OPTIMAL™ PAYMENTS

NETELLER REST API v1

Isle of Man

Registered Office & Headquarters:
Audax House
6 Finch Road
Douglas
IM1 2PT
Isle of Man
Tel: +44 (0) 1624 698700
Fax: +44 (0) 1624 615320

Europe

10 Albemarle Street
London
W1S 4BL
United Kingdom
Tel: +44 (0) 1223 44 6020
Fax: +44 (0) 1223 44 6021

Optimal Payments Ltd
Compass House
Vision Park
Chivers Way
Cambridge
CB24 9AD
United Kingdom
Tel: +44 (0) 1223 238 002
Fax: +44 (0) 1223 238 021

North America

3rd floor, 215 16th Street SE
Calgary, Alberta
T2E 7P5
Canada
Tel: +1 403 769 3600
Fax: +1 403 277 4266

2 Place Alexis-Nihon
3500 de Maisonneuve Blvd. W., Suite 700
Montreal, Quebec
H3Z 3C1
Canada
Tel: +1 514 380 2700, +1 866 826 8003
Fax: +1 514 380 2760

75 Promenade du Portage
Gatineau, Quebec
J8X 2J9
Canada
Tel: +1 514 380 2700, +1 866 826 8003
Fax: +1 514 380 2760

NETELLER REST API Reference

The NETELLER REST API offers a complete modern replacement to the legacy Quick Signup, NetDirect, InstantPayout and NetCheck APIs. The REST (REpresentational State Transfer) style API provides predictable, resource based URI's that removes the complexity that often comes with implementing traditional SOAP or XML driven APIs. The NETELLER REST API works with JSON request and response object notation and will return appropriate HTTP response codes which can be understood by browsers and HTTP clients.

This API leverages OAUTH2 as an authorization framework.

⚠ **cURL Samples**
The cURL samples included in this documentation are for reference purposes only. Substitute your own values as appropriate. You can also refer to the online cURL documentation for additional information on how to use cURL.

The following flags are optional and may be useful sandbox testing:

-k : *(SSL) This option explicitly allows curl to perform "insecure" SSL connections and transfers. All SSL connections are attempted to be made secure by using the CA certificate bundle installed by default. This makes all connections considered "insecure" fail unless -k, --insecure is used. THIS SHOULD NEVER BE USED IN A PRODUCTION IMPLEMENTATION!*

Note that if you only want HTTP headers in the output, -i, --include might be the option you're looking for.

-v : *Makes the fetching more verbose/talkative. Mostly useful for debugging. A line starting with '>' means "header data" sent by curl, '<' means "header data" received by curl that is hidden in normal cases, and a line starting with '*' means additional info provided by curl.*

Changelog

⚠ The API minor version will only be incremented when NETELLER introduce's a backwards incompatible change. You can maintain your API version through the merchant account portal. Please be advised that outdated minor versions will be deprecated after a rolling 12 month period.

API Version	Minor Version	Patch Description
v1	2014-11-04	*2014-12-09* • Added support for NETELLERGO! • Added order, paymentMethod, redirect resources • Modified invoice resource to support order invoices • Added order lookup • Added lookup invoice for an order • Added error codes: 20016 *2014-11-04* • Renamed account resource to customer • Updated customer lookup to new resource uris (/v1/account -> /v1/customer) • Removed linkbackurl from accountProfile resource and moved to root of create customer request • Return customer resource instead of accountProfile on payment, subscriptions, invoices resources • Renamed subscriptionPlan resource to plan • Renamed invoice amountDue to totalAmount • Lists no longer contain un-neccesary object wrapper, rather returns a list of the appropriate object.

v1	2014-04-15	*2014-09-30* • Added API minor version support *2014-09-03* • Added OAUTH2 authorization grant • Added account, balance resources • Added account lookup functionality • Added new scopes: subscription_payment, account_enhanced_profile, account_contacts, account_balance • Modified create subscription to require scope: subscription_payment *2014-07-22* • Added Quick Signup functionality *2014-06-17* • Added ability to list subscription invoices and subscription plans • Added subscription invoice lookup • Added subscription webhooks • Added error codes: 20050 • Added new languages Greek (el_GR), Korean (ko_KR) • Added new subscription status: pending *2014-05-21* • Added ability to list invoices for a subscription • Added support for resource expansion • Added new currencies: Malaysian Riggit (MYR), Moroccan Dirham (MAD), Tunisia Dinar (TND) and Swiss Franc (CHF) • Added error codes: 20102 *2014-04-15* • Modified authentication to OAUTH2 authentication using client_credentials • Added subscription billing support • Added error codes: 20100, 20101

2014-11-04 Upgrade Notes

The following page outlines considerations for review when upgrading your REST API from v1 (2014-04-15) to v1 (2014-12-09).

✓ You can trial your upgraded integration with your TEST sandbox account prior to upgrading in production. Contact NETELLER Merchant Support if you require assistance with your sandbox account.

Upgrade Procedure

1. Modify any calls to /v1/accounts to the new /v1/customers resource
   a. If you are looking up profile details using the the former call /v1/accounts/{accountId} this should now change to this format /v1/customers/?accountId={accountId}.
   b. To query for a profile using email, the same convention is used but the resource uri should change as follows: /v1/customers/?email={email}.
   c. When creating accounts, you now post an accountProfile and receive a customer object in response. If linkbackurl was previously supplied as part of the accountProfile, this has been moved to the root of that request and is no longer part of the accountProfile.
2. Ensure any embedded references to accountProfile information are updated to refer to customer details

API Endpoints

The NETELLER REST APIs are available in the NETELLER Sandbox environment for integration testing purposes.   To switch between the sandbox and the live production system you only need change the endpoint URI and the credentials.

The following endpoints form the basis of a resource URI:

- Sandbox (Testing) endpoint: https://test.api.neteller.com
- Production endpoint: https://api.neteller.com

Versioning

The first time you make an API request, NETELLER will automatically record the the current API version and associate it with your merchant account.

The API version will influence the request/response object structure as well as the functionality that is available through the API.   A new version of the API will only be introduced if we roll out a backwards incompatible change that may impact your existing integrations. As your API version can be maintained through the merchant portal, you can manage when to upgrade to the latest version, helping ensure your integration is not disrupted as new functionality becomes available.

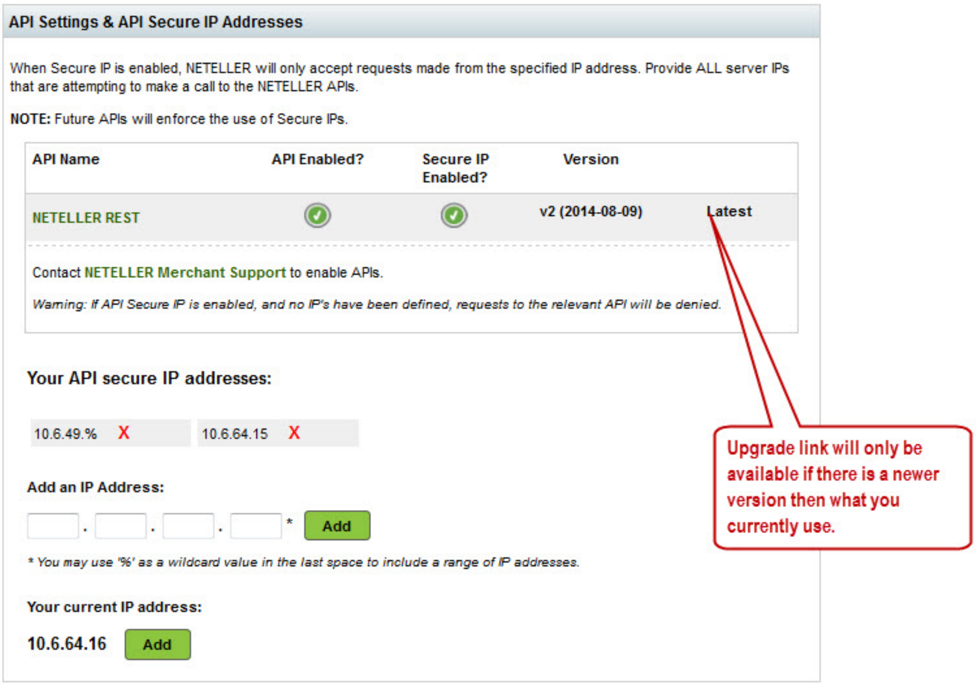
⚠ **Outdated minor versions of the API will be supported for 12 months, after which point they may be deprecated and you will be required to upgrade to a newer version.**

Major Version

The major version (ie: /**v1**/) represents a major technology shift between versions of the API.   Changes in the major version are reserved for large scale changes to the API that fundamentally change integration requirements from the previous version. Modified business domain models, re-design of functional flows or change in API resource patterns are all considered major changes. Changes to major version are rare and typically require an extended deprecation cycle (~2 years) from prior versions to allow all consumers of the API to adapt.

Minor Version

The minor version is a date stamped version of the API indicating when a backwards incompatible change was released in production. Only changes that are deemed backwards incompatible will trigger a new minor version.   Other changes that are considered backwards compatible are integrated in the last version and rolled out as patches as soon as they become available. The minor version is NOT specified as part of the URL request, it is defined as part of your merchant account and can be controlled through your merchant account portal.



API Settings & API Secure IP Addresses

When Secure IP is enabled, NETELLER will only accept requests made from the specified IP address. Provide ALL server IPs that are attempting to make a call to the NETELLER APIs.

NOTE: Future APIs will enforce the use of Secure IPs.

API Name	API Enabled?	Secure IP Enabled?	Version	
NETELLER REST	✓	✓	v2 (2014-08-09)	Latest

Contact **NETELLER Merchant Support** to enable APIs.

Warning: If API Secure IP is enabled, and no IP's have been defined, requests to the relevant API will be denied.

Your API secure IP addresses:

10.6.49.% X 10.6.64.15 X

Add an IP Address:

** You may use '%' as a wildcard value in the last space to include a range of IP addresses.*

Your current IP address:

10.6.64.16

Upgrade link will only be available if there is a newer version then what you currently use.

What is considered a backwards compatible change?

- Adding API resources or adding new properties to existing API resources
- Modification to the order of parameters or JSON object elements
- Adding optional request parameters to API methods
- Adding webhook event types

What is considered a backwards incompatible change?

- Removal of a API resource
- Removal of an API Resources object attribute
- Changes to required parameter / JSON object elements
- Removal of previously supported API call

Pagination

REST calls that return multiple resources will wrap the response as a JSON array named 'list'. By default, 10 items will be returned at a time and you can navigate to the next page of data using the returned HATEOAS links.

> ✓ You can change the default list page size by defining a **limit** parameter value.
>
> For example, if a request to look up subscriptions identifies that there are 134 subscriptions and you have set your limit to 15, the first 15 items will be returned with navigation links to the next 15 items in the list (items 16-30). If you navigate to the next page using the provided HATEOAS link, you will now retrieve items 16-30 with a new navigation link provided to return to the previous page, as well as a link to continue to the next page of items.

Parameter Name	Required?	Description
limit	**optional**	The number of records to be returned per page. Default = 10, Max = 100
offset	**optional**	Identifies your location within the list and allows you to fetch the next set of resources

Sample Response with List Data

```
HTTP/1.1 204 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
    "list": [
        {
            "planId": "MONTHLYGREENPLAN",
            "planName": "Sample Premier Monthly Membership",
            "interval": 3,
            "intervalType": "monthly",
            "intervalCount": 4,
            "amount": 2995,
            "currency": "EUR",
            "status": "active"
        },
        {...},
        {...},
    ],
    "links": [
        {
            "url": 
"https://api.neteller.com/v1/plans?sessionid=234243242&limit=10&offset=10",
            "rel": "self",
            "method": "GET"
        },
        {
            "url": 
"https://api.neteller.com/v1/plans?sessionid=234243242&limit=10&offset=0",
            "rel": "prev_offset",
            "method": "GET"
        },
        {
            "url": 
"https://api.neteller.com/v1/plans?sessionid=234243242&limit=10&offset=20",
            "rel": "next_offset",
            "method": "GET"
        }
    ]
}
```

Resource Expansion

Resource expansion can be compared to eager loading in the database realm. It offers a way to load a specific resource as well as other resources related to it. You are encouraged to use resource expansion when you know you are going to need the extra information. This will reduce the amount of API calls as well as the amount of traffic to and from your server.   You should avoid resource expansion when you want to reduce the amount of information returned or if you are unsure if you will need the additional resource object.

By adding an **expand** query parameter to the query you are effectively telling the server to include the details for the reference object in-line, in a single response. Anytime you encounter an object whose only child attribute is a link, you can consider that object expandable. Passing that object name as part of your expand parameter list will automatically substitute the link with the relevant details.

