

Project : Nummus Payment Gateway  
Subject : Payment Gateway API  
Date : April 2020  
Version : 1.1



The NUMMUS **PAYMENT GATEWAY** has 3 service points for interacting with external parties :

1. Authorization token
2. Post payment / Notification
3. Query payment status

**Process Flow**

1. External party requests a token by supplying account credentials.
2. On account verification success the external party receives an authorization token.
3. Post the payment info along with the received token.

The token will expire after a fixed amount of time or when used in payment.

## API Service Points

### [ 1 ] Authorization Token

---

URL :

<https://smarptest.online/nummus/service/WebshopAuth/user@email.com/password/invoice#/amount>

Fail:

```
{ "Code":"1","Resp":"SERVICE_OFFLINE" }
```

Success:

```
{ "Code":"0","Resp":"1d3496824b626ed7cf712512ec42e3c3" }
```

Code	Description
0	Success (Got Token)
1	Service Offline
2	Account Violation (IP Black-Listed)
3	Input Not Allowed (Dirty/SQL)
4	Account Inactive
5	Password Incorrect

**Table 1: Token API Response Code Values**

## [ 2 ] POST Payment / Notification / returnUrl [ METHOD: POST ]

---

URL :

<https://smarttest.online/nummus/service/ScanController.php?TokenID=&Desc=/service/ScanController.php?TokenID=&Desc=>

Code	Description
0001A	Incoming Data From External Party Is Dirty (SQL/XSS_Inject)
0001B	Invalid Auth. Token (Empty Token Or Garbage/Guessing)
0001C	Authorization Page Expired
0001D	Empty Variable Main Submit Form

**Table 2: Payment Gateway API Response Code Values**

Name	Description
TokenID	Authorization Token
Desc	Purchase description
returnURL	Callback url for status updates (pending / cancel / paid)

**Table 3: POST Parameter Name Values**

All parameter names are case sensitive.

### Notification

The payment gateway will use the returnUrl for call backs by appending the api key, invoice# & status to it before calling the url. The returnUrl needs to be sent as url\_encoded.

### Web API Credentials

Make sure to activate the web-api by selecting the FREE bundle in the Billing page (Web API). The user can request the api credentials from within the Nummus account on Settings page, Web API credentials → **Send Email** button. If you want to reset your api credentials check the Reset box and click the **Send Email** button.

On success the user will receive api credentials via email. The Api Key will be used to verify the notification at the external party to prevent fraudulent notifications.

External party sends this :

returnURL : <http://some-domain-name/path/to/service>

Payment gateway calls this url :

<http://some-domain-name/path/to/service/key/invoice#/status>  
(status = pending / paid / cancel / failed)

On payment submit to gateway, the gateway will reply with :

<http://some-domain-name/path/to/service/key/invoice#/pending>

When the user completes the payment process successfully, the gateway will send :

<http://some-domain-name/path/to/service/key/invoice#/paid> (**notification**)

The user will be **REDIRECTED** to the returnUrl by the payment gateway :

[http://some-domain-name/path/to/service/0/invoice#/success/payment\\_gateway\\_message](http://some-domain-name/path/to/service/0/invoice#/success/payment_gateway_message)

When the payment process fails, the gateway will send :

<http://some-domain-name/path/to/service/key/invoice#/failed> (**notification**)

The user will be **REDIRECTED** to the returnUrl by the payment gateway :

[http://some-domain-name/path/to/service/0/invoice#/failed/payment\\_gateway\\_message](http://some-domain-name/path/to/service/0/invoice#/failed/payment_gateway_message)

payment\_gateway\_message = A text message describing the status of the transaction.

i.e: Approved / Not Enough Funds etc...

When the gateway redirects the user to the success or failed url the key will be 0 or not shown at all, so as not to expose the api key to the user.

The external party sends a returnUrl with the payment info.

