional	String	Order ID.
period_from	Required	Date	Invoice start date.
period_to	Required	Date	Invoice end date.

/invoices/update

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
id	Required	Number	ID of the invoice being updated.
sum	Required	Number	The amount of the payment. For example: 100, 20.50

display	Required	Number	Display invoice. Possible values: 1 or 0
save_as_template	Required	Number	Save invoice as template. Possible values: 1 or 0.
status_installment	Optional	Number	Enables installment payments. Possible values: 1 or 0.
name	Optional	String	Name
description	Optional	String	Payment Description. No more than 1000 characters.
phone	Optional	String	Phone.
email	Optional	String	Email.
inn	Optional	String	TIN.
contract_number	Optional	String	Contract number.
merchant_order_id	Optional	String	Order ID.
period_from	Required	Date	Invoice start date.
period_to	Required	Date	Invoice end date.

/invoices/view

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
id	Required	Number	Invoice ID.

/invoices/list

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
type	Optional	String	Invoice type - incoming, outgoing, static

order	Optional	String	Sorting by ascending, descending
-------	----------	--------	----------------------------------

/invoices/send-sms

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
id	Required	Number	Invoice ID.
phone	Required	String	Phone.

/invoices/send-email

Parameter	Required	Type	Description
public_key	Required	String	The public key is the identifier of the created merchant. For example: i000000001
id	Required	Number	Invoice ID.
email	Required	String	Email.

Heartbeat API

To check the service availability, send a GET request to the following endpoint:

<https://epoint.az/api/heartbeat>

This endpoint is designed to verify if the service is operational. A successful response with status: "ok" indicates that the service is running properly.

```
{
  "status": "ok"
}
```

Bank Response Codes

Code	Description
------	-------------

000	Confirmed
100	Rejected (general, no comment)

101	Declined, your card has expired
102	Rejected, suspected fraud
103	Rejected, cardholder will contact acquirer
104	Rejected, Restricted Card
105	Rejected, Card Receiver will contact Acquirer Security
106	Rejected, PIN attempts exceeded
107	Declined, please contact your card issuer
108	Declined, please refer to card issuer's special terms
109	Rejected, invalid merchant
110	Rejected, incorrect amount
111	Rejected, incorrect card number
112	Rejected, PIN required
113	Denied, inappropriate payment
114	Rejected, no account of the requested type
115	Denied, requested function is not supported
116	Declined, insufficient funds
117	Rejected, incorrect PIN
118	Rejected, no card data
119	Rejected, transaction not allowed by cardholder
120	Rejected, transaction not allowed to terminal
121	Declined, withdrawal limit exceeded
122	Rejected, Safety Violation
123	Declined, withdrawal limit exceeded
124	Rejected, violation of the law
125	Rejected, card not valid
126	Rejected, invalid PIN block
127	Rejected, PIN length error
128	Rejected, PIN key synchronization failed
129	Rejected, suspected fake card
180	Rejected, at the request of cardholders
200	Pick-up (general, no comment)
201	Pick-up, expired card

202	Pick-up, suspected fraud
203	Pick-up, the card receiver will contact the acquirer
204	Pick-up, restricted card

205	Pick-up, the cardholder will contact the acquirer's security department
206	Pick-up, PIN limit exceeded
207	Pick-up, Special Conditions
208	Pick-up, lost card
209	Pick-up, stolen card
210	Pick-up, suspected fake card
300	Status Message File Action Successful
301	Status Message File Action Not Supported by Recipient
302	Status Message: Could not find an entry in the file
303	Status Message Duplicate Record, Old Record Replaced
304	Status Message File Write Field Edit Error
305	Status Message File Locked
306	Status Message File Action Failed
307	Status Message File Data Format Error
308	Status Message Duplicate Record, New Record Rejected
309	Status Message: Unknown File
400	Accepted (for cancellation)
499	Confirmed, no original message data
500	Status message: agreed, in the balance sheet
501	Status message: agreed, out of balance
502	Status Message: Amount Not Agreed, Amount Provided
503	Status Message - Amount not available for negotiation
504	Status message: not agreed, amount provided
600	Accepted (for administrative information)
601	Status Message - Original Transaction Cannot Be Tracked
602	Status Message Invalid Transaction Reference Number
603	Status Message Link Number/PANs are Incompatible
604	Status message: POS photo not available
605	Status Message: Requested Item Provided
606	Status Message Request Failed - Required Documentation Unavailable

680	The list is ready
681	The list is not ready
700	Accepted (for payment collection)
800	Accepted (for network management)

900	The recommendation has been taken into account, no financial obligations have been accepted
901	Recommendations taken into account, financial liability accepted
902	Rejection Reason Message: Invalid Transaction
903	Status Message - Re-Enter Transaction
904	Rejection Reason Message: Format Error
905	Deviation Cause Message Received, Acquirer Not Supported by Switch
906	Reason for Deviation Report: Process Reduction
907	Reason for Rejection Message - Card Issuer or Switch Not in Effect
908	Rejection Reason Message: Unable to find the destination of the routing transaction
909	Cause of Deviation Report: System Failure
910	Reason for Rejection Message: Card Issuer Disabled
911	Reason for Rejection Message: Card Issuer Expired
912	Reason for Rejection Message: Issuer Unavailable
913	Deviation Cause Message - Duplicate Transmission
914	Rejection Reason Message: Failed to track original transaction
915	Deviation Cause Message: Failure to Disable Negotiation or Checkpoint
916	Deviation Cause Message: MAC Incorrect
917	Reject Cause Message: MAC Key Synchronization Error
918	Rejection Reason Message: No binding keys available for use
919	Reject Cause Message: Encryption Key Synchronization Error
920	Cause of Deviation Message: Software/Hardware Security Error - Try Again
921	Deviation Cause Message: Software/Hardware Security Error - No Action
922	Rejection Reason Message: Incorrect Message Number Sequence

923	Status Message In-Process Query
950	Reason for Rejection Message - Business Agreement Violation
XXX	Code to be replaced by card status code or stop list reason code
0Y1	Confirmed, offline ICC
0Y3	Confirmed, offline ICC
1Q1	Rejected due to ICC offline mode
1Z1	Rejected due to ICC offline mode
1Z3	Rejected due to ICC offline mode

