ORDER PAYMENT BY TOKEN (APPLEPAY/GOOGLEPAY/SAMSUNGPAY) IN INTERNET PAYMENT SYSTEM ASSIST

1. The main features of payment by the token

Payment by token is a technology of payment in shops and in the online stores by means of some mobile devices, the main manufacturers of which (for example, Apple Pay, Samsung Pay or Google Pay) provide means for payments execution through the tokenization system.

There are 4 ways for token payment processing:

1. Payment through the payment terminal, while the mobile device is applied to the card reader similarly to the application of cards with the support of contactless payment. Such payments are conducted as POS transactions.
2. Payment by the button built into the online -store mobile app (InApp scheme). Such payments are conducted as ECOM transactions.
3. Payment in the browser (Web scheme), while receiving of encrypted package with payment data and a token is performed on the side of the Assist payment pages. Such payments are conducted as ECOM transactions.
4. Payment in the browser (Web scheme), while receiving of encrypted package with payment data and token is performed on the page of the online store. Such payments are conducted as ECOM transactions.

Note. If the customer chooses to pay using Apple Pay or Google Pay in a browser on a device that does not support such payment type, then IPS Assist (if configured appropriately) can display a special QR code on the payment page. The customer will be asked to use another suitable device to read this QR code and successfully complete the payment.

Assist provides the possibility to organize the receiving of payments by tokens in all 4 ways. However, individual token providers do not support all payment ways. The table below shows possible options of payments processing by tokens using different providers.

Token provider	POS payment	Pay by button in mobile app (InApp)	Payment in the web-browser on the IPS Assist side (Web)	Payment in the browser on the online store side (Web)
Apple Pay	Through the Assist.mPOS app + card reader	Using Assist.SDK for iOS or Apple Pay SDK + TokenPay service	Supported only with payment in the Safari browser on the MacOS platform	Only in the Safari browser on the MacOS platform, using samples for working with ApplePay JS API on the Apple Pay Developers site and the TokenPay service
Samsung Pay	Through the Assist.mPOS app + card reader	Using Samsung Pay SDK + TokenPay service	Through integration with the IPS Assist payment pages or from a mobile app via WebView	Not supported by Samsung Pay
Google Pay*	Through the Assist.mPOS app + card reader	Using Google Pay API + TokenPay service	Payment on the mobile device, through the browser via the Google Pay API when switching to the IPS Assist payment page from the online store site	When making a payment on a mobile device, through a browser via Google Pay API: using code samples for working with the Google Pay API on the Google Developers site and the TokenPay service

*Payment by non-tokenized cards that saved in your Google account is possible depending on the merchant settings. Payments by such cards can be processed either as COF operation or as E-COM operation with additional payment confirmation by the payer via entering the CVC2 code and passing 3DSecure authentication.

Thereby, the following payment options are available for **mobile apps developers**:

For **Apple Pay**:

- Using Assist SDK for iOS (via WebView) - for payment by the button on the IPS Assist payment page;
- using Assist.SDK for iOS - for payment by the button in app.;
- using Apple Pay SDK + TokenPay service (see Annex 1) - for payment by the button in app..

For **Google Pay**:

- using Assist SDK for Android (via WebView) — for payment by the button on the IPS Assist payment page;
- using Google Pay API + TokenPay service (see Annex 1) — for payment by the button in app.

For **Samsung Pay**:

- using Samsung Pay SDK + TokenPay service (see Annex 1) — for payment by the button in app.
- using Assist SDK for Android (via WebView) — for payment by the button on the IPS Assist payment page.

To receive payments by tokens on the online store website, the following options are available:

For **Google Pay**: only at the opening a payment page on mobile device:

- by the button on the IPS Assist payment page;
- using Google Pay API and TokenPay service (see Annex 1).

For **Samsung Pay**:

- by the button on the IPS Assist payment page.

For **Apple Pay** only at the opening a payment page in the Safari browser on the MacOS platform:

- by the button on the IPS Assist payment page
- using ApplePay JS API and TokenPay service (see Annex 1).

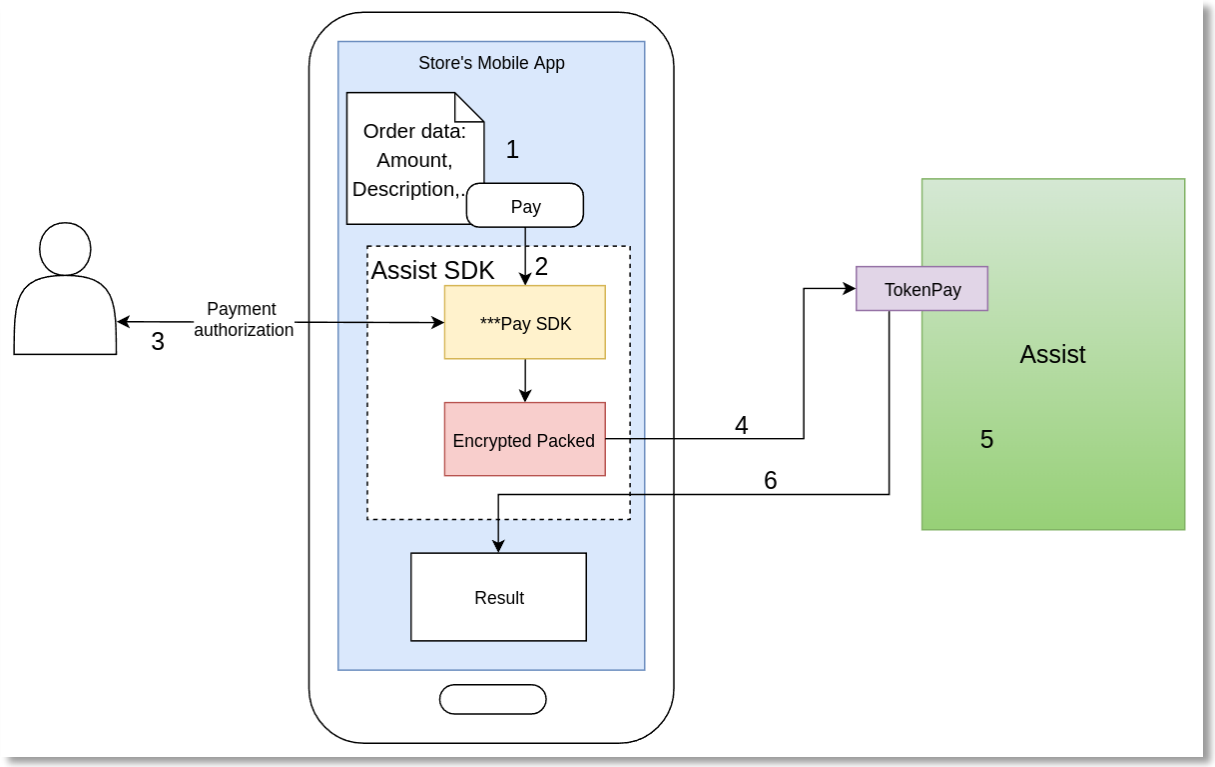
2. Payment by a token via a mobile terminal

