

Online Merchant Purchase

Online Purchase by Merchants REST API Documentation

Online Merchant Purchase Version 1.8

Last Update: 2023-01-08

Snapp! Pay

Version Information		
Document Version	Date	Note
1.0	13 th February 2022	Online Merchant Purchase 1.0
1.1	5 th March 2022	Remove callback Url from purchase process api
1.2	16 th April 2022	Update get payment token and check merchant eligibility APIs
1.3	1 st May 2022	Update get payment token API (param cartList - totalAmount - cartitems changed to mandatory)
1.4	8 th May 2022	Get Payment Status API
1.5	14 th May 2022	Cancel API
1.6	16 th May 2022	Update API
1.7	20 th September 2022	Get payment token and Update APIs changed (additional params: externalSourceAmount and commissionType)
1.8	8 th January 2023	Update Get Payment Status api (Add amount parameter to response)

Table Content

Introduction	5
General Response Format	6
Authentication Mechanism	7
Check Merchant Eligibility	8
Purchase process	9
Get Payment token service	10
Verify	13
Settle	14
Revert	15
Get Payment Status	16
Cancel	17
Update	18
Error	20
Appendix(online merchant purchase flow)	21

Introduction

SnappPay Installment Purchase is a payment method launched by SnappPay, which is available for SnappPay's users as a payment method after activation. It allows accessing the installment payment method from a SnappPay credit. Your customers can link to their wallet and buy now by this method without paying the total order amount. they only pay the first installment, and they should repay the rest of the installments in X months later.

It helps you to increase your conversion rate.

General Response Format

In any successful endpoint call (2xx HTTP status codes) the response is a JSON content with the form of :

```
{
  "successful": true,
  "response": {
    "y": "x",
    "z": 0
  }
}
```

In any unsuccessful endpoint call (4xx and 5xx HTTP status codes) the response is a JSON content

with the form of :

```
{
  "errorData": {
    "errorCode": 400,
    "message": "",
    "data": null
  },
  "successful": false
}
```

Authentication Mechanism

In order to be able to call endpoints of this service you should first generate a JWT; that can be generated by the following endpoint, we have to put your production IP range into a white list for the sake of security.

POST: /api/online/v1/oauth/token

Sample of curl request:

```
curl --location --request POST 'https://{url}/api/online/v1/oauth/token' \
--header 'Authorization: Basic base64_encode({client_id} + ':' + {client_secret})' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'scope=online-merchant' \
--data-urlencode 'username={username}' \
--data-urlencode 'password={password}'
```

Sample of response:

```
{
  "access_token": "eyJhbGciOiJ...",
  "token_type": "bearer",
  "expires_in": 3600,
  "scope": "online-merchant",
  "iat": 1645369630,
  "jti": "ogYjKGgZACeJ_Qc7Jb5e7v2AWWU"
}
```

After getting a token you should put it in the Authorization header for all services.

Check Merchant Eligibility

As an API caller, to make sure that you can show a credit option to users on the checkout page, you must first call this service. For calling this API merchant has to put a valid JWT token in the header.

GET: /api/online/offer/v1/eligible?amount=12000

Property name	Mandatory	type	Possible values
amount	true	Query param	Currency: IRR

If the answer is true in the service, the merchants can show the credit option to the user, otherwise, it is not possible.

Sample of response:

```
{
  "response": {
    "eligible": true,
    "title_message": "string"
    "description": "string"
  },
  "successful": true
}
```


Purchase process

The purchase process is such that the user selects the credit purchase option. First, the merchant must call the token service with the requested information and receive a token for this purchase from Snapp! Pay. After receiving the token, the purchase process is practically started.

The rest of the user's activities are performed by this token.

After receiving the token, the Merchant must redirect the user to the Snapp! Pay single purchase page. On this page, the user completes his purchase and if the purchase is successful, the Snapp! Pay server informs the Merchant server by the URL that was at the entrance of the purchase token service.

In order to complete the purchase process on the Merchant side, after receiving the purchase response from Snapp! Pay, according to the response, the verification or reverse service must be called.

In this step, if the verification service responds without any problems, the merchant can provide the goods to the customer. To finalize the Merchant purchase, it is required to call the settle Service by the end of the day.

Get Payment token service

As explained in the previous section, you must first receive a token to start the purchase.

Notice: For the sake of security only requests are acceptable whose returnUrl matches with the defined base url that you provided at service register time.

POST:/api/online/payment/v1/token

Sample of request:

```
{
  "amount": 0,
  "cartList": [
    {
      "cartId": 0,
      "cartItems": [
        {
          "amount": 0,
          "category": "string",
          "count": 0,
          "id": 0,
          "name": "string",
          "commissionType": 0
        }
      ],
      "isShipmentIncluded": true,
      "isTaxIncluded": true,
      "shippingAmount": 0,
      "taxAmount": 0,
      "totalAmount": 0
    }
  ],
  "discountAmount": 0,
  "externalSourceAmount": 0,
  "mobile": "string",
  "paymentMethodTypeDto": "INSTALLMENT",
  "returnURL": "string",
  "transactionId": "string"
}
```

Property name	Mandatory	Note	Possible values
amount	true	Total amount of purchase	Currency: IRR
discountAmount	false		Currency: IRR
¹ externalSourceAmount	false		Currency: IRR
mobile	false	User mobile number	
paymentMethodTypeDto	true		INSTALLMENT
returnURL	true	Snapp! Pay redirect user to return url (should be post)	
transactionId	true	Unique id for each transaction	
cartList	true		
cartList - cardId	true	Cart identifier	
cartList - isShipmentIncluded	false		
cartList - isTaxIncluded	false		
cartList - shippingAmount	false	Shipping amount of cart	Currency: IRR
cartList - taxAmount	false	Tax amount of cart	Currency: IRR
cartList - totalAmount	true	Total amount of cart	Currency: IRR
cartItems	true		
cartItems - id	true	Cart item identifier	
cartItems - amount	true		Currency: IRR
cartItems - category	true		
cartItems - count	true		1,2,3,...
cartItems - name	true		
cartItems - commissionType	false	Business confirm needed	1,2,3,...