✓ For resources that have multiple expandable children, you can specify which resources to expand in a comma separated list. Eg.   **?expand=resource1,resource2**

For example, below is a sample subscription lookup. By default the plan is not expanded and instead includes a reference link that would provide the relevant details.

```
 {
    "subscriptionId": "234234224",
    "plan": {
        "link": {           {
                "url": "https://api.neteller.com/v1/plan/MONTHLYGREENPLAN",
                "rel": "plan",
                "method": "GET"
        }
    },
    "accountProfile": {
        "email": "johndoe@email.com",
        "accountId": "451234567890",
        "firstName": "John",
        "lastName": "Doe",
        "country": "CA",
        "dateOfBirth": {
            "year": "1975",
            "month": "01",
            "day": "31"
        }
    },
    "status": "active",
    "startDate": "2014-06-01T00:00:00Z",
    "endDate": "2014-12-31T00:00:00Z",
    "termEndDate": "2014-12-31",
    "currentPeriodStart": "2014-07-01T00:00:00Z",
    "currentPeriodEnd": "2014-07-31T00:00:00Z",
    "cancelAtPeriodEnd": false,
    "cancelDate": null,
    "lastCompletedPaymentDate": "2014-06-31T00:00:00Z",
    "links": [
        {
            "url": "https: //api.neteller.com/v1/subscriptions/234234224",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

If you would rather have the full subscription plan object returned, then add the '**expand=plan'** parameter to your request and you will see the following result:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
    "subscriptionId": "234234224",
    "plan": {
        "planId": "MONTHLYGREENPLAN",
        "planName": "Sample Premier Monthly Plan",
        "interval": 3,
        "intervalType": "monthly",
        "intervalCount": 4,
        "amount": 2495,
        "currency": "EUR",
        "status": "active"
    },
    "accountProfile": {
        "email": "johndoe@email.com",
        "accountId": "451234567890",
        "firstName": "John",
        "lastName": "Doe",
        "country": "CA",
        "dateOfBirth": {
            "year": "1975",
            "month": "01",
            "day": "31"
        }
    },
    "status": "active",
    "startDate": "2014-06-01T00:00:00Z",
    "endDate": "2014-12-31T00:00:00Z",
    "termEndDate": "2014-12-31",
    "currentPeriodStart": "2014-07-01T00:00:00Z",
    "currentPeriodEnd": "2014-07-31T00:00:00Z",
    "cancelAtPeriodEnd": false,
    "cancelDate": null,
    "lastCompletedPaymentDate": "2014-06-31T00:00:00Z",
    "links": [
        {
            "url": "https: //api.neteller.com/v1/subscriptions/234234224",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

Getting Started

Overview

This section outlines some common use scenarios of the NETELLER REST API.

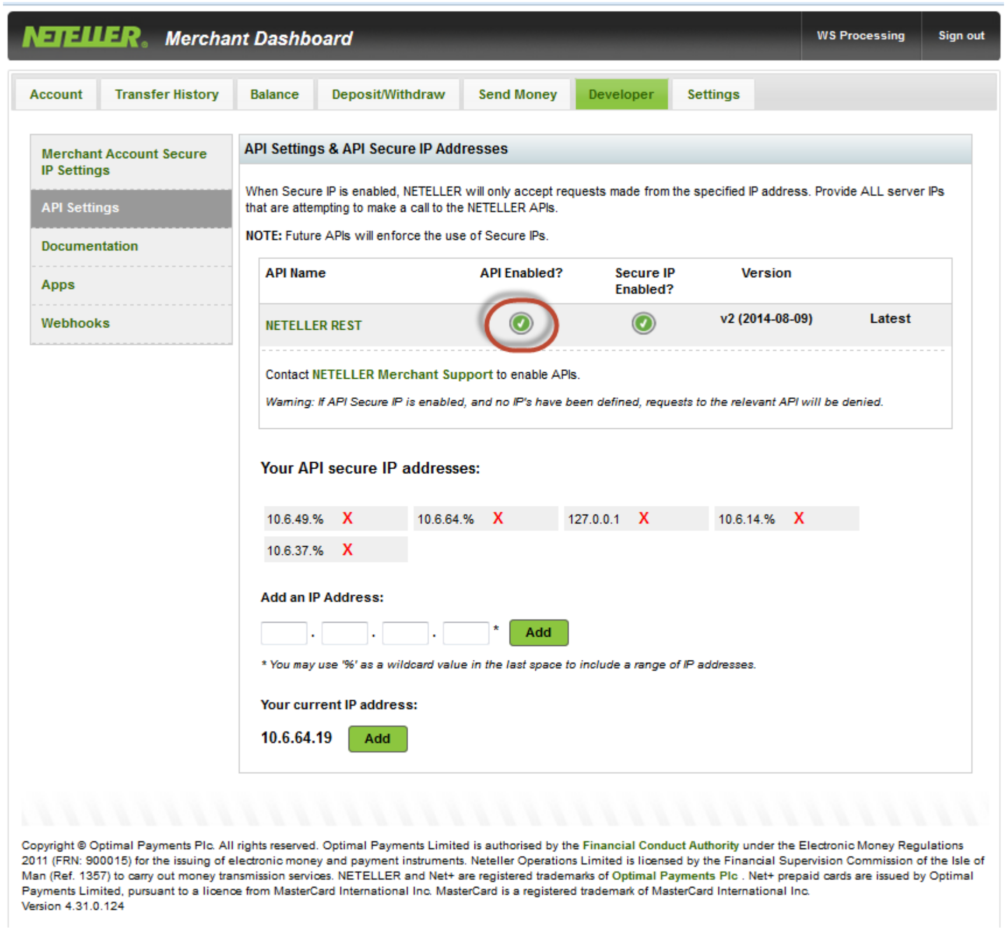
All examples in this section use cURL, which is a command line tool for transferring data using protocols such as HTTPS.

Obtain your sandbox (test) or production account details

The NETELLER REST API implements the OAUTH2 standard for authentication. Below are the steps to follow when first setting up your application. Ths same steps should be completed on both sandbox https://test.merchant.neteller.com and production https://merchant.neteller.com environments. You must complete all of these steps ***before*** you can attempt to obtain an access token for use in your REST API request.

1) Verify that NETELLER REST API is enabled for your merchant account.

If the API is not enabled, please contact NETELLER Merchant Support to have it enabled.



API Settings & API Secure IP Addresses

When Secure IP is enabled, NETELLER will only accept requests made from the specified IP address. Provide ALL server IPs that are attempting to make a call to the NETELLER APIs.

NOTE: Future APIs will enforce the use of Secure IPs.

API Name	API Enabled?	Secure IP Enabled?	Version	
NETELLER REST	✓	✓	v2 (2014-08-09)	Latest

Contact NETELLER Merchant Support to enable APIs.

Warning: If API Secure IP is enabled, and no IP's have been defined, requests to the relevant API will be denied.

Your API secure IP addresses:

10.6.49.% X 10.6.64.% X 127.0.0.1 X 10.6.14.% X
10.6.37.% X

Add an IP Address:

□ . □ . □ . □ * Add

** You may use '%' as a wildcard value in the last space to include a range of IP addresses.*

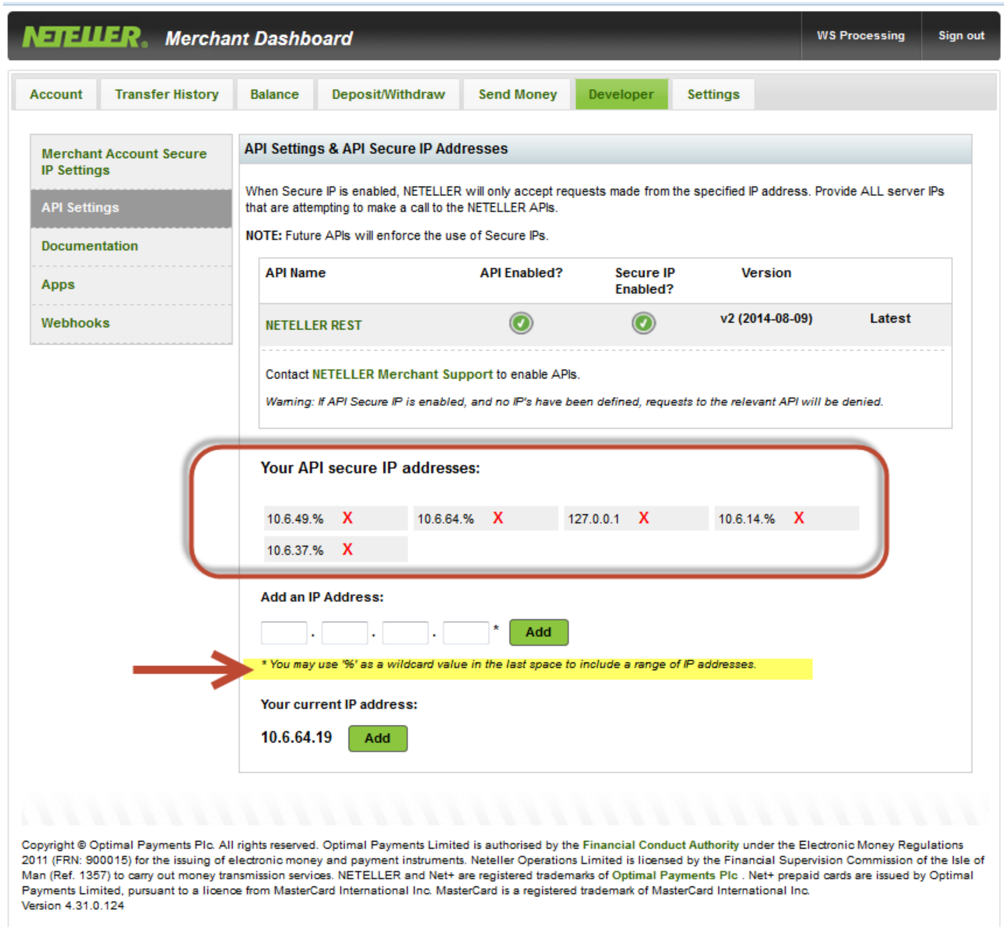
Your current IP address:

10.6.64.19 Add

Copyright © Optimal Payments Plc. All rights reserved. Optimal Payments Limited is authorised by the Financial Conduct Authority under the Electronic Money Regulations 2011 (FRN: 900015) for the issuing of electronic money and payment instruments. Neteller Operations Limited is licensed by the Financial Supervision Commission of the Isle of Man (Ref. 1357) to carry out money transmission services. NETELLER and Net+ are registered trademarks of Optimal Payments Plc . Net+ prepaid cards are issued by Optimal Payments Limited, pursuant to a licence from MasterCard International Inc. MasterCard is a registered trademark of MasterCard International Inc.
Version 4.31.0.124

2) Register the IP address(es) that will be allowed to access the API

When registering your secure API IP addresses, you must ensure you are using the IP(s) of your networks outbound gateway. If you are working within a corporate network, or are sitting behind a proxy, you may have to contact your administrator to help identify the appropriate IP addresse(s).



API Settings & API Secure IP Addresses

When Secure IP is enabled, NETELLER will only accept requests made from the specified IP address. Provide ALL server IPs that are attempting to make a call to the NETELLER APIs.

NOTE: Future APIs will enforce the use of Secure IPs.

API Name	API Enabled?	Secure IP Enabled?	Version	
NETELLER REST	✓	✓	v2 (2014-08-09)	Latest

Contact NETELLER Merchant Support to enable APIs.

Warning: If API Secure IP is enabled, and no IP's have been defined, requests to the relevant API will be denied.

Your API secure IP addresses:

10.6.49.% X 10.6.64.% X 127.0.0.1 X 10.6.14.% X
10.6.37.% X

Add an IP Address:

. . . * Add

** You may use '%' as a wildcard value in the last space to include a range of IP addresses.*

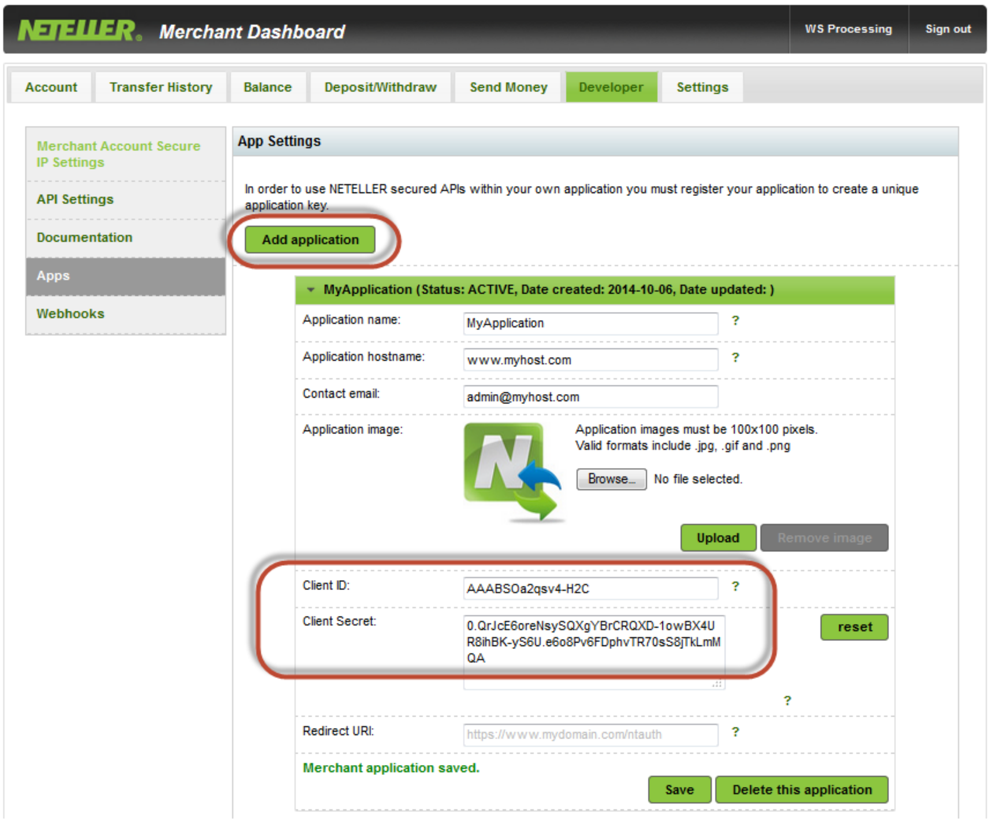
Your current IP address:

10.6.64.19 Add

Copyright © Optimal Payments Plc. All rights reserved. Optimal Payments Limited is authorised by the Financial Conduct Authority under the Electronic Money Regulations 2011 (FRN: 900015) for the issuing of electronic money and payment instruments. Neteller Operations Limited is licensed by the Financial Supervision Commission of the Isle of Man (Ref. 1357) to carry out money transmission services. NETELLER and Net+ are registered trademarks of Optimal Payments Plc . Net+ prepaid cards are issued by Optimal Payments Limited, pursuant to a licence from MasterCard International Inc. MasterCard is a registered trademark of MasterCard International Inc.
Version 4.31.0.124

3) Register your application to obtain your API credentials

You must register you application by providing the appropriate detail about its name, where it will be hosted, and what the primary contact email is to be. You may also choose to upload and application logo that will provide your customers brand recognition if you are planning to implement the authorization flow.



App Settings

In order to use NETELLER secured APIs within your own application you must register your application to create a unique application key.

MyApplication (Status: ACTIVE, Date created: 2014-10-06, Date updated:)

Application name:	MyApplication
Application hostname:	www.myhost.com
Contact email:	admin@myhost.com
Application image:	Application images must be 100x100 pixels. Valid formats include .jpg, .gif and .png No file selected.
Client ID:	AAABSOa2qsv4-H2C
Client Secret:	0.QrJcE6oreNsySQXgYBrCRQXD-1owBX4UR8ihBK-yS6U.e6o8Pv6FDphvTR70sS8jTkLmMQA
Redirect URI:	https://www.mydomain.com/ntauth

Merchant application saved.

You are now ready to obtain an access token to use in your REST API request. Please refer to the Authentication section for more information on how to obtain an access token.

Subscription (Recurring Billing) Setup

NETELLER offers recurring billing support through the REST API with the plan and subscription resources.

Create a subscription plan

To get started, you must first define your subscription plan. The plan provides details about your re-curring billing such as the charge amount, currency, frequency of the billing.   You can provide a custom plan id to identify the subscription plan, however the planid must be unique across all active subscription plans.

Sample Request