Assist provides the possibility to pay by tokens Apple Pay, Samsung Pay and Google Pay within the framework of a customized mobile application Assist.mPOS using a card reader. With the Assist.MPOS and the card reader you can also accept payment by bank cards and cash.

To discuss possibilities to provide this service, you should contact the technical support service of the IPS Assist support@assist.ru.

3. Payment by the button in the mobile app (InApp)

When the token payment is executing by button in a mobile app, the following actions are performed:



1. The buyer selects the product or service in the store mobile app and presses the pay button.
2. An encrypted packet with payment data and a token is generated by accessing the SDK of the respective provider: Apple Pay, Samsung Pay or Google Pay API.
3. The SDK of the provider conducts the required payment authorization by the buyer on the mobile device (PIN / fingerprint).
4. The encrypted packet with payment data and a token is transferred to the TokenPay payment service on the side of IPS Assist (see Annex 1).
5. IPS Assist decrypts the package and performs payment with a token.
6. IPS Assist returns the payment results to the mobile app.

Actions from 2 to 6 can be executed once by calling SDK Assist:

- SDK Assist for Apple Pay is available on https://github.com/assist-group/assist-mcommerce-sdk-ios

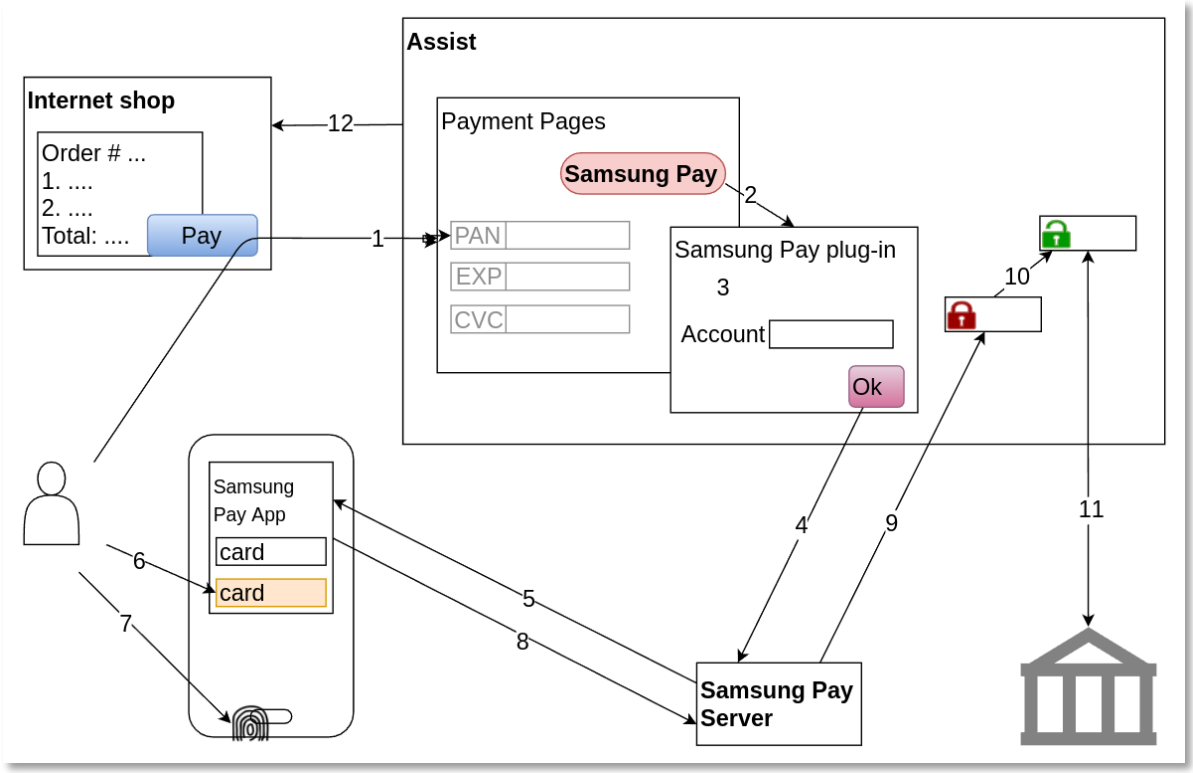
You have to use the corresponding SDK - Apple Pay SDK for IOS or Google Pay API for Android, or Samsung Pay SDK if you don't want to use SDK Assist in the online store mobile app, for creatation of the encrypted package with payment information. The created encrypted packet with payment details then must be transferred to the IPS Assist through the TokenPay service (see Annex 1).