The returnUrl can have the form of :

1. <http://domain-name/path/to/service> (**REST-API**)
2. <http://domain-name/path/to/orderstatus.php> (**Interface File**)

When the returnUrl has the form of orderstatus.php :

Payment gateway calls this url for notification

<http://some-domain-name/path/to/orderstatus.php?key=&invoice=&status=>

Payment gateway calls this url for REDIRECT back to merchant site:

<http://some-domain-name/path/to/orderstatus.php?invoice&status=&message=>

Status	Description
pending	Payment process initiated, waiting user approval
paid	Payment success
failed	Payment failed
cancel	Payment canceled

**Table 4: Callback Status Description**

**Post Payment Example :**

[https://smarttest.online/nummus/service/ScanController.php?](https://smarttest.online/nummus/service/ScanController.php?TokenID=1d3496824b626ed7cf712512ec42e3c3&Desc=Some_Item_description&returnURL=http%3A%2F%2Fdemo.web.com%2Fpayments%2Fservice)

[TokenID=1d3496824b626ed7cf712512ec42e3c3&Desc=Some\\_Item\\_description&returnURL=http%3A%2F%2Fdemo.web.com%2Fpayments%2Fservice](https://smarttest.online/nummus/service/ScanController.php?TokenID=1d3496824b626ed7cf712512ec42e3c3&Desc=Some_Item_description&returnURL=http%3A%2F%2Fdemo.web.com%2Fpayments%2Fservice)

### [ 3 ] Query Payment Status

---

An external party can query the gateway for the status of a payment by supplying an invoice# and the date (formatted as DDMMYY). If no date is given or the date format is incorrect the gateway will assume the current date and look for a transaction that matches the criteria with the current date.

URL :

[https://smarttest.online/nummus/service/PaymentStatus/user@email.com/password/invoice#/date\(DDMMYY\)](https://smarttest.online/nummus/service/PaymentStatus/user@email.com/password/invoice#/date(DDMMYY))

**Fail:**

```
{ "Code": "1", "Resp": "SERVICE_OFFLINE" }
```

**Success:**

```
{ "Code": "0", "Txn_Number": "123456", "Status": "APPROVED", "RespCode": "00", "Email": "user@mail.com", "Amount": "6.00", "Time": "18-09-2017 22:54:02" }
```

**No Records Found:**

```
{ "Code": "6", "Resp": "EMPTY_DATA_SET" }
```

**Not unique invoice# & multiple records**

In case an invoice# is used more than once, that may be the case for the utility companies if a customer's payment is not successful (for example : Exceeds Funds Available). The same invoice# can be submitted again for payment processing.

When the status query is run the response will be a JSON object named "records" with multiple arrays :

```
{ "records":  
{  
  1:[{ "Txn_Number": "000001", "Status": "EXCEEDS FUNDS  
AVAILABLE", "RespCode": "51", "Email": "user@msn.com", "Amount": "20000.00", "Time": "27-03-  
2018 11:10:00" }],  
  2:  
  [{ "Txn_Number": "838524", "Status": "APPROVED", "RespCode": "00", "Email": "user@msn.com", "  
Amount": "1.00", "Time": "27-03-2018 11:11:17" }]  
}  
}
```

Code	Description
0	Success
1	Service Offline
2	Account Violation (IP Black-Listed)
3	Input Not Allowed (Dirty/SQL)
4	Account Inactive
5	Password Incorrect
6	No Records Found

**Table 5 : Payment Status API Response Code Values**

Name	Description
Txn_Number	Authorization number from gateway
Status	Text description of response
RespCode	Txn response code
Email	User account
Time	Txn time
Amount	Txn amount
<b>Note</b>	<b>RespCode 00 is success, everything else is failed.</b>

**Table 6 : Payment Status Query Response**

#### **[4] Common issues connecting to the gateway**

---

Error	Description
Account Violation IP Black-Listed	Your IP is black listed due to security concerns
Invalid External Data	POST data submitted contains special chars or sql keywords used in injection attacks
Input Not Allowed (Dirty/SQL)	POST data submitted contains special chars or sql keywords used in injection attacks

**Table 7 : Errors connecting to the gateway**

The payment gateway is **NOT** redirecting the user after a transaction is **authorized**. This happens when the submitted returnUrl parameter is empty or the submitted returnUrl contains invalid data.

Always make sure the returnUrl is [url encoded](#) before submitting to the gateway. When the returnUrl contains invalid data, the gateway scrubs the url and no returnUrl is provided for the rest of the process.

If the gateway detects invalid data, the connecting IP will be banned for 1 hour. If this keeps happening regularly, the IP may be banned for a longer period.

Always [sanitize](#) your data before submitting to the gateway.

Variable	Restriction
Description	MAX 100 chars
Invoice#	MAX 40 chars
Return URL	MAX 1000 chars

**Table 8 : Max length payment post data**

When posting payment data to the ScanController make sure these variables are within the mentioned limits. If not, they will be truncated to the maximum allowable length.