```
POST /v1/plans HTTP/1.1
Host: test.api.neteller.com
Authorization: Bearer
0.AQAAAUjshWvyAAAAAAAEk-Cj7vq1mEuIVz-2TomxIvnu.a4ixcq15cM6lCShJMIL7orHCI5w
Content-Type: application/json
Cache-Control: no-cache
Postman-Token: 01f33640-4ecb-370b-fe23-0f9b2d972b91

{
    "planId": "DEMOGOLDPLAN",
    "planName": "Sample Gold Plan",
    "interval": 1,
    "intervalType": "monthly",
    "intervalCount": 12,
    "amount": 2995,
    "currency": "EUR"
}
```

Sample Response

```
{
    "planId": "DEMOGOLDPLAN",
    "planName": "Sample Gold Plan",
    "interval": 1,
    "intervalType": "monthly",
    "intervalCount": 12,
    "amount": 2995,
    "currency": "EUR",
    "status": "active",
    "numberOfSubscriptions": 0,
    "links": [
        {
            "url": "https://api.neteller.com/v1/plans/DEMOGOLDPLAN",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

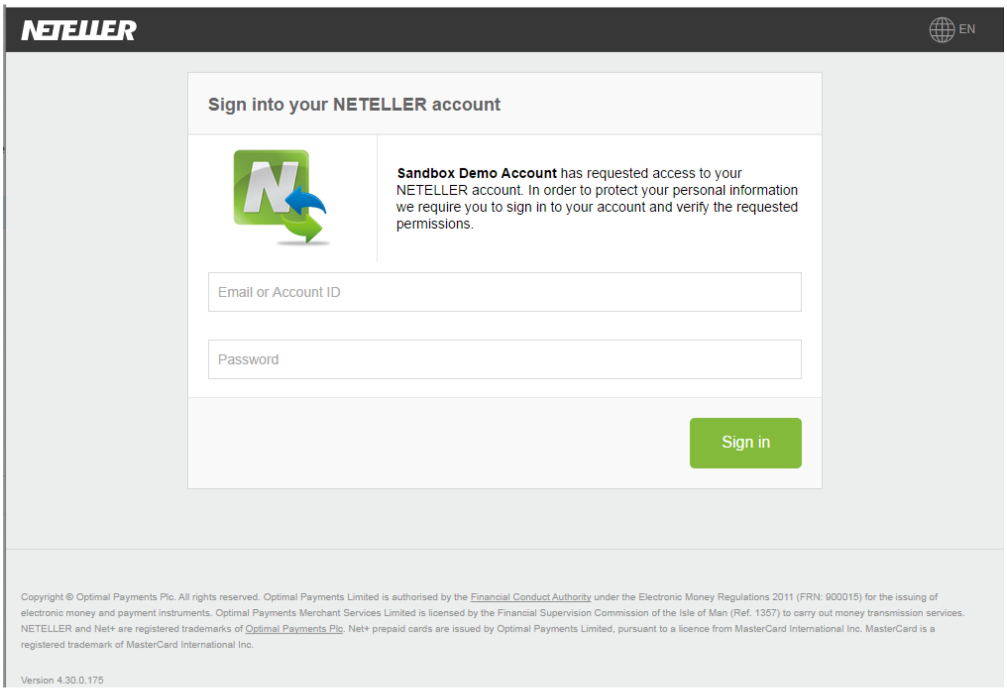
Obtain member authorization to enroll them in your subscription plan

Once you have created a plan, you will want to enroll members (subscribe them) to the plan. Before this can be done, ***you must first obtain authorization from the member*** so that NETELLER can issue re-curring payments on the member's behalf. Obtain member authorization by redirecting the customer to the hosted authorization flow with scope of 'subscription_payment', where they can approve your request.

See Obtain an access token using the auth_code grant type for more information.

Sample Request

```
GET
https://test.api.neteller.com/v1/oauth2/authorize?client_id=AAABRbRHwWkslo-u&redirect_u
ri=http://requestb.in/1kk4mrn1&scope=subscription_payment&response_type=code
```

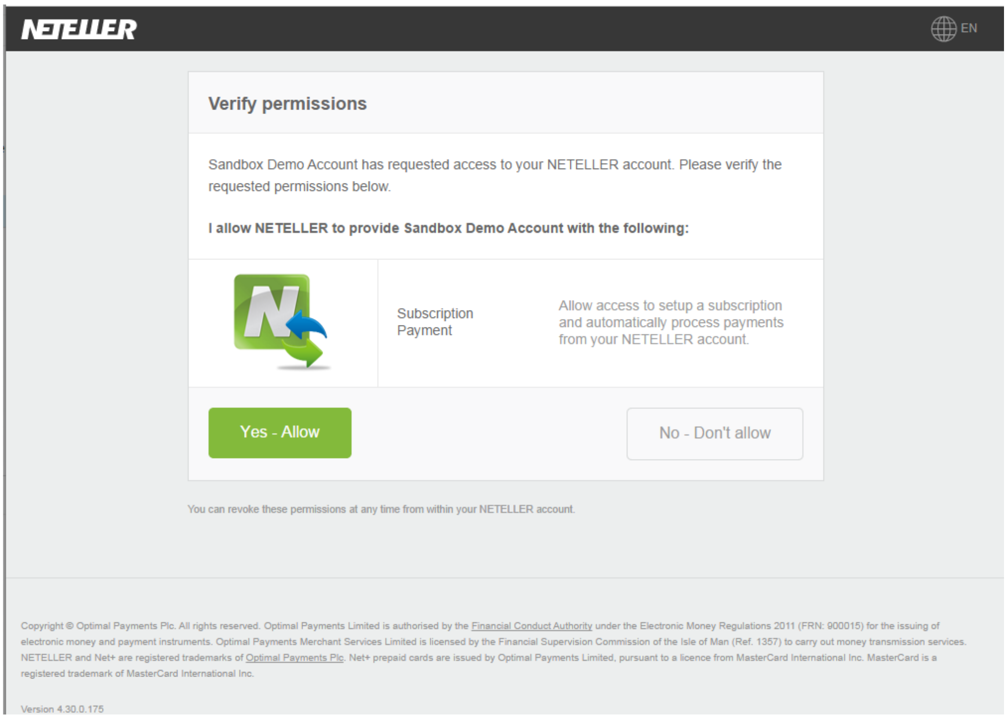


Sign into your NETELLER account

Sandbox Demo Account has requested access to your NETELLER account. In order to protect your personal information we require you to sign in to your account and verify the requested permissions.

Copyright © Optimal Payments Plc. All rights reserved. Optimal Payments Limited is authorised by the Financial Conduct Authority under the Electronic Money Regulations 2011 (FRN: 900015) for the issuing of electronic money and payment instruments. Optimal Payments Merchant Services Limited is licensed by the Financial Supervision Commission of the Isle of Man (Ref. 1357) to carry out money transmission services. NETELLER and Net+ are registered trademarks of Optimal Payments Plc. Net+ prepaid cards are issued by Optimal Payments Limited, pursuant to a licence from MasterCard International Inc. MasterCard is a registered trademark of MasterCard International Inc.

Version 4.30.0.175



Verify permissions

Sandbox Demo Account has requested access to your NETELLER account. Please verify the requested permissions below.

I allow NETELLER to provide Sandbox Demo Account with the following:

	Subscription Payment	Allow access to setup a subscription and automatically process payments from your NETELLER account.

You can revoke these permissions at any time from within your NETELLER account.

Copyright © Optimal Payments Plc. All rights reserved. Optimal Payments Limited is authorised by the Financial Conduct Authority under the Electronic Money Regulations 2011 (FRN: 900015) for the issuing of electronic money and payment instruments. Optimal Payments Merchant Services Limited is licensed by the Financial Supervision Commission of the Isle of Man (Ref. 1357) to carry out money transmission services. NETELLER and Net+ are registered trademarks of Optimal Payments Plc. Net+ prepaid cards are issued by Optimal Payments Limited, pursuant to a licence from MasterCard International Inc. MasterCard is a registered trademark of MasterCard International Inc.

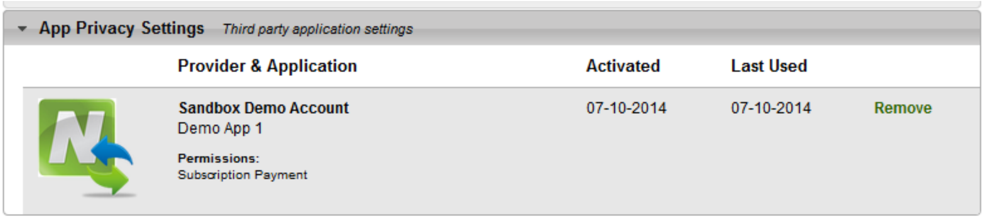
Version 4.30.0.175

Upon successful authorization, you will be returned the authorization code on your specified redirect uri.

Sample Response

```
http://requestb.in/1kk4mrn1?auth_code=0.AAAAAUjxrXbeAAAAAAAEk-AMxBtcId3NytItAJjLUG6C.JD
OulU_E5v2wSOvoqrSldXFv49Y
```

Members that have authorized your app, will now be able to see the authorization they have provided in their member account settings.



App Privacy Settings *Third party application settings*

	Provider & Application	Activated	Last Used	
	Sandbox Demo Account Demo App 1 **Permissions:** Subscription Payment	07-10-2014	07-10-2014	Remove

Subscribe a NETELLER member to your plan

Once you have obtained the required authorization, you can proceed to subscribe the member in your plan through the create subscription call.

Subscriptions can be post dated to start at a later date, but by default will start immediately and will attempt to collect payment as soon as you sent the request.

The first step in this process is to obtain a new access token. This is completed by exchanging the members authorization_code for an access token. Or by exchanging a previously issued refresh token if you have already obtained subscription_payment authorization from the NETELLER member.

Sample Request

```
POST
/v1/oauth2/token?grant_type=authorization_code&code=0.AAAAAUjxrXbeAAAAAAEk-AMxBtcId3Ny
tItAJjLUG6C.JDOulU_E5v2wSOvoqrSldXFv49Y HTTP/1.1
Host: test.api.neteller.com
Content-Type: application/json
Authorization: Basic
QUFBQlJiUkh3V2tzbG8tdTowLmtnOG1wSnhOeER5ZUx2SVNVZ04wbHFsWWVWQS1UeFZSWlVFWThIdWljOEEuUzR
uRnpPWS0wLTF3bW1qd3lGV2l0aUZwRGxz
Cache-Control: no-cache
Postman-Token: b22baf1f-256c-52c0-a52f-b6e1e04f19ec
```

The returned access token is associated explicitly with the member that you requested authorization from and can only be used for activities involving that member.

Sample Response

```
{
  "accessToken":
"0.AQAAAUjxrhKmAAAAAAA27oDRqZ5aM7mnxrr7cgwhVFt3.3RdQp5PrOlirCqhmHybKsRUBINI",
  "expiresIn": 3600,
  "refreshToken":
"0.AgAAAUjxrhKfAAAAB1jwsOBSE_xJsAYmavHS30lwvJ1B.YwlwdWKjPSa12NXvqvgV_C2922E",
  "scope": "subscription_payment"
}
```

Next, use the access token to subscribe the member to your subscription plan.

Sample Request

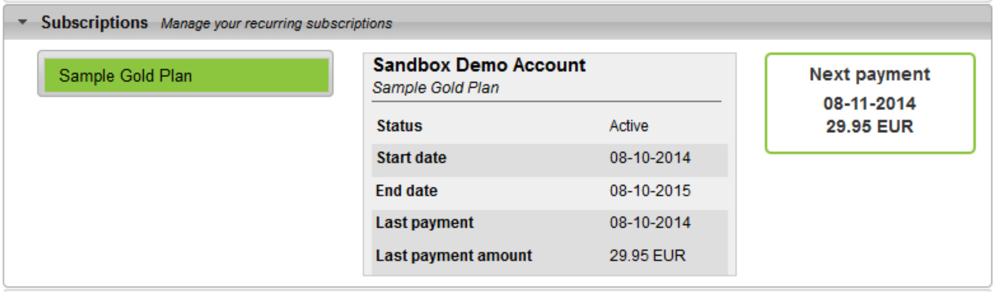
```
POST /v1/subscriptions HTTP/1.1
Host: test.api.neteller.com
Authorization: Bearer
0.AQAAAUjxrhKmAAAAAAA27oDRqZ5aM7mnxrr7cgwhVFt3.3RdQp5PrOlirCqhmHybKsRUBINI
Content-Type: application/json
Cache-Control: no-cache
Postman-Token: 5b2d14b3-963f-6ca4-d008-2952a013f7b8

{
    "planId": "DEMOGOLDPLAN",
    "accountProfile": {
        "email": "netellertest_GBP@neteller.com"
    }
}
```

If successful you will receive a response with details about the subscription, and if was set to start immediately, a HATEOAS link to the corresponding payment transaction.