To organize the receipt of payments by the button in the mobile app, you need to do the following preparatory steps:

- make a request to connect TokenPay payment processing service functionality to the IPS Assist technical support service support@assist.ru;
- implement in your app the support for payment of orders with a token, using the Assist SDK, SDK Apple Pay, Google Pay API or Samsung Pay SDK along with access to TokenPay service;
- obtain confirmation from the IPS Assist technical support service that all the necessary technical settings for the functioning of the TokenPay payment processing service when paying for goods and services to your merchant have been fulfilled;
- prepare for receiving payments:
    - when using Apple Pay, create and sign an Apple Pay certificate to make payments using the corresponding section in the IPS Assist Personal Account;
    - when using Google Pay, you should register with Google Pay API to get a MerchantID for use it as *MerchantID* parameter, set *gateway=*"assist" and specify *allowedCardAuthMethods* = ["CRYPTOGRAM_3DS"] to organize the scheme for encrypting a data packet with a token or *allowedCardAuthMethods* = ["PAN_ONLY", "CRYPTOGRAM_3DS"], if you also need to accept payments with non tokenized cards that saved in Google account;
    - when using Samsung Pay you need to request to the IPS Assist technical support to receive the CSR-file.

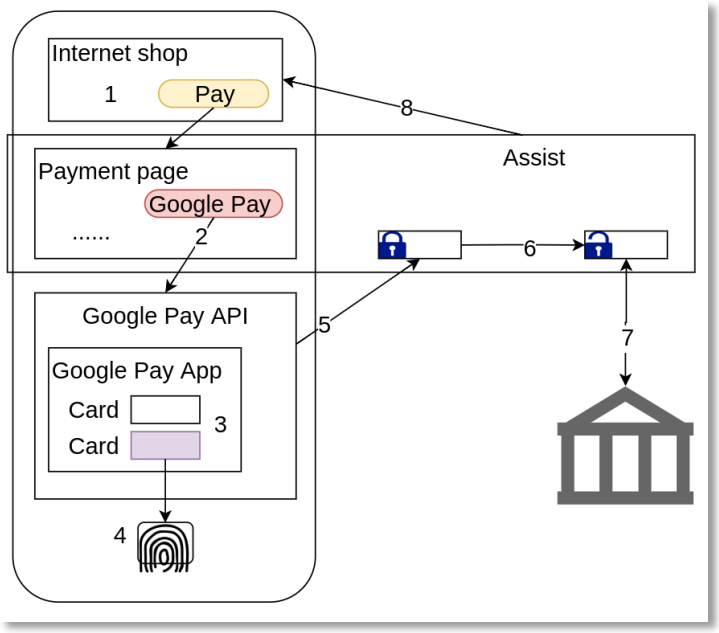
4. Payment by a token on the IPS Assist payment page

When you pay by the Samsung Pay token on the IPS Assist payment page, the following actions are performed:



1. The buyer selects the product or service on the online store site and presses the pay button, after which the store redirects the buyer to the IPS Assist payment page.
2. The buyer can press the button of token payment on the IPS Assist payment page.
3. Special plug-in Samsung Pay requests SamsungID of the user.
4. The plug-in sends a payment request to the Samsung Pay server.
5. Samsung Pay server sends a PUSH notification to the mobile device of the buyer
6. The buyer selects one of the attached cards.
7. The buyer confirms the payment (PIN or fingerprint).
8. The mobile device transmits data to the Samsung Pay server.
9. The IPS Assist receives an encrypted packet with a token and payment data from the Samsung Pay server.
10. The IPS Assist decrypts the packet with the token and payment data.
11. The IPS Assist performs payment by a token through the processing of a settlement bank.
12. The IPS Assist returns the results of payment to the online store site.

When you pay by the Google Pay token on the IPS Assist payment page, the following actions are performed:

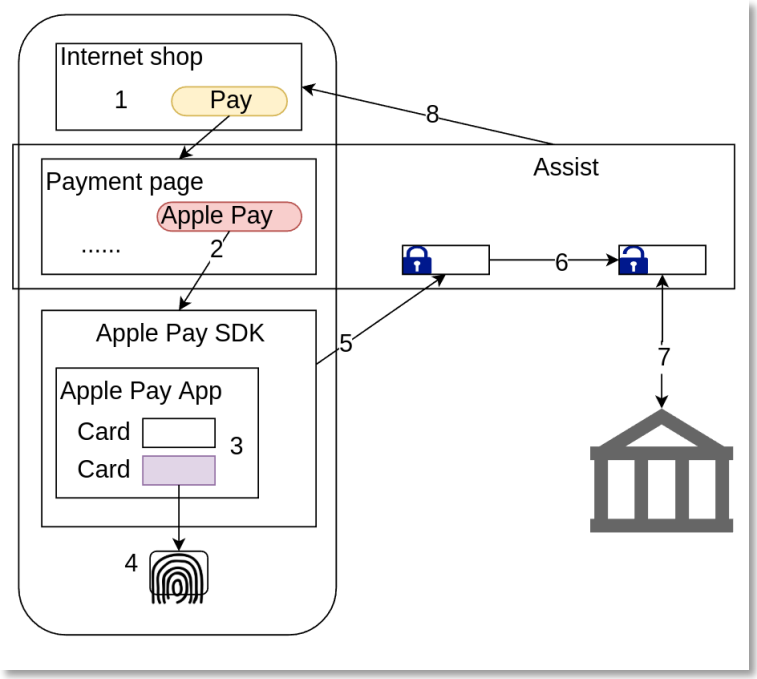


1. The buyer selects the product or service on the online store site and presses the pay button, after which the store redirects the buyer to the IPS Assist payment page.*
2. If the user opens the IPS Assist payment page on a mobile device in a browser and the Google Pay application is installed on this mobile device, he will see and can to click the Google Pay payment button.
3. After clicking the Google Pay payment button, a special dialog opens, where the buyer can select one of the cards added to the Google Pay mobile app.
4. After selecting the card, the buyer has to pass verification to confirm the payment.
5. Google Pay app creates an encrypted package with a token and payment data and returns it to the IPS Assist.
6. The IPS Assist decrypts the packet with the token and payment data.
7. The IPS Assist makes payment through the processing of the settlement bank in one of the following ways:
    - by token if the customer used a tokenized card;
    - as COF-payment by non-tokenized card, if it is allowed for the merchant and there is a corresponding processing;
    - as ECOM-payment by non-tokenized card with customer redirection to the CVC2 entry page and, if necessary, additional 3DSecure authentication.
8. The IPS Assist returns the results of payment to the online store site.

