Stored credential transactions (COF - card on file)

General information

Stored credential transactions (COF) provides the simple way to make the subsequent operations with the card data that stored during the initial operation.

The initial operation is carried out **with the obligatory receipt of the payer's consent** to store of card details for subsequent operations.

The initial operation can be:

- financial transaction - with card verification and following payment;
- registration transaction - with card verification without a following payment.

The initial operation is carried out using a conventional bankcard, or using a mobile device tokenization system, for example, Apple Pay or Samsung Pay.

The initial COF operation is carried out with verification by one of the possible ways (CVC2 and 3dS or using a mobile device).

Subsequent operations can be carried out after the successful completion of the primary transaction with the obligatory consent of the payer for the further use of card data. Subsequent operations are carried out without a re-verification procedure, if the processing protocol allows it.

> **Note.** If the initial operation is carried out using a mobile device tokenization system (Apple Pay, Samsung Pay or Google Pay), then the subsequent operations will be similar to the initial ones, without additional features.

CIT COF operations

Subsequent operations can be initiated by the payer. These operations are called «Customer Initialized Transaction» (CIT).

The One Click service is used to implement CIT COF operations in IPS Assist.

By participating in the service, the company maintains its database of customers, giving them an unique number (*customerNumber*), while the Assist system stores the encrypted credit card data corresponding to those merchant customers (*customerNumber*).

While performing of order payment, the customer gets an offer to save the card for future payments using the One Click service.

The One Click service allows a regular customer of a merchant to make new purchases without entering card data. The payment requires only confirmation by the CVC2 code. However, for subsequent operations CIT COF with a saved card, CVC2 will usually not be requested (except the cases where special settings are made at the request of the merchant).

> To start use the COF operations the merchant should send request to support team support@belassist.by. The request should contain the merchant identifier (Merchant_ID).

> Since unique numbers in the customer's database of the merchant (*customerNumber*) are generated and stored on the side of the merchant, it should pay special attention to protecting this data from hacking. It is necessary to increase the protection of personal data on the side of the merchant with the help of modern means (two-factor authentication for customers when entering the merchant's personal store, increased requirements for client passwords by security level, periodic updating of client passwords, etc.).

If the customer agrees to save the card, and proceed the successful payment then the IPS Assist generates a token - a unique card identifier that is stored in the system.

After the customer completes the payment for the order, a message with the payment parameters can be sent to the server of the merchant using the HTTP POST or SOAP method (for more details, see section 3.5 of the document «Setting up the technical interaction with IPS Assist») or the merchant can request the payment results by calling the *orderresult* web service.

In this case, an additional parameter appears in the list of payment parameters:

Name	Description
token	The unique identifier of the saved customer card

When the merchant builds the next order, it is possible to redirect to the IPS Assist payment page to pay with the earlier saved card. In this case, the merchant must send order data with a unique customer number, i.e. in addition to the authorized request parameters, it is necessary to transfer the *customerNumber*.

> ⚠ This parameter must not be transmitted simultaneously with the *recurringIndicator, recurringMinAmount, recurringMaxAmount, recurringPeriod, rRecurringMaxDate* parameters.

All actions related to payment by a saved card are performed on the Assist side.

If for any reason it is more convenient for the merchant to independently display an offer to the customer to pay the order with a previously saved card (on its website or in a specialized mobile application), then in order to use CIT COF, the merchant has to call the Token Pay payment web-service.

Note to the parameter *tokenType*, the value of which should be equal to 5, and also the value of the *paymentToken* parameter, which must contain the unique customer number in the merchant database (*customerNumber*) and the unique identifier of the saved customer card received during the initial operation of this customer from the IPS Assist.

The payment will be made as a CIT COF operation with the earlier saved card at the initial payment for this token of this customer.

The list of response parameters is similar to the standard response of the Token Pay service.

> ⚠ Payment will be made as a CIT COF operation only if the corresponding processing is set up for the merchant in the IPS Assist. Otherwise, the operation will fail.

> ⚠ If the merchant does not use the Token Pay payment web service and the information about the initial payment in the form of a token for CIT COF received from IPS Assist, then subsequent payments for the customer will be made according to the One Click program.

MIT COF operations

Subsequent operations can be initiated by the merchant. These operations are called «Merchant Initialized Transaction» (MIT).

Recurrent payments are used for the implementation of MIT COF operations in APC Assist.