Sample Response

```
{
  "subscriptionId": "46",
  "plan": {
    "link": {
      "url": "https://api.neteller.com/v1/plans/DEMOGOLDPLAN",
      "rel": "plan",
      "method": "GET"
    }
  },
  "accountProfile": {
    "email": "netellertest_gbp@neteller.com",
    "accountId": "458591047553",
    "firstName": "GBPFirstname",
    "lastName": "GBPLastname",
    "country": "GB",
    "dateOfBirth": {
      "year": "1978",
      "month": "03",
      "day": "01"
    }
  },
  "status": "active",
  "startDate": "2014-10-08T21:35:19Z",
  "endDate": "2015-10-08T21:35:19Z",
  "currentPeriodStart": "2014-10-08T21:35:15Z",
  "currentPeriodEnd": "2014-11-08T21:35:15Z",
  "cancelAtPeriodEnd": "false",
  "cancelDate": null,
  "links":
  [
    {
      "url": "https://api.neteller.com/v1/subscriptions/46",
      "rel": "self",
      "method": "GET"
    },
    {
      "url": "https://api.neteller.com/v1/payments/417412804117011",
      "rel": "payment",
      "method": "GET"
    }
  ]
}
```

Members that have subscribed to a subscription, will be able to see the subscription details in their member account settings.



Sandbox Demo Account
Sample Gold Plan

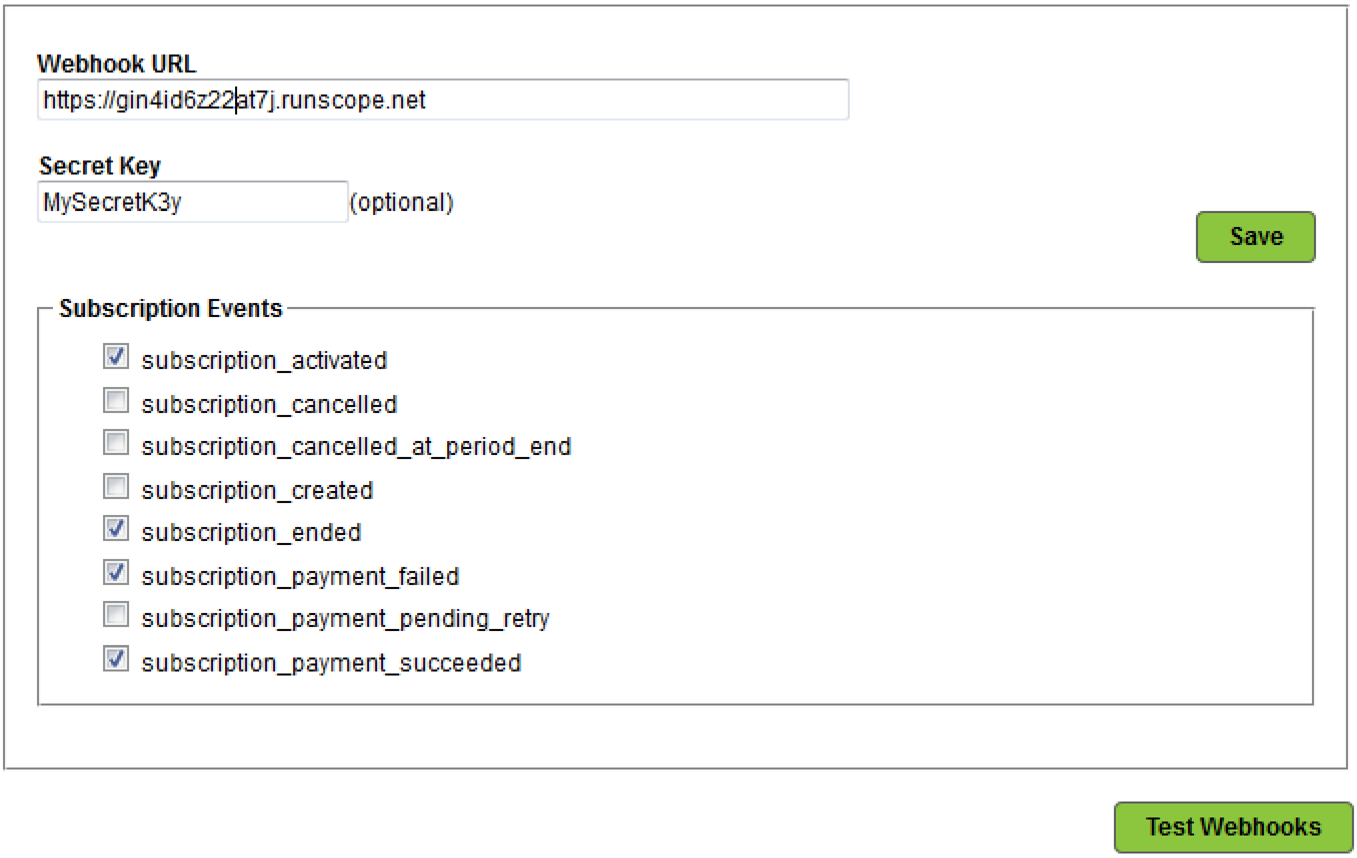
Status	Active
Start date	08-10-2014
End date	08-10-2015
Last payment	08-10-2014
Last payment amount	29.95 EUR

Next payment
08-11-2014
29.95 EUR

Setup Webhooks to receive updates about subscription payments

A webhook allows you to define an HTTP callback that will be POST to when an event occurs that you would like to be notified of. You can enable webhooks through your merchant account portal. Start by saving your webhook URL and selecting the specific events you wish to receive notification for.

See Webhooks for more information.



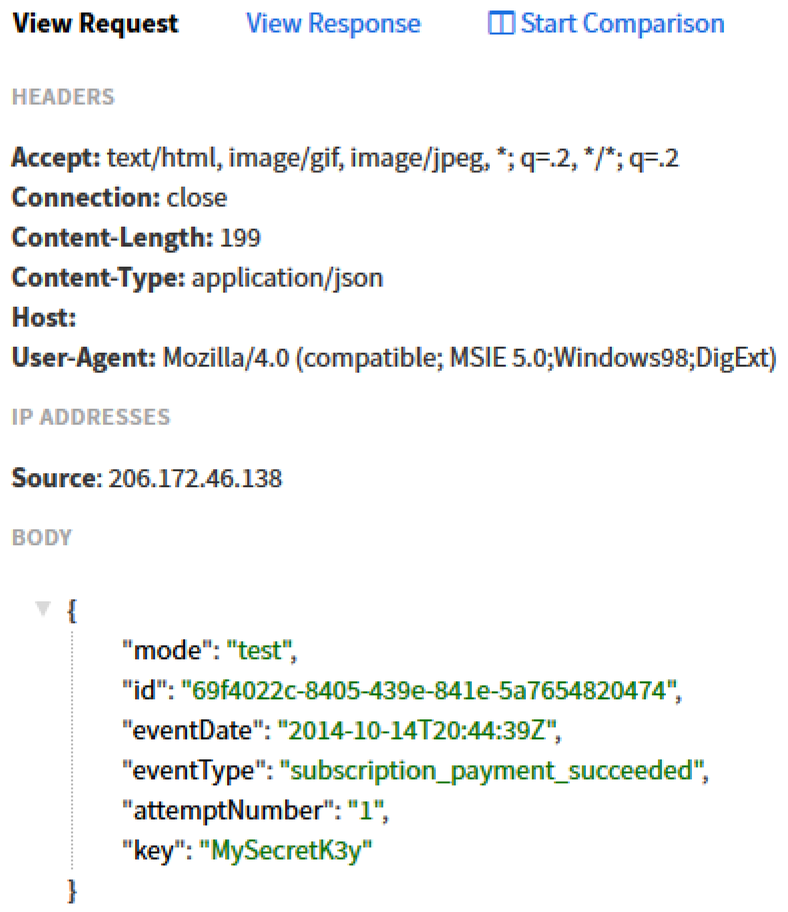
Webhook URL
https://gin4id6z22at7j.runscope.net

Secret Key
MySecretK3y (optional)

Subscription Events

☑ subscription_activated
☐ subscription_cancelled
☐ subscription_cancelled_at_period_end
☐ subscription_created
☑ subscription_ended
☑ subscription_payment_failed
☐ subscription_payment_pending_retry
☑ subscription_payment_succeeded

You can trigger sample webhook event through the Test Webhooks function to ensure your service is properly exposed to capture the event.



HEADERS

Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: close
Content-Length: 199
Content-Type: application/json
Host:
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0;Windows98;DigExt)

IP ADDRESSES

Source: 206.172.46.138

BODY

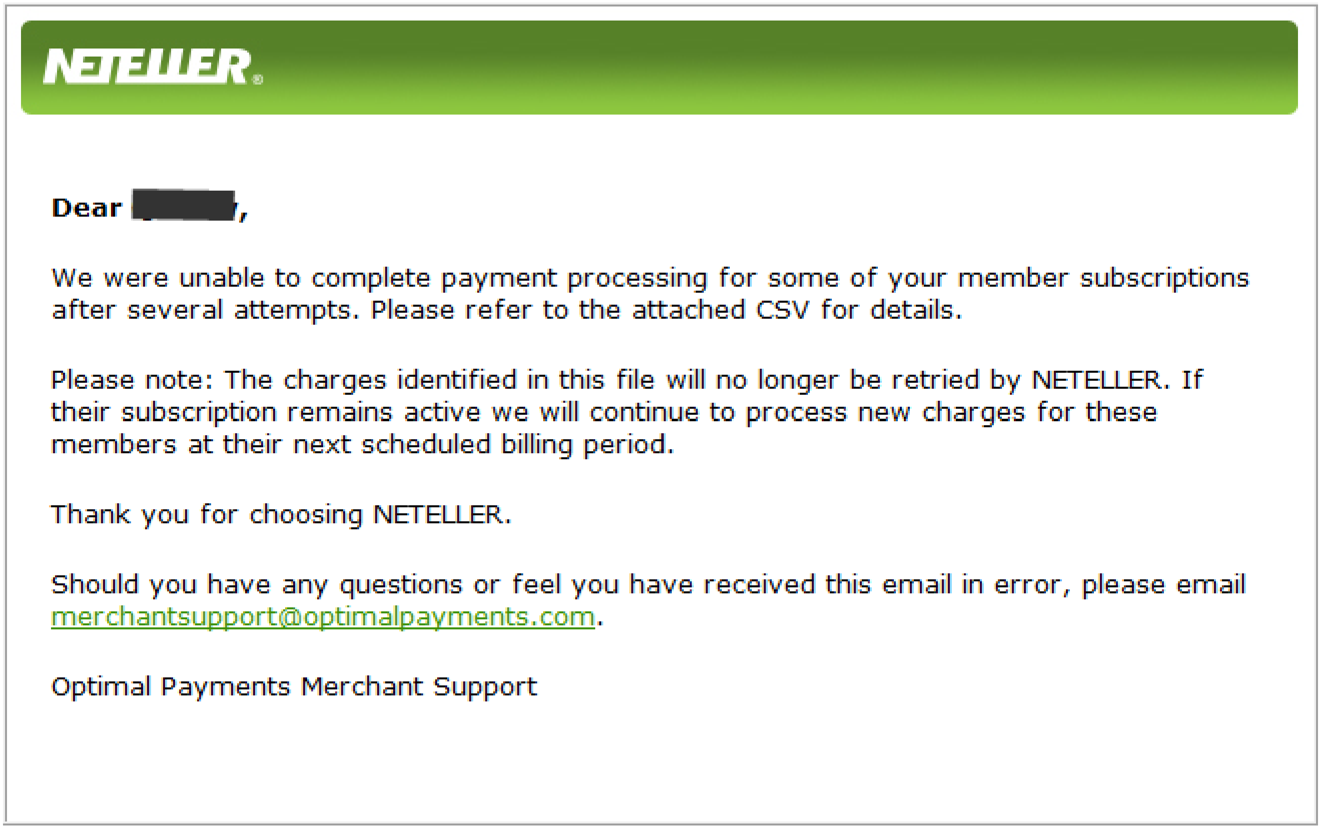
```
{
    "mode": "test",
    "id": "69f4022c-8405-439e-841e-5a7654820474",
    "eventDate": "2014-10-14T20:44:39Z",
    "eventType": "subscription_payment_succeeded",
    "attemptNumber": "1",
    "key": "MySecretK3y"
}
```

Billing Process

NETELLER will automatically attempt to collect payment for all your subscriptions on their due date. In the event the member has insufficient account balance to process the payment, NETELLER will re-try payment collection 3 times over a 7 day period. (1st, 3rd, 7th day after initial charge attempt). The member will be notified that they have insufficient funds in their account and that they need to top up their balance before the next retry.

Failed charge notification email

In the unlikely event that NETELLER is unable to collect payment for the current period billing, you will receive an email notification with an attached CSV file detailing any failed payments for the current day.



Dear [redacted],

We were unable to complete payment processing for some of your member subscriptions after several attempts. Please refer to the attached CSV for details.

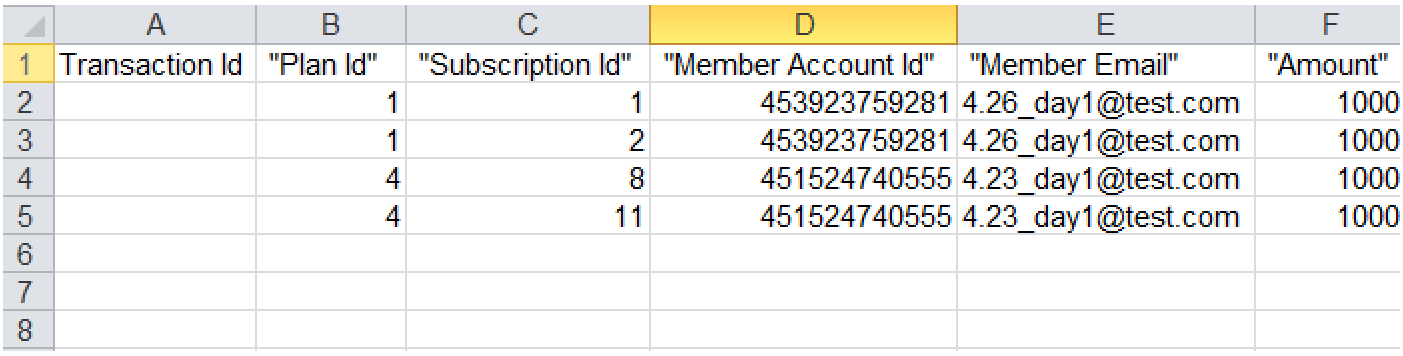
Please note: The charges identified in this file will no longer be retried by NETELLER. If their subscription remains active we will continue to process new charges for these members at their next scheduled billing period.