***Attention!** When using an iframe to display the payment page of the IPS Assist and the payment via Google Pay, you have to add the *allowpaymentrequest* permission.

Example: *<iframe allowpaymentrequest src="...">*.

When you pay by the Apple Pay token on the IPS Assist payment page, the following actions are performed:



1. The buyer selects the product or service on the online store site and presses the pay button, after which the store redirects the buyer to the IPS Assist payment page.*
2. If the user opened the IPS Assist payment page on a mobile device in a browser with Apple Pay SDK support, and the Apple Pay application is installed on this mobile device, he will see and be able to click the Apple Pay button.
3. After pressing the Apple Pay button a special dialog opens, where the buyer can select one of the cards added to the Apple Pay mobile app.
4. After selecting the card the buyer is asked to apply a finger to the reader to confirm the payment.
5. Apple Pay app creates an encrypted package with a token and payment data and returns it to the IPS Assist.
6. The IPS Assist decrypts the packet with the token and payment data.
7. The IPS Assist makes payment through the processing of the settlement bank.
8. The IPS Assist returns the results of payment to the online store site.

***Attention!** Payments via Apple Pay are not supported when using an iframe to display the IPS Assist payment page.

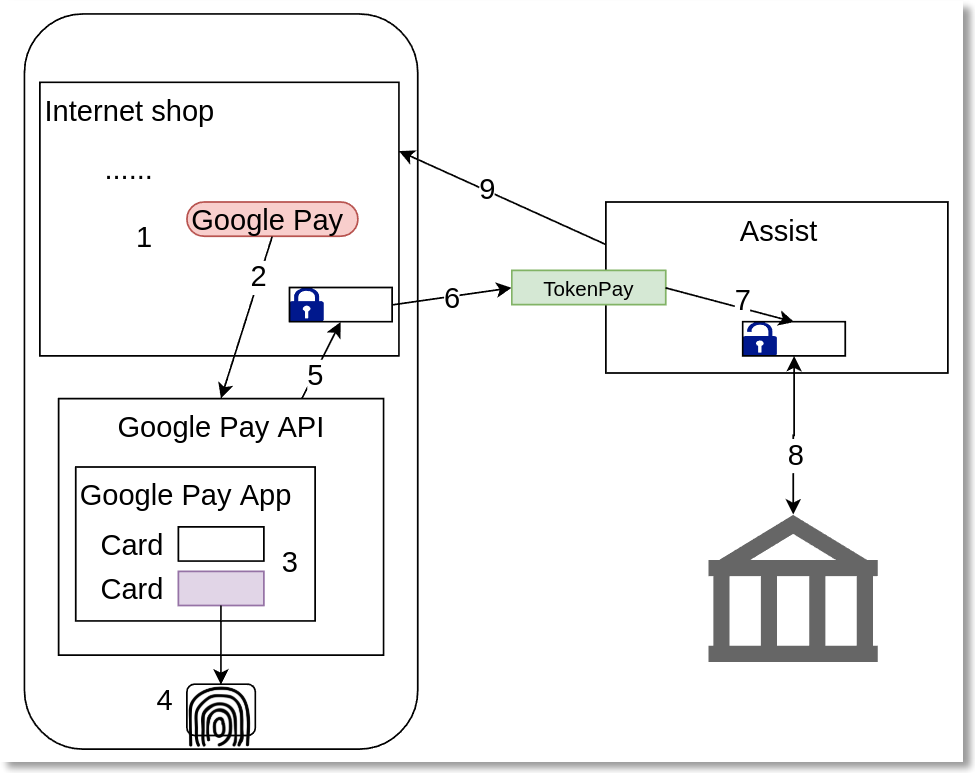
To organize the receipt of payments by the button on the IPS Assist payment page, you need to do the following preparatory steps:

- make a request to connect payment means for payments by token(Google Pay, Samsung Pay) to the IPS Assist technical support service support@assist.ru;

- obtain confirmation from the IPS Assist technical support service that all the necessary technical settings for the functioning of the TokenPay payment processing service when paying for goods and services to your merchant have been fulfilled.

5. Payment in the browser on the online store side

When you pay by the Google Pay token in the browser on the online store side, the following actions are performed:

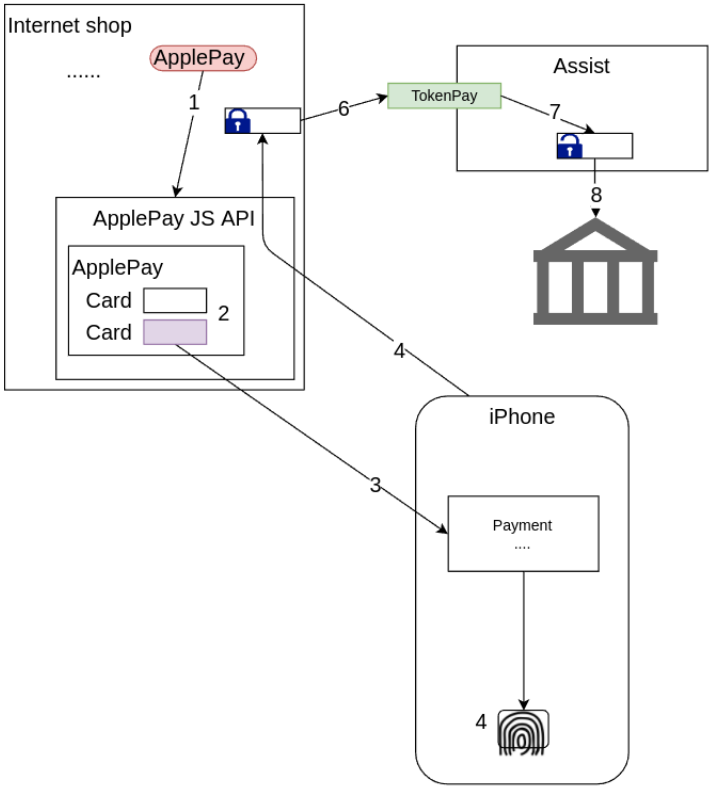


1. The buyer selects the product or service on the online store site and presses the Google Pay button (it is available in browsers on mobile devices).
2. After pressing the Google Pay button the Google Pay API is called and the certificate is transmitted with the Assist public key.
3. Then a special browser dialog opens, where the buyer can select one of the cards added to the Google Pay mobile app.
4. After selecting the card, the buyer has to pass verification to confirm the payment
5. Google Pay app creates an encrypted package with a token and payment data and returns it to online store page script.
6. The encrypted packet with the token and payment data is sent to the IPS Assist TokenPay service (see Annex 1).
7. The IPS Assist decrypts the packet with the token and payment data.
8. The IPS Assist performs payment by a token through the processing of a settlement bank in one of the following ways:
    - by token if the customer used a tokenized card;

- as COF-payment by non-tokenized card, if it is allowed for the merchant and there is a corresponding processing;
- as ECOM-payment by non-tokenized card with customer redirection to the CVC2 entry page and, if necessary, additional 3DSecure authentication.

9. The IPS Assist returns the results of payment to the online store.

When you pay by the Apple Pay token in the browser on the online store side, the following actions are performed:



1. The buyer selects the product or service on the online store site and presses the Apple Pay button (it is available only in Safari browser on the MacOS platform).
2. After pressing the Apple Pay button the Apple Pay API is called through the Apple Pay and the certificate with the merchant public key is transmitted. At the same time, a special dialog opens, where the buyer can select one of the cards added to the Apple Pay mobile app.
3. After selecting the card, a PUSH notification is sent to the iPhone of the buyer and the buyer is asked to apply a finger to the reader to confirm the payment.
4. Apple Pay app creates an encrypted package with a token and payment data and returns it to online store page script.
5. The encrypted packet with the token and payment data is sent to the IPS Assist TokenPay service (see Annex 1).
6. The IPS Assist decrypts the packet with the token and payment data.
7. The IPS Assist performs payment by a token through the processing of a settlement bank.
8. The IPS Assist returns the results of payment to the online store.

To organize the receipt of payments by token in the browser on the online store side, you need to do the following preparatory steps:

- make a request to connect payment means for payments by token(Apple Pay, Google Pay) to the IPS Assist technical support service support@assist.ru;
- embed into the code of your page:
  - to receive an encrypted packet with a token and payment data:
    - Google Pay API for Google Pay;
    - ApplePay JS API for Apple Pay;
  - for payment - access to TokenPay service;
- obtain confirmation from the IPS Assist technical support service that all the necessary technical settings for the functioning of the TokenPay payment processing service when paying for goods and services to your merchant have been fulfilled;
- prepare for payments receiving;
- when using Apple Pay, create and sign an Apple Pay certificate to make payments using the corresponding section in the IPS Assist Personal Account;
- when using Google Pay, you should register with Google Pay API to get a MerchantID for use it as *MerchantID* parameter, set *gateway*=”assist” and specify *allowedCardAuthMethods* = ["CRYPTOGRAM_3DS"] to organize the scheme for encrypting a data packet with a token or *allowedCardAuthMethods* = ["PAN_ONLY", "CRYPTOGRAM_3DS"], if you also need to accept payments with non tokenized cards that saved in Google account.

Annex 1

TokenPay payment information transfer

To transfer the encrypted payment data block send a request to the IPS Assist server via HTTP POST method (n UTF-8 coding).

The request URL for transfer the encrypted payment data block:

https://<SERVER_NAME>/pay/tokenpay.cfm ,

As <SERVER_NAME> you should use the received from support service (support@assist.ru) value as the domain name.

List of request parameters:

Parameter	Mandatory field	Adopted values	Default value	Description
Merchant_ID	Yes	Number		The enterprise identifier in IPS Assist
Login	Yes	8-20 characters		Login in IPS Assist (Latin letters, digits and symbol _)
Password	Yes	8-20 characters		Password in IPS Assist (Latin letters and digits)
OrderNumber	Yes	128 characters		Order number in the merchant payments system
OrderAmount	Yes	Number, 15 digits (delimiter: '.')		Payment amount, in original currency (e.g., 10.34).
OrderCurrency	No	3 characters	Currency of legal entity/enterprise	Code of currency of the **OrderAmount**
OrderComment	No	256 characters		Comment.
Delay	No	0 – one-stage operation 1 – double-stage operation	0	Attribute of a bankcard authorization for the double-stage operation mode.
Language	No	RU – Russian, EN – English	Language of legal entity/enterprise	Language of authorized pages
ClientIP	No	Maximum of 15 digits, 4 delimiters «.»		IP-address of the customer
TokenType	No	1 – Apple Pay; 2 – Google Pay; 3 – Samsung Pay	1	Payment Token Type Identifier
PaymentToken	No	JSON		Payment Token
Lastname	No	70 characters		Customer's last name.
Firstname	No	70 characters		Customer's first name.
Middlename	No	70 characters		Customer's middle name.
Email	No	128 characters		Customer's e-mail.
Address	No	256 characters		Customer's address.
HomePhone	No	64 characters		Customer's home phone number.
WorkPhone	No	20 characters		Customer's work phone number.
MobilePhone	No	20 characters		Customer's mobile phone number.
Fax	No	20 characters		Customer's fax number.