Recurrent payments are used when periodic withdrawal of funds from a customer’s bank card to the account of the service provider’s company is required. This is a different kind of subscription - payment for hosting, mobile phone, access to resources, etc. This method is convenient for the user, since the card data is entered by the customer once during the first payment.

The merchant can store the subscription schedule on its side and initiate subsequent payments through a request to the web service of the IPS Assist (description of the parameters of the initiating authorization request and the web service for subsequent payments see the corresponding section.

In contrast to recurrent payments, the amounts of MIT COF payments, as well as the intervals at which they are made, are not fixed. Thus, the merchant can initiate a payment of the required amount at any time (for example, in the case of a zeroing of the balance of a service).

To make subsequent payments as MIT COF, only one of the additional parameters must be transferred:

Parameter	Mandatory field	Adopted values	Default value	Description
recurringIndicator	Yes	1 –recurring payment 0 – standard payment	0	Recurring payment indicator ⚠ This parameter must not be transmitted simultaneously with the *CustomerNumber* parameter.
recurringMinAmount	Yes	Number, 15 digits, two digits after the delimiter (delimiter '.')		Min amount of recurrent payments. Mandatory if RecurringIndicator = 1 ⚠ This parameter must not be transmitted simultaneously with the *customerNumber* parameter.

recurringMaxAmount	Yes	Number, 15 digits, two digits after the delimiter (delimiter '.')		Max Amount of recurrent payments. Mandatory if *recurringIndicator = 1* This parameter must not be transmitted simultaneously with the *customerNumber* parameter.
recurringCount	Yes	Number, 3 digits		Frequency of recurrent payments in days. Mandatory if *recurringIndicator = 1* This parameter must not be transmitted simultaneously with the *customerNumber* parameter.
recurringMaxDate	Yes	Date as string in DD.MM.YYYY format		The end date of recurrent payments. Mandatory if *recurringIndicator = 1* This parameter must not be transmitted simultaneously with the *customerNumber* parameter.

Payment will be made as a MIT COF operation only if the corresponding processing is set up for the merchant in the IPS Assist. Otherwise, the transaction will be carried out as a regular recurrent payment.

Additional web services for COF

Getting a list of customer cards tokens

The web service allows to a merchant to receive a list of tokens of cards by the unique identifier of their client.

To access the web service send a request using the POST method to the request URL https://<SERVER_NAME>/pay/oneclick/v1/token.cfm with parameters in JSON format.

List of request parameters:

Parameter		Mandatory field	Adopted values	Default value	Description
merchant	**merchant_Id**	Yes	Number		The enterprise identifier in IPS Assist
	login	Yes	20 characters		Web service user login
	password	Yes	30 characters		Web service user password
customerNumber*		Yes	32 characters		Merchant's internal customer identification
language		No	RU - Russian, EN - English	Language of legal entity/enterprise	Language of authorized pages

*The parameter is automatically validated according to the rules.

An example of request:

```
POST https://<SERVER-NAME>/pay/oneclick/v1/token.cfm
POST data:{"merchant":{"merchant_id":"452739", "login":"LOGIN", "password":"PASSWORD"}, "customernumber":"
1234", "language":"RU"}
```

The request result is sent in JSON format and contains the following parameters.

List of response parameters:

Parameter	Value
token	Token of stored card
post	Last 4 digits of the card number
bin	First 6 digits of the card number

brand	Payment system of the card
bank	Name of Bank-Issuer
expire	Month and year of card expiration (MM.YYYY)
active	Whether the card is valid
needCVC	Entering of CVC is required for authorization
assets*	Media data array of the card (determine the appearance of the card)

*If IPS Assist does not have media data for the card, then this parameter is returned with an empty value in the response.

Contents of the media data array

Parameter	**Adopted values**	**Description**
assetType	BRAND_LOGO ISSUER_LOGO CO_BRAND_LOGO CARD_BACKGROUND_COMBINED CARD_BACKGROUND ICON TERMS_AND_CONDITIONS	Type of card design graphic component
altText		
paymentType	VISA MC	Payment system type
assetData		
url		URL of the file containing the image of card design graphic component
mediaType	image/png image/pdf (VISA) application/pdf (MC) text/plain text/html	Image file format
width		Image dimensions in pixels
height		

In case of an error, the service returns the error data in JSON format.

{"fault":{"detail":"<error message>","faultcode":<first tcode>,"faultstring":<second code >}}.