Thank you for choosing NETELLER.

Should you have any questions or feel you have received this email in error, please email merchantsupport@optimalpayments.com.

Optimal Payments Merchant Support

Sample CSV attachment:



	A	B	C	D	E	F
1	Transaction Id	"Plan Id"	"Subscription Id"	"Member Account Id"	"Member Email"	"Amount"
2		1	1	453923759281	4.26_day1@test.com	1000
3		1	2	453923759281	4.26_day1@test.com	1000
4		4	8	451524740555	4.23_day1@test.com	1000
5		4	11	451524740555	4.23_day1@test.com	1000
6						
7						
8						

Authentication

NETELLER REST APIs are based on the OAUTH2 standard as defined by RFC6749 (http://tools.ietf.org/html/rfc6749) and require that you supply a valid OAUTH2 access token with each API request.

The REST APIs utilize a 'bearer' token. For more information on bearer tokens see: http://tools.ietf.org/id/draft-ietf-oauth-v2-bearer-23.txt

Sub Sections:

- Obtain an access token
- Error Response
- OAUTH Scopes (Permissions)
- OAUTH2 Workflow Diagram

Token URIs

Obtain an access token using the client_credentials grant type
POST https://api.neteller.com/v1/oauth2/token?grant_type=client_credentials

Obtain an access token using your authorization code
POST https://api.neteller.com/v1/oauth2/token?grant_type=authorization_code&code=YOUR_AUTHORIZATION_CODE&redirect_url=YOUR_REDIRECT_URL

Obtain an access token using your refresh token
POST https://api.neteller.com/v1/oauth2/token?grant_type=refresh_token&refresh_token=YOUR_REFRESH_TOKEN

Authorization URIs

Obtain an authorization code
GET https://api.neteller.com/v1/oauth2/authorize?client_id=YOUR_APPLICATION_ID&redirect_uri=YOUR_URI&scope=YOUR_REQUESTED_SCOPE&response_type=code

Obtain an access token

To access the NETELLER REST API's, you will require an access token that can be obtained from the NETELLER OAUTH2 token endpoint. Obtain an access_token by making a request to the NETELLER Token endpoint with the appropriate grant_type identified.

> ⓘ To obtain access tokens you ***must*** first register your client (application) you wish to integrate with. APP registration can be completed from your merchant account portal and will provide you with your APP specific client_id and client_secret. By default, the access token is a short lived one-time use token that expires in 5 minutes. If your integration requires frequent callbacks and you would like to be able to re-use the access token, or require an extended token expiry time, please contact NETELLER Merchant Support to have your APP settings adjusted accordingly.

The NETELLER Platform supports the following OAuth 2.0 flows:

client_credentials grant

Allows you to request an access token using only the client credentials provided to you. This grant is used to access API functions that are under your control, and that do not require member authorization.

authorization_code grant

If your API request requires member permission (required scope permissions are identified in the section detailing each API operation), you will be required to use the authorization_code grant flow and will have an additional step of obtaining member authorization before you can obtain an access token. This flow requires you to redirect your users to the authorization server where they will authenticate as a NETELLER member and approve the requested permissions. The supplied auth_code can then be exchanged for an access_token allowing you access API functions on the members behalf. The client must be capable of interacting with the resource owner's user agent (typically a web browser) and capable of receiving incoming requests (via redirection) from the authorization server.

> ⓘ These flows are intended for clients capable of maintaining their client credentials in a confidential manner, such as a client implemented on a secure server (https)

Obtain an access token using the client_credentials grant type

You can request an access token using only your client credentials (client_id:client_secret).

ⓘ Your client credentials consist of a unique client_id and client_secret that can be found under the registered APPs section of your NETELLER Merchant Account.

The Authorization header should be constructed to pass a base64 encoded version of these values (client_id:client_secret). If you are testing using cURL or a browser client, you can specify the HTTP Basic credentials by using the **client_id** as your username and **client_secret** as your password.

POST https://api.neteller.com/v1/oauth2/token?grant_type=client_credentials

The client_secret for your application should not be shared with anyone or embedded in any code that you will distribute. This exclusion also extends to an app binary that could be de-compiled. This API call should only be made using server-side code over TLS.

Request Parameters

Parameter Name	Required?	Description
grant_type	**required**	set to '**client_credentials**'

Sample Request #1 (using HTTP Basic Authorization)

The authorization value is equal to Basic: Base64Encoded(client_id:client_secret)

```
curl -X POST \
-H "Authorization:Basic
QUFBQlJiUkh3V2tzbG8tdTowLmtnOG1wSnhOeER5ZUx2SVNVZ04wbHFsWWVWQS1UeFZSWlVFWThIdWljOEEuUzR
uRnpPWS0wLTF3bW1qd3lGV2l0aUZwRGxz" \
-H "Content-Type:application/json" \
-H "Cache-Control:no-cache" \
https://api.neteller.com/v1/oauth2/token?grant_type=client_credentials
```

Sample Request #2 (cURL specific sample, using client_id and client_secret as username/password)

```
curl -X POST \
-u
"AAABRbRHwWkslo-u:0.kg8mpJxNxDyeLvISUgN0lqlYeVA-TxVRZUEY8Huic8A.S4nFzOY-0-1wmmjwyFWitiF
pDls" \
-H "Content-Type:application/json" \
-H "Cache-Control:no-cache" \
https://api.neteller.com/v1/oauth2/token?grant_type=client_credentials
```

Response Structure

Field Name	Field Type	Constraints	Description

accessToken	string	length <= 200	The access token to be used in calls to the protect NETELLER REST APIs.
tokenType	string	length <= 50	Identifies the type of token to the client. Possible values: **Bearer**
expiresIn	number		The lifetime in seconds of the access token.

A refresh token WILL NOT be included for this grant type. It is only applicable for type: authorization_grant

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{

"accessToken":"0.AQAAAUW-9Tu4AAAAAAAEk-BNppPGFNwoWODBcOzrnHwA.skB0dDtyMrW4xCZJw__FGNtL-
08",
   "tokenType":"Bearer",
   "expiresIn":300
}
```

If the request failed client authentication or is invalid, the authorization server returns an error response.

```
{
  "error": "unsupported_grant_type"
}
```

Obtain an access token using the auth_code grant type

Is member authorization required?
If the API resource does not require member authorization (identified by a scope of '**default**'), you can Obtain an access token using the client_credentials grant type instead.

The authorization grant flow has two parts and utilizes two authorization server endpoints.

1) **Authorization Endpoint** - used to obtain authorization (auth_code) from the resource owner via user-agent redirection.

Your application redirects the member to NETELLER where they are asked to grant the requested permissions. if the member approves, NETELLER will post back to your redirect uri with a unique authorization code that represents this members acknowledgement of your request.

2) **Token Endpoint** - used to trade in the auth_code, or previously issued refresh_token for an new access token

The authorization code can then be exchanged for an access_token to use with your REST API calls. This process can be repeated until authorization is revoked by the member or until different permissions are required.

The flow consists of the following steps:

1. The client app requests authorization by redirecting the user to the ***authorization*** endpoint (NETELLER OAuth dialog).
2. The resource owner (member) must verify their identity via a hosted authentication page.
3. The resource owner (member) is prompted to authorize the application. The requested scope (permissions) are displayed and the member must Allow or Deny the authorization.
4. If the member authorized the requested scope, an auth_code is returned to your redirect uri. This code will have a limited lifetime for you to complete the callback to redeem the access_token.
5. Now you need to authorize your app. Call the ***token*** endpoint with a grant type of 'authorization_code' and provide the auth_code from (4). If successful, an access_token is returned that can be used to make API requests on the users behalf.

When invoking this flow, you must pass in the redirect_uri that the user's browser will be redirected back to once app authorization is completed. ***The redirect_uri must be the same URI you specify when registering your web application*** **OR** ***must be the same one that was passed in the original authorization call***.

Obtain an authorization code

```
GET https://api.neteller.com/v1/oauth2/authorize?client_id=YOUR_APPLICATION_ID&redirect_uri=YOUR_URI&scope=YOUR_REQUESTED_SCOPE&response_type=code
```

Authorization Request Parameters

Parameter Name	Required?	Description
client_id	required	Unique identifier generated when you registered your application.
redirect_uri	optional	The uri that the user's browser will be redirected back to once app authorization is completed. If supplied during this authorization request, the same uri must be supplied in the call to the token endpoint. When not supplied, the default registered redirect uri will be used. **The redirect_uri must secured over TLS (https://)**

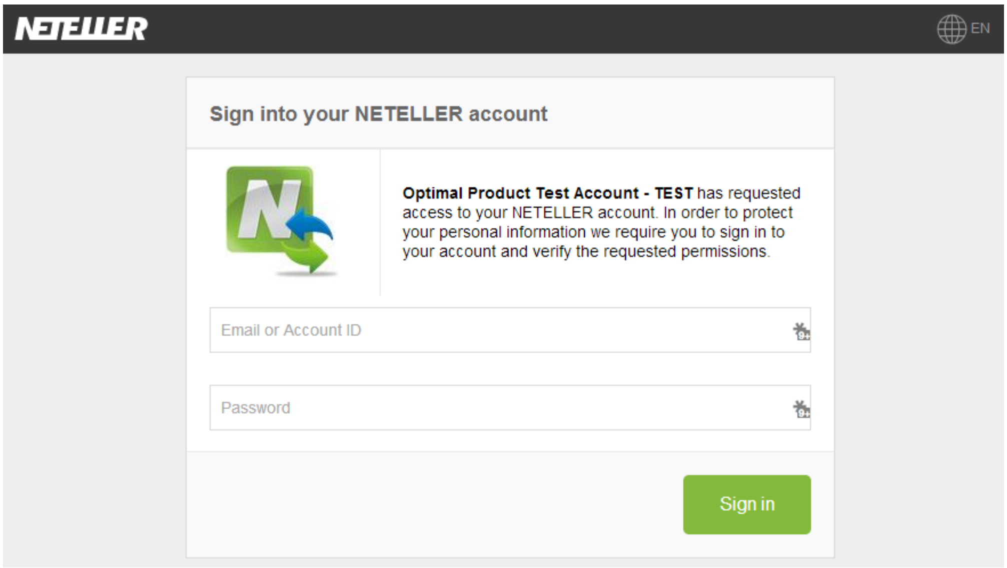
scope	required	The supplied scope parameter will indicate the level of access you require for your app. The scope parameter MUST be supplied. If you application requires multiple scopes then you should supply a space seperated list of the required permissions. Eg. scope=perm1+perm2
state	optional	
response_type		Identifies the type of response for this request. Valid values: **code** - for server side flows **Only code is permitted at this time.
lang		OPTIONAL: preferred language to render the authorization flow in. see Languages for complete list

The following request sample shows how to ask for authorization for enrolling a member in a subscription plan so you can receive automated recurring payments..

```
GET
https://api.neteller.com/v1/oauth2/authorize?client_id=AAABRbRHwWkslo-a&redirect_uri=ht
tps://www.mysite.com/cashier&scope=subscription_payment&response_type=code
```

The following flows will be presented:

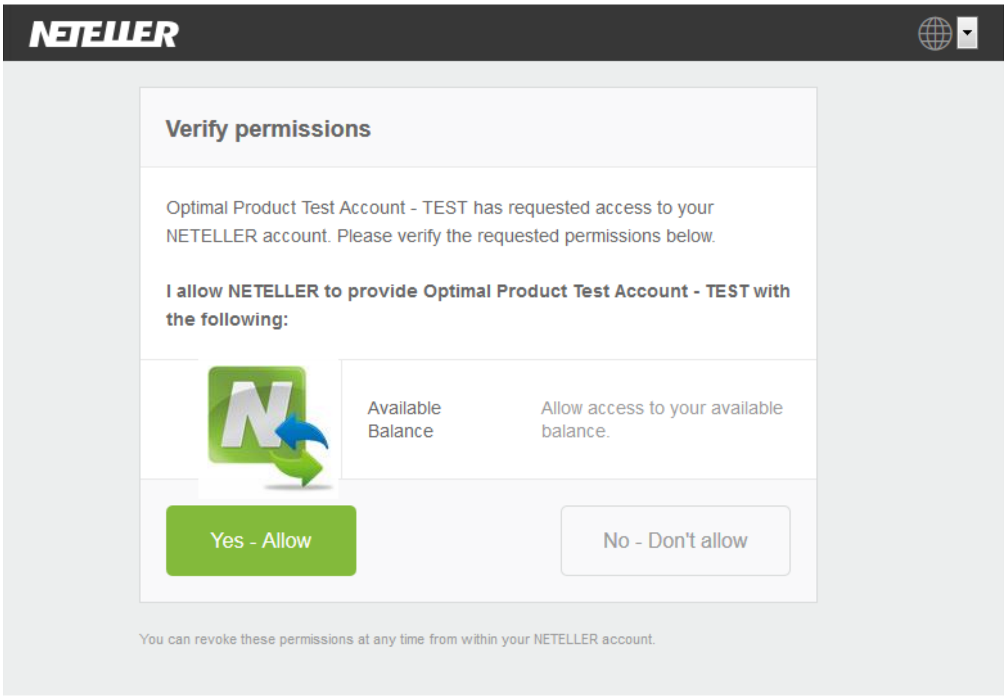
Member Authentication - Member will need to verify their identify using their standard credentials.



Sign into your NETELLER account

Optimal Product Test Account - TEST has requested access to your NETELLER account. In order to protect your personal information we require you to sign in to your account and verify the requested permissions.

Member will be prompted to authorize the requested permissions.



Verify permissions

Optimal Product Test Account - TEST has requested access to your NETELLER account. Please verify the requested permissions below.

I allow NETELLER to provide Optimal Product Test Account - TEST with the following:

	Available Balance	Allow access to your available balance.

You can revoke these permissions at any time from within your NETELLER account.

Authorization Response Structure

Authorization Success

If the member authorizes your request NETELLER will redirect (via HTTP 302) the user's browser by adding the following parameters to the query component of the redirection URI using the "application/x-www-form-urlencoded" format.

Parameter Name	Description
code	A short lived authorization code that can be exchanged for an access token. This value is short lived (10 minutes) and must be exchanged for a valid access token before it expires.
state	Will be provided if the **state** parameter was present in the authorization request.

Sample Response