Parameter	Mandatory field	Adopted values	Default value	Description
Country	No	3 characters		Customer's country.
State	No	3 characters		Customer's region.
City	No	70 characters		Customer's city.
Zip	No	25 characters		Customer's post zip code.
isConvert	No	0 – don't convert to the base currency 1 - don't convert to the base currency if possible 2 – always convert to the base currency	1	Currency conversion indicator
Format	No	4 – SOAP 5 – JSON	5	Results format.
Signature	No	String		The string is joined from the following order parameters: Merchant_ID;OrderNumber;OrderAmount ;OrderCurrency with semicolon as delimiter. Then the MD5 hash prepared from this string. Hash is signed by private RSA key of the merchant. Key length - 1024. Received bit sequence is a signature. Signature is transferred BASE64 coded string.
RecurringIndicator	No	1 –recurring payment 0 – standard payment	0	Recurring payment indicator
RecurringMinAmount	No/Yes	Number, 15 dig. (decimal separator '.')		Minimum payment amount for recurring payments Mandatory when RecurringIndicator = 1
RecurringMaxAmount	No/Yes	Number, 15 dig. (decimal separator '.')		Maximum payment amount for recurring payments Mandatory when RecurringIndicator = 1
RecurringPeriod	No/Yes	10 digits number		Recurring interval in days Mandatory when RecurringIndicator = 1
RecurringMaxDate	No/Yes	Date in string representation DD.MM.YYYY		Finish date of recurring payments Mandatory when RecurringIndicator = 1
Chequeitems	No/Yes**	String in JSON-format		Cheque items according to the document "Payment with cheque".
GenerateReceipt	No	0 or 1	1	Permission to generate of a fiscal receipt. If the value of parameter is 0, the generation of a fiscal receipt is prohibited for this order.
Tax	No	10 characters	Parameter is determined by the "Default tax rate" option of the merchant and used in the mode without transfer of the cheque items (the entire amount must be paid with one rate).	Tax rate identifier, reference value (novat, vat0, vat10, vat18, vat110, vat118)

Parameter	Mandatory field	Adopted values	Default value	Description
ReceiptLine	No	128 characters	Parameter is determined by the "Default text of cheque item" option of the merchant and used in the mode without transfer of the cheque items.	Text description of the cheque item, if the cheque has one single item.
FPMode	No	Number	Parameter is determined by the "Default payment method" option of the merchant and used in the mode without transfer of the cheque items.	The payment method.
TaxationSystem	No/Yes*	Number	From the merchant's or CRE settings	The taxation system.

*If the merchant uses several tax systems, the transfer of this parameter becomes mandatory.

**The cosist of the cheque (fields with the name, price per unit of product, tax rate and payment method for each item) is determined according to the following rules.

1. If the request contains the *ChequeItems* structure (see the document "Payment with a cheque"), then a cheque is generated with the number of items according to the number of rows passed to *ChequeItems*, the parameters must be specified in each item:
   - *product и/или name*;
   - *price*;
   - *quantity*;
   - *amount*;

   herewith:

   the name of each item is built as the combination of the *product* and *name* parameters, separated by a space (if both are transferred) or as one of the *product* or *name* parameters, respectively (if only one is transferred);

   if the *Tax* or *FPmode* parameters are not transferred in items, then they can be transferred in the request parameters (in this case, the values of these parameters will be the same for all cheque items transferred to *ChequeItems*);

   if the *Tax* or *FPmode* parameters are not transferred in the request parameters, then their values are taken from the merchant settings (the values of these parameters will be the same for all cheque items transferred to *ChequeItems*);

   if the merchant settings for these parameters are missing, the request processing ends with an error (there is not enough data to perform the operation).

2. If the request does not contain the *CheckItems* structure, then a cheque is generated with one item, in which:
   - the item name is taken from the *ReceiptLine* parameter, if it is transferred in the request; if the *ReceiptLine* parameter is not transferred in the request, then the item name is taken from the merchant settings; if the merchant settings for these parameters are missing, then the name "*Payment for the order*" is used for the name;
   - the unit price of product is taken from the request parameter *Amount*;
   - the quantity is always equal 1;

   if the Tax or FPmode parameters are not transferred in the request parameters, then their values are taken from the merchant settings;

   if the merchant settings for these parameters are missing, the request processing ends with an error (there is not enough data to perform the operation).

The list of response parameters:

Name	Description
Order parameters	
ordernumber	Order number
billnumber	Unique order number in IPS Assist
testmode	Test mode
ordercomment	Comment
orderamount	Original amount of order
ordercurrency	Original currency of order
firstname	Payer's first name
lastname	Payer's last name
middlename	Payer's middle name
Email	Payer's e-mail
orderdate	Date of order (GMT)
orderstate	Order status
packetdate	Request issue date (GMT)
signature	• For signature type 'MD5' - empty • For signature type 'PGP' – value X signed by secured key of IPS Assist, BASE64 encoded
checkvalue	uppercase(md5(uppercase(md5(SALT) + md5(X)))), where SALT – secret word ; X – result of the following parameters string concatenation merchant_id, ordernumber, orderamount, ordercurrency, orderstate (without delimiters), + - means string concatenation
Operation parameters	
billnumber	Extended format of Billnumber: billnumber.<operation number>
operationtype	Operation type
operationstate	Operation status
amount	Operation amount
currency	Currency of operation
ipaddress	Payer's IP-address
clientip	Client's IP-address
meantype_id	Payment means ID
meantypename	Type of payment means
meansubtype	Payment means subtype
meannumber	Number of payment means
cardholder	Payment means holder
issuebank	Name of issue bank
bankcountry	Country of issue bank
responsecode	Response code
message	Operation result message
customermessage	Result message for a customer
recommendation	Recommendation

approvalcode	Authorization code
protocoltypename	Protocol
processingname	Processing name
operationdate	Operation date and time(GMT)
authresult	3Ds authorization result (Y – success, N - fail, A - Attempt, U – unknown)
authrequired	The card involvement in 3Ds check result (1 – involved, 0 – not involved, -1 – unknown, null – error appear during involvement check)
slipno	Financial transaction identifier
firstcode*	The first code of automated interfaces error
secondcode*	The second code of automated interfaces error
continueurl*	The address to redirect to the authentication page for the buyer

*Parameters are responsed only if the payment is made with a non- tokenized card for ECOM operations with customer redirection to the CVC2 entry page and, if necessary, additional 3DSecure authentication. In this case, the parameters firstcode and secondcode take the values "3" and "115", respectively.