Example of the response when the customer has several cards:

```
[{"expire":"12.2025","token":"49D902C6-1CCF-4605-9228-11D0E8115C77","bank":"","brand":"VISA","bin":412362,"
active":true,"needCVC":false,"post":"0028","assets":[{"assetType":"ICON","altText":"cardSymbol","paymentType":"
VISA","assetData":[
{"url":"https://emvts-test-assets.s3.amazonaws.com/A5/VISA/aa3dcd69e8e8407fa2a59a1956994610/w100_h100.png","
mediaType":"image/png","width":100,"height":100}
]},{"assetType":"CARD_BACKGROUND","altText":"digitalCardArt","paymentType":"VISA","assetData":[
{"url":"https://emvts-prod-assets.s3.eu-north-1.amazonaws.com/B5/VISA/fa0ff42cfa9e41b09ae8c2899d61394a
/w1536_h969.png","mediaType":"image/png","width":1536,"height":969}
]}]},{"expire":"12.2025","token":"4D7F0B0D-B701-47B0-779A-BB0CB5F25A14","bank":"","brand":"MasterCard","bin":
518615,"active":true,"needCVC":false,"post":"1030","assets":[{"assetType":"BRAND_LOGO","altText":null,"
paymentType":"MC","assetData":[
{"url":"https://emvts-test-assets.s3.amazonaws.com/A5/MC/3789637f-32a1-4810-a138-4bf34501c509/w0_h0.pdf","
mediaType":"image/pdf","width":null,"height":null}
]},{"assetType":"ICON","altText":null,"paymentType":"MC","assetData":[
{"url":"https://emvts-prod-assets.s3.eu-north-1.amazonaws.com/B5/MC/a7556277-51e9-4ef1-b503-dd8989e7a8bf
/w100_h100.png","mediaType":"image/png","width":100,"height":100}
]},{"assetType":"CARD_BACKGROUND_COMBINED","altText":null,"paymentType":"MC","assetData":[
{"url":"https://emvts-prod-assets.s3.eu-north-1.amazonaws.com/B5/MC/eb50276e-7e28-4cd1-9b95-c8dfbe595db6
/w1536_h969.png","mediaType":"image/png","width":1536,"height":969}
]},{"assetType":"ISSUER_LOGO","altText":null,"paymentType":"MC","assetData":[
{"url":"https://emvts-prod-assets.s3.eu-north-1.amazonaws.com/B5/MC/de7ce012-641e-450c-8b4f-a65d78062999
/w1372_h283.png","mediaType":"image/png","width":1372,"height":283}
]}]}]
```


An example of the response when there is no media data for the card:


```
[{"expire":"12.2025","token":"91720736-CF93-469C-63A0-E5859815D6BA","bank":"Bank USA","brand":"MasterCard","
bin":546792,"active":true,"needCVC":false,"post":"4128","assets":[]}]
```


Deleting a stored card (token) of a merchant customer

The web service allows a merchant to remove a card token by the unique identifier of its customer.

To access the web service send a request using the POST method to the request URL https://<SERVER_NAME>/pay/oneclick/v1/rem_token.cfm with parameters in JSON format.

List of request parameters:

Parameter		Mandatory field	Adopted values	Default value	Description
merchant	**merchant_Id**	Yes	Number		The enterprise identifier in IPS Assist
	login	Yes	20 characters		Web service user login
	password	Yes	30 characters		Web service user password
customerNumber*		Yes	32 characters		Merchant's internal customer identification
language		No	RU - Russian, EN - English	Language of legal entity/enterprise	Language of authorized pages
token		Yes	36 characters		Token of stored card

*The parameter is automatically validated according to the rules.

An example of request:

```
POST https://<SERVER-NAME>/pay/oneclick/v1/rem_token.cfm
POST data:{"merchant":{"merchant_id":"452739","login":"LOGIN", "password":"PASSWORD"}, "token": "54F5A3DE-4B47-
4B40-03B5-7FF992D6B401", "language": "RU", "customernumber":"1234"}
```

The request result is sent in JSON format with the message «*Token is remowed*».

Example of the response:

```
{"message":"Token 33479991-B4E5-40FE-D9ED-EDC94BA7FA14 is removed"}
```

In case of an error, the service returns the error data in JSON format.

{"fault":{"detail":"<error message>","faultcode":<first tcode>,"faultstring":<second code >}}.

Back on top