```
https://https://www.mysite.com/cashier?auth_code=0.AAAAAUmlhiRUAAAAAAAEk-DKmil1fF5vleYL
DaiRhPDi.si-fsMDooIjhdGrShvafUALhQP8&state=abc
```

Exchange your authorization code for an access token

Once you have received member authorization, you can exchange your authorization_code for an access token.

POST https://api.neteller.com/v1/oauth2/token?grant_type=authorization_code&code=YOUR_AUTHORIZATION_CODE&redirect_uri=YOUR_REDIRECT_URI

Token Request Parameters

Request Parameters

Field Name	Field Type	Field Size	Required?	Description
grant_type	string	100	**required**	set to '**authorization_code**'
code	string	20	**required**	set to the auth_code previously received from the authorization server.
redirect_uri	string	500	optional	The TLS enabled redirection url that will be returned to. This value must match the URL used to obtain authorization, If not supplied during the authorization flow then the redirect_uri will be the registered redirect url for your application and should not be passed.

Sample Request #1 (using client_id and client_secret as username/password)

```
curl -X POST \
-u
"AAABRbRHwWkslo-u:0.kg8mpJxNxDyeLvISUgN0lqlYeVA-TxVRZUEY8Huic8A.S4nFzOY-0-1wmmjwyFWitiF
pDls" \
-H "Content-Type:application/json" \
-H "Cache-Control:no-cache" \
"https://api.neteller.com/v1/oauth2/token?grant_type=authorization_code&code=0.AAAAAUfv
VDS3AAAAAAAEk-B5Ei4gPXAQA4B8EDH4jpdC._Cm7iYsUq1SQgJNiZg11yQwfpEg&redirect_uri=https://w
ww.mysite.com"
```

Sample Request (using HTTP Basic Authorization)

The authorization value is equal to Basic: Base64Encoded(client_id:client_secret)

```
curl -X POST \
-H "Authorization:Basic
QUFBQlJiUkh3V2tzbG8tdTowLmtnOG1wSnhOeER5ZUx2SVNVZ04wbHFsWWVWQS1UeFZSWlVFWThIdWljOEEuUzR
uRnpPWS0wLTF3bW1qd3lGV2l0aUZwRGxz" \
-H "Content-Type:application/json" \
-H "Cache-Control:no-cache" \
"https://api.neteller.com/v1/oauth2/token?grant_type=authorization_code&code=0.AAAAAUfv
VDS3AAAAAAAEk-B5Ei4gPXAQA4B8EDH4jpdC._Cm7iYsUq1SQgJNiZg11yQwfpEg&redirect_uri=https://w
ww.mysite.com"
```

If your app is successfully authenticated, and the authorization code is valid, the authorization server will return the following:

Token Request Success

Response Structure

A **token** object will be returned.

Field Name	Field Type	Field Size	Description
accessToken	string	100	The access token to be used in calls to the protect NETELLER REST APIs.

tokenType	string	20	Identifies the type of token to the client. For now only 'Bearer' is supported.
expiresIn	number	6	The lifetime in seconds of the access token. Default is 300 seconds (5 minutes).
refreshToken	string	100	The refresh token which can be used to obtain new access tokens using the same authorization grant that was previously associated to it.
scope	string	250	The scope associated with the access token.

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
  "accessToken":
"0.AQAAAUgUGIk5AAAAAAAEk-CgJxQeNBiq7ES8yQaLITPU.JGrVsVsiy3ASoQsYED9VkJv25eY",
  "tokenType": "Bearer",
  "expiresIn": 300,
  "refreshToken":
"0.AgAAAUgUGIkyAAAAB1jwsODI5PCkVNOZul5AJ01mYtdh.ezGCYhN5YD22-BCOPX6U-muc72o",
  "scope": "subscription_payment"
}
```

Obtain an access token using your refresh token

If the previously issued access token has expired (obtained from the authorization grant authenticaiton flow) and you receive a **HTTP 401 - Unauthorized error**, you can utilize the refresh token that was issued to you with the initial authorization response to obtain a new one.

POST https://api.neteller.com/v1/oauth2/token?grant_type=refresh_token&refresh_token=YOUR_REFRESH_TOKEN

Request Parameters

Parameter Name	Required?	Description
grant_type	**required**	set to '**refresh_token**'
refresh_token	**required**	The refresh token previously issued to the client

Sample Request

```
curl -X POST \
-H "Authorization:Basic
QUFBQlJiUkh3V2tzbG8tdTowLmtnOG1wSnhOeER5ZUx2SVNVZ04wbHFsWWVWQS1UeFZSWlVFWThIdWljOEEuUzR
uRnpPWS0wLTF3bW1qd3lGV2l0aUZwRGxz" \
-H "Content-Type:application/json" \
-H "Cache-Control:no-cache" \
"https://api.neteller.com/v1/oauth2/token?grant_type=refresh_token&refresh_token=0.AgAA
AUnPnWEeAAAAB1jwsOAq6iK0G2op1zi_sQC3KE22.oD4wHNZjuF5tWOZh2ozEpi2E7BE"
```

Response Structure

Field Name	Field Type	Field Size	Description
accessToken	string	length <= 200	The access token to be used in calls to the NETELLER REST APIs.
expiresIn	number		The lifetime in seconds of the access token.
tokenType	string	20	Identifies the type of token to the client. "Bearer"
scope	string		The comma delimited list of scopes that have been authorized for this token.

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
   "accessToken":
"0.AQAAAUfvtyJyAAAAAAA27oALPWxe111yJMNH3HEa6l_o.EddShan6OS7sQX28UhYz5cz_LHE",
   "expiresIn": 300,
   "tokenType": "Bearer",
   "scope": "subscription_payment, account_available_balance"
}
```

Error Response

Authorization Request Errors

If the member does not authorize your request, or the request fails for reasons other then a missing or invalid redirection URI NE TELLER will redirect (via HTTP 400 - Bad Request) the user's browser by adding the following parameters to the query component of the redirection URL using the "application/x-www-form-urlencoded" format.

Response Parameters

Parameter Name	Description
error	The error that occurred.
error_description	Reason for the error.
state	Will be provided if the **state** parameter was present in the authorization request.

Possible Errors

Error	Error Reason
unauthorized_client	The client is not authorized to request an authorization code using this method.
access_denied	The resource owner or authorization server denied your request.
unsupported_response_type	The authorization server does not support obtaining an authorization code using this method.
invalid_scope	The requested scope is invalid, unknown, or malformed.
server_error	The authorization server encountered an unexpected condition that prevented it from fulfilling the request.(This error code is needed because a 500 Internal Server Error HTTP status code cannot be returned to the client via an HTTP redirect.)
temporarily_unavailable	The authorization server is currently unable to handle the request due to a temporary overloading or maintenance of the server. (This error code is needed because a 503. Service Unavailable HTTP status code cannot be returned to the client via an HTTP redirect.)

Sample Response

```
https://YOUR_URL/?error=access_denied&error_description=The+resource+owner+or+authoriza
tion+server+denied+your+request
```

Token Request Errors

The authorization server responds with an HTTP 400 (Bad Request) status code (unless specified otherwise) and includes the following parameters with a JSON response:

Response Structure

Field Name	Description

error	The error that occurred.
error_description	Reason for the error.
state	Will be provided if the **state** parameter was present in the authorization request.

ⓘ **Possible Errors**

Error	Error Reason
invalid_request	The request is missing a required parameter, includes an unsupported parameter value (other than grant type), repeats a parameter, includes multiple credentials, utilizes more than one mechanism for authenticating the client, or is otherwise malformed.
invalid_client	Client authentication failed (e.g., unknown client, no client authentication included, or unsupported authentication method). The authorization server MAY return an HTTP 401 (Unauthorized) status code to indicate which HTTP authentication schemes are supported. If the client attempted to authenticate via the "Authorization" request header field, the authorization server MUST respond with an HTTP 401 (Unauthorized) status code and include the "WWW-Authenticate" response header field matching the authentication scheme used by the client. **This error might also indicate that you are attempting access from an unauthorized IP address.
invalid_grant	The provided authorization grant (e.g., authorization code, resource owner credentials) or refresh token is invalid, expired, revoked, does not match the redirection URI used in the authorization request, or was issued to another client.
unauthorized_client	The authenticated client is not authorized to use this authorization grant type.
server_error	The authorization server encountered an unexpected condition that prevented it from fulfilling the request.(This error code is needed because a 500 Internal Server Error HTTP status code cannot be returned to the client via an HTTP redirect.)
unsupported_grant_type	The authorization grant type is not supported by the authorization server.
invalid_scope	The requested scope is invalid, unknown, malformed, or exceeds the scope granted by the resource owner.

Sample Response

```
{
  "error": "invalid_grant"
}
```

OAUTH Scopes (Permissions)

Some resources available through the NETELLER REST API require that your APP be granted appropriate authorization by the member. This authorization of permissions, also referred to as authorized scope, is managed by directing the NETELLER member to a page where they can choose to approve or deny the requested access.

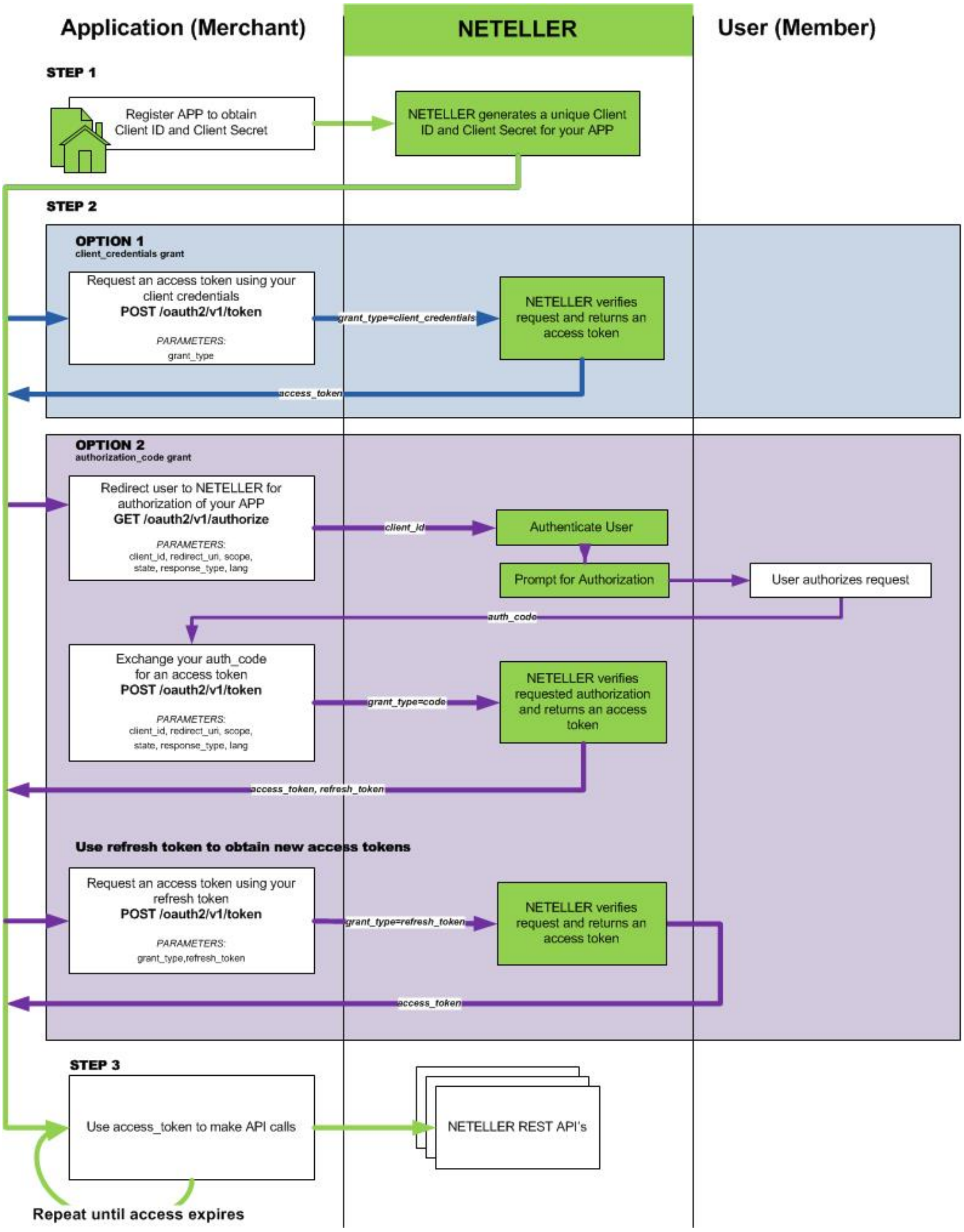
The following list identifies the authorized scopes that are available in the NETELLER API. These permissions, or scopes, are handled on a per member / per access token basis. To retrieve a resources protected fields, you must first obtain an access token using the authorization grant flow or previously issued refresh token. You can continue to obtain new access tokens for the already authorized scopes until the authorization expires or is revoked by the member.
For API requests that permit **default** scope, you may not require member authorization if the fields you are concerned with are offered as part of the default response.   In this instance you should use your client credentials to obtain the required access token as there is no required member authorization.

⚠ The permissions associated with an access token are set in the ***/v1/oauth2/authorize*** call.

Some API methods will limit their response data dependent upon which authorized scopes have been granted. See the JSON Objects reference for details around which fields are returned by default and which ones require member authorization.

Permission	Authorized Scope	Description
Subscription Payment	subscription_payment	Permits access to setup a subscription and automatically receive recurring payments from a NETELLER account holder ***Member Prompt: Allow access to setup a subscription and receive recurring payments from your NETELLER account.***
Enhanced Profile	account_enhanced_profile	Permits access to account address, city, post code, country subdivision, account preferences and gender ***Member Prompt: Allow access to your address, account preferences and gender.***
Available Balance	account_available_balance	Permits access to view the members currently available balance ***Member Prompt: Allow access to view your currently available balance.***
Contact Information	account_contacts	Permits access to the member contact phone number(s). ***Member Prompt: Allow access to your contact phone number(s).***

OAUTH2 Workflow Diagram


Application (Merchant)
NETELLER
User (Member)
STEP 1
Register APP to obtain Client ID and Client Secret
NETELLER generates a unique Client ID and Client Secret for your APP
STEP 2
OPTION 1
client_credentials grant
Request an access token using your client credentials
POST /oauth2/v1/token
PARAMETERS:
grant_type
grant_type=client_credentials
NETELLER verifies request and returns an access token
access_token
OPTION 2
authorization_code grant
Redirect user to NETELLER for authorization of your APP
GET /oauth2/v1/authorize
PARAMETERS:
client_id, redirect_uri, scope, state, response_type, lang
client_id
Authenticate User
Prompt for Authorization
User authorizes request
auth_code
Exchange your auth_code for an access token
POST /oauth2/v1/token
PARAMETERS:
client_id, redirect_uri, scope, state, response_type, lang
grant_type=code
NETELLER verifies requested authorization and returns an access token
access_token, refresh_token
Use refresh token to obtain new access tokens
Request an access token using your refresh token
POST /oauth2/v1/token
PARAMETERS:
grant_type,refresh_token
grant_type=refresh_token
NETELLER verifies request and returns an access token
access_token
STEP 3
Use access_token to make API calls
NETELLER REST API's
Repeat until access expires

Customers

The customer resource contains the profile information for a NETELLER account holder. You can use the customer object to supply your customers profile information to speed creation of their NETELLER account, or use it to inquire about existing member details.

Sub Sections:

- Create a customer (Quick Signup)
- Lookup customer

> ⚠ A customer reference is not necessarily considered a per-requisite to initiate other REST API calls that involve that customer. For example, all that is required to initiate a payment is a recipient email address.

Resource URIs

Create a customer (Quick Signup)
POST https://api.neteller.com/v1/customers

Lookup customer
GET https://api.neteller.com/v1/customers/CUSTOMER_ID **GET** https://api.neteller.com/v1/customers/?email=CUSTOMER_EMAIL **GET** https://api.neteller.com/v1/customers/?accountId=ACCOUNT_ID

Create a customer (Quick Signup)

This API call will return you a link that you can redirect your customers to.so then can quickly complete their NETELLER account signup. The signup pages will be pre-populated with profile information you have already collected.

> ⚠ The user must complete the signup process by following the supplied link in order to complete setup of their NETELLER account.

POST https://api.neteller.com/v1/customers

Scope required to initiate request: none (default)

Scope(s) impacting response: none (default)

Expandable Resources: none

Related Resources (if applicable): none

Request Structure

Field Name	Field Type	Constraints	Required?	Description
accountProfile	accountProfile		required	The customer profile information to be used to pre-fill the signup pages with.
linkBackUrl	linkBackUrl		optional	A valid http:// or https:// url that should be presented to the member upon completion of their signup/deposit that will direct them back to your site. ⚠ If the URL cannot be validated it will not be displayed after completion of the account signup.
btag	string	length <= 100	optional	This value should be provided if you are registered in the NETELLER affiliate program. The btag can be found as a parameter in the banner URL string. A_<siteid>B_<creativeid>C_

Sample Request