Attention! Please, note that several operations can be created within one order (payment, payment confirmation, cancellation). Furthermore, there can be several payment operations within one order, if some of them were unsuccessful. The order can have the only one successful payment operation. Thus there can be several enclosed operations within one order number in the response to a request for the operations results.

Attention! The *testmode* value of response has to be checked. If the payment was made in test mode (*testmode* = 1), then the shipment of goods or providing of services for the current request is not required.

An example of a request result in JSON format if the customer used a tokenized card:


```
{"order":
{
  "ordernumber":"26012015_4"
  ,"billnumber":"5899110210668265"
  ,"testmode":"0"
  ,"ordercomment":"test payment"
  ,"orderamount":"24.00"
  ,"ordercurrency":"RUB"
  ,"firstname":""
  ,"lastname":""
  ,"middlename":""
  ,"email":""
  ,"orderdate":"26.01.2015 17:25:39"
  ,"orderstate":"Timeout"
  ,"fraud_state":""
  ,"fraud_reason":""
  ,"checkvalue":"85C6C974AADC1CADFCFD195730ED090B"
  ,"operations":[
{"billnumber":"5899110210668265.1"
,"operationtype":"100"
,"operationstate":"In Process"
,"amount":"24.00"
,"currency":"RUB"
,"clientip":"10.20.10.85"
,"ipaddress":"0.0.0.0"
,"meantype_id":"61"
,"meantypename":"In Process"
,"meansubtype":""
,"meannumber":""
,"cardholder":""
,"issuebank":"UNKNOWN"
,"bankcountry":"UNKNOWN"
,"responsecode":"AS200"
,"message":""
,"customermessage":"
,"recommendation":""
,"approvalcode":""
,"protocoltypename":"NET"
,"processingname":""
,"operationdate":"26.01.2015 17:25:39"
,"authresult":""
,"authrequired":""
,"slipno":""}
]
}
,"packetdate":"26.01.2015 18:16:19"
}
```

An example of a request result in JSON format if the customer used a non-tokenized card:


```
{"order":
{
  "ordernumber":"2019.07.01-1226",
  "billnumber":"516009019047110",
  "testmode":"0",
  "ordercomment":"Test 4 mode",
  "orderamount":"101.45",
  "ordercurrency":"RUB",
  "firstname":"",
  "lastname":"",
  "middlename":"",
  "email":"",
  "orderdate":"01.07.2019 15:31:16",
  "orderstate":"In Process",
  "fraud_state":""
  "fraud_reason":""
  "checkvalue":"85C6C974AADC1CADFCFD195730ED090B"
  "operations":[
{"billnumber":"516009019047110.1",
"operationtype":"100",
"operationstate":"New",
"amount":"101.45",
"currency":"RUB",
"clientip":"10.20.10.66",
"ipaddress":"127.0.0.1",
"meantype_id":"1",
"meantypename":"VISA",
"meansubtype":"Classic",
"meannumber":"411111****1111",
"cardholder":"N/A",
"cardexpirationdate":"12/24",
"issuebank":"New Assist Bank",
"bankcountry":"Россия",
"responsecode":"AS300",
"message":"",
"customermessage":"", "
"recommendation":"",
"approvalcode":"",
"protocoltypename":"NET",
"processingname":"UCS",
"operationdate":"01.07.2019 15:31:18",
"authresult":"",
"authrequired":"",
"slipno":""}
]
}
"packetdate":"26.01.2015 18:16:19",
"firstcode":"3",
"secondcode":"115",
"continueurl":"https://<SERVER_NAME>/pay/pay.cfm?CFSID=LiRNMzMiSFc3Wy0nWFdaQCJYTiQg
Cg%3D%3D"
}
```

Annex 2

OrderState field values

Orderstate	Meaning	Description
In Process	In process	Order created.
Delayed	In wait for payment confirmation	Payment operation for this order has been performed using the double-stage operation mode but not confirmed yet.
Approved	Payment performed	Payment operation for the given order successfully completed.
PartialApproved	Partially paid	Payment operation performed for a part of the order amount (not used).
PartialDelayed	Partially confirmed	Payment confirmed for a part of the payment amount.
Canceled	Cancelled	Cancelled for full payment amount.
PartialCanceled	Partially cancelled	Cancelled for partial payment amount.
Declined	Declined	Payment failed.
Timeout	Closed upon timeout	Order closed by timeout.