¹ external source amount comes from your website wallet, account balance, gift card, coupon, or their total that user can pay the order amount simultaneously by Snapppay services and them.

Sample of response:

```
{
  "response": {
    "paymentToken": "string"
    "paymentPageUrl": "string"
  },
  "successful": true
}
```

After getting token you have to redirect user to paymentPageUrl

Verify

After Snapp! Pay service redirect users to your return Url, for verifying that you get the response of withdrawal you should call verify or revert service based on the response that you get in the return URL. Snapp! Pay service submits a form with below parameters:

Parameter name	Possible value	Description
transactionId	xxxxxx-xxxxx	Unique identifier which is sent in get payment token service
state	OK/FAILED	OK : purchase process has been done successfully and needs to call verify or revert services to finalize or revert the transaction FAILED : purchase process has been failed and needs to call revert service to revert the transaction
amount	0 (IRR)	

Notice: This service should be called just once on redirecting users from IPG even when there were multiple callbacks from IPG, so you should arrange that a mechanism to prevent further calls.

Post: `/api/online/payment/v1/verify`

Sample of request:

```
{
  "paymentToken": "string",
}
```

Sample of response:

```
{
  "response": {
    "transactionId": "string"
  },
  "successful": true
}
```

Settle

In order to complete the process of purchase, you as an API caller should call settle service, after this step, you cannot revert the transaction anymore.

Post: /api/online/payment/v1/settle

Sample of request:

```
{
  "paymentToken": "string",
}
```

Sample of response:

```
{
  "response": {
    "transactionId": "string"
  },
  "successful": true
}
```

Revert

After a successful withdrawal request, you as an API caller can revert the transaction. Be careful that you cannot revert the transaction after you settle the transaction.

Post: /api/online/payment/v1/revert

Sample of request:

```
{  
  "paymentToken": "string",  
}
```

Sample of response:

```
{  
  "response": {  
    "transactionId": "string"  
  },  
  "successful": true  
}
```

Get Payment Status

In order to inquire about a purchase status, and decide on a further operation you can call this API.

Get: `/api/online/payment/v1/status?paymentToken=[payment_token]`

Sample of response:

```
{
  "response": {
    "transactionId": "string",
    "status": "string",
    "amount": 0
  },
  "successful": true
}
```


Cancel

After a successful settle request, you as an API caller can cancel the transaction.

Post: /api/online/payment/v1/cancel

Sample of request:

```
{  
  "paymentToken": "string",  
}
```

Sample of response:

```
{  
  "response": {  
    "transactionId": "string"  
  },  
  "successful": true  
}
```

Update

After a successful settle request, you as an API caller can update the transaction with an amount which should be lower than the previous one.

Post: /api/online/payment/v1/update

Sample of request:

```
{
  "amount": 0,
  "cartList": [
    {
      "cartId": 0,
      "cartItems": [
        {
          "amount": 0,
          "category": "string",
          "count": 0,
          "id": 0,
          "name": "string",
          "commissionType": 0
        }
      ],
      "isShipmentIncluded": true,
      "isTaxIncluded": true,
      "shippingAmount": 0,
      "taxAmount": 0,
      "totalAmount": 0
    }
  ],
  "discountAmount": 0,
  "externalSourceAmount": 0,
  "paymentMethodTypeDto": "INSTALLMENT",
  "paymentToken": "string"
}
```

Property name	Mandatory	Note	Possible values
amount	true	Total amount of purchase	Currency: IRR
discountAmount	false		Currency: IRR
externalSourceAmount	false		Currency: IRR
paymentMethodTypeDto	true		INSTALLMENT
paymentToken	true	Unique token of the payment	
cartList	true		
cartList - cardId	true	Cart identifier	
cartList - isShipmentIncluded	false		
cartList - isTaxIncluded	false		
cartList - shippingAmount	false	Shipping amount of cart	Currency: IRR
cartList - taxAmount	false	Tax amount of cart	Currency: IRR
cartList - totalAmount	true	Total amount of cart	Currency: IRR
cartItems	true		
cartItems - id	true	Cart item identifier	
cartItems - amount	true		Currency: IRR
cartItems - category	true		
cartItems - count	true		1,2,3,...
cartItems - name	true		
cartItems - commissionType	false		1,2,3,...

Sample of response:

```
{
  "response": {
    "transactionId": "string"
  },
  "successful": true
}
```

Error

Http Status Code	ErrorCode	Note	Services
500	1000	Server internal error	All
401	1003	Invalid token	All
409	1008	Transaction is duplicated	Token
400	1011	Invalid transaction status	Verify Settle Revert
401	1013	Venture is blocked	All

Appendix(online merchant purchase flow)