```
curl -X POST https://api.neteller.com/v1/customers \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
{
    "accountProfile": {
        "firstName": "John",
        "lastName": "Smith",
        "email": "jsmith@email.com",
        "address1": "addressline1",
        "address2": "addressline2",
        "address3": "addressline3",
        "city": "Calgary",
  "country": "CA",
   "countrySubdivisionCode": "AB",
        "postCode": "T8A22J",
        "contactDetails": [
            {
                "type": "landLine",
                "value": "14032332333"
            },
            {
                "type": "mobile",
                "value": "14035552333"
            }
        ],
        "gender": "m",
        "dateOfBirth": {
            "year": "1975",
            "month": "01",
            "day": "31"
        },
        "accountPreferences": {
            "lang": "en",
            "currency": "EUR"
        }
    },
    "linkBackUrl": "https://mymerchant.com/cashier?TOKEN=234AJ242K22KMJKP",
    "btag": "A_234B_345C_"
}
```

Response Structure

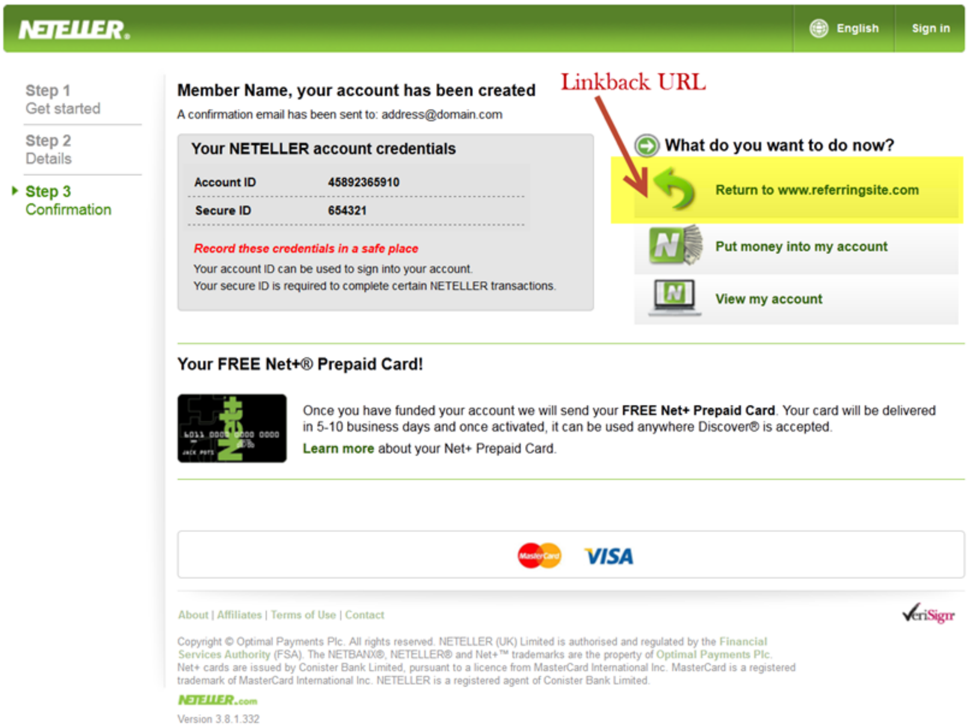
Returns related HATEOAS links.

Sample Response

```
{
    "links": [
        {
            "url":
"https://member.neteller.com/member?emailAddress=jsmith@email.com&token=f45bc7114859cda
1f46d8a2bbbc3449f5f42f",
            "rel": "member_signup_redirect",
            "method": "GET"
        }
    ]
}
```

If you supplied a valid linkbackurl, the member will have the option of returning back to your site immediately upon completion of the account signup flow.

Sample account confirmation screen


NETELLER.
English
Sign in
Linkback URL
Step 1
Get started
Step 2
Details
Step 3
Confirmation
What do you want to do now?
Return to www.referringsite.com
Put money into my account
View my account
About | Affiliates | Terms of Use | Contact
VeriSign
NETELLER.com

Member Name, your account has been created

A confirmation email has been sent to: address@domain.com

Your NETELLER account credentials

Account ID	45892365910
Secure ID	654321

Record these credentials in a safe place

Your account ID can be used to sign into your account.
Your secure ID is required to complete certain NETELLER transactions.

Your FREE Net+® Prepaid Card!

Once you have funded your account we will send your **FREE Net+ Prepaid Card**. Your card will be delivered in 5-10 business days and once activated, it can be used anywhere Discover® is accepted.

Learn more about your Net+ Prepaid Card.

Copyright © Optimal Payments Plc. All rights reserved. NETELLER (UK) Limited is authorised and regulated by the Financial Services Authority (FSA). The NETBANX®, NETELLER® and Net+™ trademarks are the property of Optimal Payments Plc. Net+ cards are issued by Conister Bank Limited, pursuant to a licence from MasterCard International Inc. MasterCard is a registered trademark of MasterCard International Inc. NETELLER is a registered agent of Conister Bank Limited.

Version 3.8.1.332

Lookup customer

Lookup customer details for a specific email or NETELLER account id.

> ⚠ Lookup of a customer will only return a result if the customer has completed their account setup.

GET https://api.neteller.com/v1/customers/CUSTOMER_ID

GET https://api.neteller.com/v1/customers/?email=CUSTOMER_EMAIL

GET https://api.neteller.com/v1/customers/?accountId=ACCOUNT_ID

Scope required to initiate request: none (default)

Scope(s) impacting response: account_enhanced_profile, account_contacts, account_available_balance

Expandable Resources: none

Related Resources (if applicable): none

Request Parameters

Parameter Name	Required?	Description
email	optional	The customers NETELLER registered email
accountId	optional	The customers NETELLER account id.

Sample Request

Using customer id

```
curl -X GET
https://api.neteller.com/v1/customers/CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325 \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Using email address:

```
curl -X GET https://api.neteller.com/v1/customers/?email=john.doe@email.com \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Using account ID:

```
curl -X GET https://api.neteller.com/v1/customers/?accountId=451234567890 \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Response Structure

Returns an expanded **customer** object with additional system generated information.

Field Name	Field Type	Field Size	Response Field Scope	Description
customerId	string	length <= 100	default	Unique identifier for this customer. ie: CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325
accountProfile	accountProfile	length <= 100	default	The member account profile.
verificationLevel	string	length = 1	default	00 - Member has not been verified, no registered payment instruments 01 - Member has not been verified, has one or more registered payment instruments 10 - Member is verified, no registered payment instruments 11 - Member is verified, has one or more registered payment instruments
availableBalance	balance		account_available_balance	Member currently available account balance.

Sample Response (default response)

```
{
   "customerId": "CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
   "accountProfile": {
        "email": "jsmith@email.com",
        "accountId": "451234567890",
        "firstName": "John",
        "lastName": "Smith",
        "country": "CA",
        "dateOfBirth": {
            "year": "1975",
            "month": "01",
            "day": "31"
        }
    },
    "verificationLevel": "01"
}
```

Sample Response (with authorized scope: account_available_balance, account_enhanced_profile)

```
{
    "customerId": "CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
    "accountProfile": {
        "email": "jsmith@email.com",
        "accountId": "451234567890",
        "firstName": "John",
        "lastName": "Smith",
        "address1": "addressline1",
        "address2": "addressline2",
        "address3": "addressline3",
        "city": "Calgary",
        "countrySubdivisionCode": "AB",
        "country": "CA",
        "postCode": "T8A22J",
        "contactDetails": [
            {
                "type": "landLine",
                "value": "14032332333"
            },
            {
                "type": "mobile",
                "value": "14035552333"
            }
        ],
        "gender": "m",
        "dateOfBirth": {
            "year": "1975",
            "month": "01",
            "day": "31"
        },
        "accountPreferences": {
            "lang": "en",
            "currency": "EUR"
        }
    },
    "verificationLevel": "01",
    "availableBalance": {
        "amount": 2341,
        "currency": "EUR"
    }
}
```

Payments

The payments resource is available to facilitate the transfer of payments between accounts within the NETELLER system. A payment to a non-registered NETELLER member can optionally include profile information (including email, name, address, phone, birthdate) that will be used to enhance the signup experience for new members, so they do not have to re-key in information that is already available in your system.

Sub Sections:

- Create a payment (outgoing transfer)
- Request a payment (incoming transfer)
- Lookup payment

Resource URIs

Create a payment (outgoing transfer)
POST https://api.neteller.com/v1/transferOut

Request a payment (incoming transfer)
POST https://api.neteller.com/v1/transferIn

Lookup payment
GET https://api.neteller.com/v1/payments/TRANSACTION_ID **GET** https://api.neteller.com/v1/payments/YOUR_MERCHANT_REF_ID?refType=merchantRefId

Create a payment (outgoing transfer)

Create a new outgoing payment. NETELLER will notify the recipient via email of the payment. If the recipient is not an existing NETELLER account holder, they will be required to complete registration in order to have access to the funds. A reference URL will be returned in the payment request response so you can optionally direct the member to complete the registration process from your own flows without them having to leave your site to check email. If the recipient is already a NETELLER member, the funds will be immediately deposited in their account.

POST https://api.neteller.com/v1/transferOut

Scope required to initiate request: none (default)

Scope(s) impacting response: account_enhanced_profile, account_contact

Expandable Resources: customer

Related Resources (if applicable): none

Request Structure

Field Name	Field Type	Constraints	Required?	Description
payeeProfile	accountProfile		required	The recipient of the payment ⚠ At minimum, the payee's email address or NETELLER account ID must be supplied.
transaction	transaction		required	Details about the payment
message	string	length <= 255	*optional*	The message what will be shown to the member notifying them of the pending transfer.

Sample Request

```
curl -X POST https://api.neteller.com/v1/transferOut \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
     -d '
{
    "payeeProfile": {
        "email": "john.doe@email.com"
    },
    "transaction": {
        "amount": 2500,
        "currency": "EUR",
        "merchantRefId": "20140203113703"
    },
    "message": "sample message"
}'
```

Sample Request w/Optional Recipient Profile

```
curl -X POST https://api.neteller.com/v1/transferOut \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
     -d '
{
    "payeeProfile": {
        "email": "jsmith@email.com",
        "firstName": "John",
        "lastName": "Smith",
        "address1": "address line 1",
        "address2": "address line 2",
        "address3": "address line 3",
        "city": "Calgary",
        "countrySubdivisionCode": "AB",
        "postCode": "T8A22J",
        "contactDetails": [
            {
                "type": "landLine",
                "value": "14032332333"
            },
            {
                "type": "mobile",
                "value": "14035552333"
            }
        ],
        "gender": "m",
        "dateOfBirth": {
            "year": "1975",
            "month": "12",
            "day": "10"
        },
        "accountPreferences": {
            "lang": "en",
            "currency": "EUR"
        }
    },
    "transaction": {
        "amount": 2500,
        "currency": "EUR",
        "merchantRefId": "20140203113704"
    },
    "message": "sample message"
}'
```

Response Structure

Returns expanded payment object with related HATEOAS links.