OperationType field values

Code	Parameter description
100	Approve
200	Charge
300	PaymentCancel

OperationState field values

Operationstate	Description
New	Created
In Process	In the process
Success	Successfully done
Failure	Finished with a failure
TimeOut	Closed by time out

Types of payment means *MeanTypeName*

ID	Type name	Description
1	VISA	Visa
2	MC	MasterCard
3	DCL	Diners Club
4	JCB	Japan Credit Bureau
5	AMEX	American Express
6	MIR	MIR Card
10	Discover	Discover
12	Points	Points

30	WebMoney	WebMoney
32	YandexMoney	YandexMoney
36	QIWI	QIWI
37	BankClient	BankClient
39	QIWIBeeline	QIWIBeeline
40	QIWIMts	QIWIMts
41	QIWIMegafon	QIWIMegafon
42	QIWITele2	QIWITele2
60	ApplePay	Apple Pay
61	SamsungPay	SamsungPay
62	GooglePay	GooglePay

The adopted values of payment types (parameter *Paymenttype*)

Code	**Description**
1	Bank card
2	Wallets
3	Cashless payments
10	Cash
21	Prepayment (offset of advance payment)
22	Postpayment (on credit)
23	Payment by counter submission

The adopted values of tax rate identifier

Code	**Description**
novat	without VAT
vat0	VAT rate 0%
vat10	VAT rate 10%
vat18	VAT rate 18%
vat110	VAT calculated by rate 10/110
vat118	НДС VAT calculated by rate 18/118

The adopted values of payment method

Code	**Name**	**Description**
1	Prepayment 100%	Full prepayment before the item is transferred
2	Prepayment	Partial prepayment before the item is transferred
3	Advance payment	Advance payment
4	Full payment	Full payment, including taking into account the advance payment at the time of transfer of the subject of payment
5	Partial payment and credit	Partial payment of the object of payment at the time of it's transfer with subsequent payment on credit
6	Transfer on credit	Transfer of the subject of payment without payment at the time of its transfer with subsequent payment on credit
7	Credit payment	Payment for the subject of payment after it's transfer with payment on credit (credit payment)

The adopted values of taxation system

Code	Description
0	General
1	Simplified income
2	Simplified income minus expenses
3	Unified tax on imputed income
4	Unified agricultural tax
5	Patent taxation system

Codes of automated interfaces

First code	Description
0	Success
1	Error
2	Internal error
3	No mandatory parameter
4	Parameter format error
5	Incorrect parameter value
6	Incorrect system version
7	Authentication error
8	Authorization error
9	Encryption error
10	Object not found
11	Duplicate object
12	Object is blocked
14	Forbidden object
15	Forbidden operation
16	Operation timeout
17	Limits error
18	Suspicion of fraud
19	Access denied
20	3D secure error
21	Operation declined

Second code	Description
0	No additional information
1	Unexpected error
2	Generated document is too large
3	Interface request interval is exceeded
4	Sampling interval is too big
5	Key encryption error
6	Key decryption error
100	Parameter MERCHANT_ID
101	Parameter LOGIN
102	Parameter PASSWORD
103	Parameter FORMAT
104	Parameter DATE
105	Parameter CURRENCY
106	Parameter MEANNUMBER
107	Parameter ORDERNUMBER
108	Parameter AMOUNT
109	Parameter DELAY
110	Parameter COMMENT
111	Parameter MEANTYPE
112	Parameter EXPIREMONTH
113	Parameter EXPIREYEAR
114	Parameter CARDHOLDER
115	Parameter CSC2
116	Parameter CLIENTIP
117	Parameter LASTNAME
118	Parameter FIRSTNAME
119	Parameter MIDDLENAME
120	Parameter EMAIL
121	Parameter ADDRESS
122	Parameter PHONE
123	Parameter CITY
124	Parameter STATE
125	Parameter ZIP
126	Parameter LIMITTYPE
127	Parameter LANGUAGE
128	Parameter COUNTRY
129	Parameters STARTDAY and/or STARTMONTH and/or STARTYEAR
130	Parameters ENDDAY and/or ENDMONTH and/or ENDYEAR
131	Parameter SUCCESS

132	Parameter ZIPFLAG
133	Parameter HEADER
134	Parameter HEADER1
135	Parameter DELIMITER
136	Parameter OPENDELIMITER
137	Parameter CLOSEDELIMITER
138	Parameter ROWDELIMITER
139	Parameter FIELDS
140	Parameter SSL
141	Parameters LOGIN and/or PASSWORD
142	Parameter EXPIREMONTH and/or EXPIREYEAR
143	Parameter BILLNUMBER
144	Parameter PROTECTCODE
145	Parameter OPTYPE
146	Parameter OPSTATE
147	Parameter RPSERIES
148	Parameter RPNUMBER
149	Parameter ASSISTID
150	Parameter PIN
153	Parameter TICKET_NUMBER, PNR
154	Parameter URL
155	Parameter TRANSACT_ID
156	Parameter TID
157	Parameter MID
159	Parameter BIN
161	Parameter BillingNumber
163	Parameter TRANSACTSTATE
164	Parameter ORDERSTATE
165	Parameter TRANSACTTYPE
167	Parameter Currency RATE
170	Parameter ResponseCode
173	Parameter IP-ADDRESS
176	Parameter PNR
177	Parameter PaymentMode
179	Parameter CHEQUE
185	Parameter BILLSENDTYPE
186	Parameter HASHTYPE
187	Parameter BILLNO
188	Parameter BILLNOTEMPLATE
189	Parameter BILL_ID

190	Parameter BILLSTATE
200	Object Enterprise
201	Object Order
202	Object Customer
203	Object Bankcard
204	Object Bank
205	Object Processing
206	Object Terminal
207	Object Country
208	Object Currency
209	Object Currency rate
210	Object Commission
211	Object Limit
212	Parameter TestMode
213	Parameter PaymentType
214	Object Template
215	Object SOAP PACKET
216	Object Operation
217	Object Meantype (PaymentSystem)
218	Object Payment means
220	Object TRANSACTION
221	Object User
225	Object Enterprise
226	Object Company
228	Object Bill
300	Authorization cancellation
301	Refund
302	Financial confirmation (deposit)
305	Financial transaction cancellation
306	Payment operation
307	Confirmation operation
308	Cancellation operation
309	Operation Bill Revoke
320	Recurring payment
350	Web service
400	Error: Directory Server
401	Reaction settings not found
402	Authorization expectation via 3D-Secure
403	Authorization denied by DS