Field Name	Field Type	Constraints	Description
customer	customer		The associated customer information for the payment
transaction	transaction		Details about the payment
links	array of link		Array of HATEOAS links (if applicable)

Sample Response (existing NETELLER account holder)

```
{
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "transaction": {
        "merchantRefId": "20140203113703",
        "amount": 2500,
        "currency": "EUR",
        "id": "176391453448397",
        "createDate": "2014-02-03T18:50:48Z",
        "updateDate": "2014-02-03T18:50:51Z",
        "status": "Accepted",
        "fees": [
            {
                "feeType": "Service_fee",
                "feeAmount": 71,
                "feeCurrency": "EUR"
            }
        ]
    },
    "links": [
        {
            "url": "https://api.neteller.com/v1/payments/176391453448397",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

Sample Response (non-existing NETELLER account holder)

```
{
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "transaction": {
        "merchantRefId": "20140203113704",
        "amount": 2500,
        "currency": "EUR",
        "id": "129391454779189",
        "createDate": "2014-02-03T19:12:59Z",
        "updateDate": "2014-02-03T19:12:59Z",
        "status": "pending",
        "fees": [
            {
                "feeType": "service_fee",
                "feeAmount": 71,
                "feeCurrency": "EUR"
            }
        ]
    },
    "links": [
        {
            "url": "https://api.neteller.com/v1/payments/129391454779189",
            "rel": "self",
            "method": "GET"
        },
        {
            "url":
"https://member.neteller.com/member?emailAddress=jsmith@email.com&token=f45bc7114859cda
1f46d8a2bbbc3449f5f42f",
            "rel": "member_signup_redirect",
            "method": "GET"
        }
    ]
}
```

Request a payment (incoming transfer)

Request a new incoming transfer payment. This function is similar to the legacy NetDirect API. The member will be required to enter the amount they wish to transfer and supply a verification code (their Secure Id) which will be passed back in your request. If approved, funds will immediately be pulled from the member account and deposited in your merchant account.

NETELLER will notify the sender via email of the completed payment.

POST https://api.neteller.com/v1/transferIn

Scope required to initiate request: none (default)

Scope(s) impacting response: account_enhanced_profile, account_contact

Expandable Resources: customer

Related Resources (if applicable): none

Request Structure

Field Name	Field Type	Constraints	Required?	Description
paymentMethod	paymentMethod		required	The source of funds for the transfer into your merchant account.
transaction	transaction		required	
verificationCode	string	length <=25	required	The member verification code for the transaction. This is the members 6 digit Secure Id, or Google Authenticator OTP if the member has two factor authentication enabled.

Sample Request

```
curl -X POST https://api.neteller.com/v1/transferIn \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
 -d '
{
    "paymentMethod": {
        "type": "neteller",
        "value": "gb_gbp@neteller.com"
    },
    "transaction": {
        "merchantRefId": "20140203122501",
        "amount": 5000,
        "currency": "USD"
    },
    "verificationCode": "234124"
}'
```

Response Structure

Returns expanded payment object with related HATEOAS links.

Field Name	Field Type	Constraints	Description
customer	customer		The associated customer information for the payment
transaction	transaction		Details about the payment

links	array of link		Array of HATEOAS links (if applicable)

Sample Response (existing NETELLER account holder)

```
{
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "transaction": {
        "merchantRefId": "20140203122501",
        "amount": 5000,
        "currency": "USD",
        "id": "672391456803770",
        "createDate": "2014-02-03T19:46:44Z",
        "updateDate": "2014-02-03T19:46:44Z",
        "status": "accepted",
        "fees": [
            {
                "feeType": "service_fee",
                "feeAmount": 195,
                "feeCurrency": "USD"
            }
        ]
    },
    "links": [
        {
            "url": "https://api.neteller.com/v1/payments/672391456803770",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

Lookup payment

Return details about a previous payment request. Additional detail will be provided depending on the transaction type and the current state of the request.

GET https://api.neteller.com/v1/payments/TRANSACTION_ID

GET https://api.neteller.com/v1/payments/YOUR_MERCHANT_REF_ID?refType=merchantRefId

Scope required to initiate request: none (default)

Scope(s) impacting response: account_enhanced_profile, account_contact

Expandable Resources: customer

Related Resources (if applicable): none

⚠ When performing a look up using MERCHANT_REF_ID, you must also pass the parameter of **?refType=merchantRefId** to denote that your are querying with your merchant reference and not the NETELLER transaction ID

This API request returns a payment object. The response object values vary dependent upon the type of payment that is being queried..

Request Parameters

Parameter Name	Required?	Description
refType	optional	An optional type of 'merchantRefId' can be specified to query using your own reference as opposed to the NETELLER system generated Id.

Sample Request

Using NETELLER transaction ID:

```
curl -X GET https://api.neteller.com/v1/payments/154391457301406 \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Using MerchantRefId

```
curl -X GET https://api.neteller.com/v1/payments/20140203122502?refType=merchantRefId \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Response Structure

Returns expanded payment object with related HATEOAS links.

Field Name	Field Type	Constraints	Description
customer	customer		The associated customer information for the payment
transaction	transaction		Details about the payment

links	array of link		Array of HATEOAS links (if applicable)

Sample Response

```
{
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "transaction": {
        "merchantRefId": "20140203122501",
        "amount": 5000,
        "currency": "USD",
        "id": "672391456803770",
        "createDate": "2014-02-03T19:46:44Z",
        "updateDate": "2014-02-03T19:46:44Z",
        "status": "accepted",
        "fees": [
            {
                "feeType": "service_fee",
                "feeAmount": 195,
                "feeCurrency": "USD"
            }
        ]
    },
    "links": [
        {
            "url": "https://api.neteller.com/v1/payments/672391456803770",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

Orders

The orders resource offers a hosted payment solution, also known as NETELLERGO!, so you can begin collecting payments from your customers with minimal custom development.   Simply utilize the appropriate API request to create an order, then re-direct your customer to our flexible checkout page where they are offered the ability to pay using their NETELLER account or with one of numerous alternative payment options.

Currently supported options include: Ukash, Sofort Banking, Giropay, Poli

> Contact the NETELLER Merchant Support team to have additional payment methods enabled for your merchant account.

Sub Sections:

- Create a order
- Lookup order
- Lookup invoice for a order

Resource URIs

Create a order
POST https://api.neteller.com/v1/orders

Lookup order
GET https://api.neteller.com/v1/orders/ORDER_ID

Lookup invoice for a order
GET https://api.neteller.com/v1/orders/ORDER_ID/invoice

Create a order

Create a order and redirect your customer to the supplied url to initiate the hosted Quick Checkout flow and collect payment from your customers.

POST https://api.neteller.com/v1/orders

Scope required to initiate request: none (default)

Scope(s) impacting response: none

Expandable Resources: none

Related Resources (if applicable): hosted_payment

Request Structure

Field Name	Field Type	Constraints	Required?	Description
order	order		required	The details for this order.
payerProfile	accountProfile		optional	Provide a profile object for your customer. At a minimum you should supply the customer email and country of residence (if known). If you do not know the country of residence of the customer, then they will be prompted to select it during their checkout.
attributes	array of attribute		optional	This allows you to specify additional data (e.g., affiliate codes, session IDs, etc.) to be stored against the transaction and returned to you in status calls, reports, etc.

Sample Request

```
curl -X POST https://api.neteller.com/v1/orders \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
     -d '
{
    "order": {
        "merchantRefId": "123456666",
        "totalAmount": 3599,
        "currency": "EUR",
        "lang": "en_US",
        "items": [
            {
                "quantity": 1,
                "name": "Item A",
                "description": "Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Nunc eget venenatis diam. Integer euismod magna dui, a accumsan elit interdum.",
                "sku": "XYZPART1",
                "amount": 2500
            },
            {
                "quantity": 2,
                "name": "Item B",
                "description": "Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Sed sit amet congue orci, ac euismod.",
                "amount": 200
            }
        ],
        "fees": [
            {
                "feeName": "Setup Fee",
                "feeAmount": 500
            }
        ],
        "taxes": [
            {
                "taxName": "VAT",
                "taxAmount": 199
            }
        ],
        "customerIp": "192.168.0.1",
        "paymentMethods": [
            {
                "type": "voucher",
                "value": "ukash"
            }
        ],
        "redirects": [
            {
                "rel": "on_success",
                "returnKeys": [
                    "id"
                ],
                "uri": "https://example.com/success.html"
            },
            {
                "rel": "on_cancel",
                "returnKeys": [
                    "id"
                ],
                "uri": "https://example.com/cancel.html"
            }
        ]
    },
    "payerProfile": {
        "email": "john.doe@email.com"
    },
    "attributes": [
        {
            "key": "session",
            "value": "53JK2KL2NZJK124"
        }
    ]
}
```

Response Structure

Returns an expanded **order** object, and relevant HATEOAS **links**.

Field Name	Field Type	Constraints	Description
orderId	string	length <= 100	Unique NETELLER reference for the order. (ie: ORD_7915d463-ccc8-4305-9d33-9e5c9310f12e, SUB_c46645ea-b7b0-4a91-87f7-88c00ec0dea1)
merchantRefId	string	length <= 50	The associated merchant reference id from the initial request.
totalAmount	number		
currency	string	length <= 3	The currency of the total amount due. The order must have the same currency for all items, fees and taxes.
lang	string	length <= 5	The language to render the hosted payments flows in.
items	array of item		Specific line items (products/fees or services) that you are requesting payment for.
fees	array of fee		Specific fees (setup/shipping&handling, etc...) that are to be included in the payment.
taxes	array of tax		Specific taxes(VAT, GST, PST, etc...) that are to be included in the payment.
customerIp	string	length <= 20	The value should be the incoming IP address of your customer, which will restrict the payment page to being viewable only from this IP address. ✓ Though the *customerIp* parameter is not mandatory, NETELLER recommends including it for additional security.
paymentMethods	array of paymentMethod		This allows you to define which payment options you would like to use.
redirects	array of redirect		This allows you to add redirects to the order. Redirects cause a connection back to your merchant system via the customer's browser

Sample Response (existing NETELLER account holder)

```
{
    "orderId": "ORD_959b1148-704d-4d92-9a7d-14a18990a648",
    "merchantRefId": "20141022142300",
    "totalAmount": 3599,
    "currency": "EUR",
    "lang": "en_US",
    "items": [
        {
            "amount": 2500,
            "quantity": 1,
            "sku": "XYZPART1",
            "name": "Item A",
            "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc eget venenatis diam. Integer euismod magna dui, a accumsan elit interdum."
        },
        {
            "amount": 200,
            "quantity": 2,
            "sku": "",
            "name": "Item B",
            "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Sed sit amet congue orci, ac euismod."
        }
    ],
    "fees": [
        {
            "feeAmount": 500,
            "feeName": "Setup Fee"
        }
    ],
    "taxes": [
        {
            "taxAmount": 199,
            "taxName": "VAT"
        }
    ],
    "paymentMethods": [
        {
            "type": "voucher",
            "value": "ukash"
        }
    ],
    "redirects": [
        {
            "rel": "on_success",
            "uri": "https://example.com/success.html"
        },
        {
            "rel": "on_cancel",
            "uri": "https://example.com/cancel.html"
        }
    ],
    "links": [
        {
            "url":
"https://api.neteller.com/v1/checkout/ORD_959b1148-704d-4d92-9a7d-14a18990a648",
            "rel": "hosted_payment",
            "method": "GET"
        },
        {
            "url":
"https://api.neteller.com/v1/orders/ORD_959b1148-704d-4d92-9a7d-14a18990a648",
            "rel": "self",
            "method": "GET"
            }
        ]
}
```

Lookup order

The details of an order can be queried from the API at any time. A *self* link is included in the order *link* section, which can be used to look up the status of the order. Alternatively, the order endpoint can be called directly as a GET request on the order *id*:

GET https://api.neteller.com/v1/orders/ORDER_ID

Scope required to initiate request: none (default)

Scope(s) impacting response: none (default)

Expandable Resources: none

Related Resources (if applicable): hosted_payment, invoice, payment

Sample Request

```
curl -X GET
https://api.neteller.com/v1/orders/ORD_7915d463-ccc8-4305-9d33-9e5c9310f12e\
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response:

```
{
    "orderId": "ORD_7915d463-ccc8-4305-9d33-9e5c9310f12e",
    "merchantRefId": "123456666",
    "totalAmount": 3599,
    "currency": "EUR",
    "lang": "en_US",
    "items": [
        {
            "quantity": 2,
            "name": "Item B",
            "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Sed sit amet congue orci, ac euismod.",
            "amount": 200
        }
    ],
    "fees": [
        {
            "feeName" : "Setup Fee",
            "feeAmount": 500
        }
    ],
    "taxes": [
        {
            "taxName": "VAT"
            "taxAmount": 199,
        }
    ],
    "paymentMethods": [
       {
           "type": "voucher",
           "value": "ukash"
       }
    ],
    "redirects": [
        {
            "rel": "on_success",
            "returnKeys": [
                "id"
            ],
            "uri": "https://example.com/success.html"
        },
        {
            "rel": "on_cancel",
            "returnKeys": [
                "id"
            ],
            "uri": "https://example.com/cancel.html"
        }
    ],
    "links": [
        {
           "url":
"https://api.neteller.com/v1/orders/ORD_7915d463-ccc8-4305-9d33-9e5c9310f12e",
           "rel": "self",
           "method": "GET"
        },
        {
            "url": "https://api.neteller.com/v1/checkout/ja3l3i_289lGk_3k2390",
            "rel": "hosted_payment",
            "method": "GET"
        },
        {
            "url":
"https://api.neteller.com/v1/orders/ORD_7915d463-ccc8-4305-9d33-9e5c9310f12e/invoice",
            "rel": "invoice",
            "method": "GET"
        },
        {
            "url": "https://api.neteller.com/v1/payments/154391457301406",
            "rel": "payment",
            "method": "GET"
        }
    ]
}
```

Lookup invoice for a order

GET https://api.neteller.com/v1/orders/ORDER_ID/invoice

Scope required to initiate request: none (default)

Scope(s) impacting response: none (default)

Expandable Resources: customer, order

Related Resources (if applicable): none

Sample Request

```
curl -X GET
https://api.neteller.com/v1/orders/ORD_7915d463-ccc8-4305-9d33-9e5c9310f12e/invoice \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response

```
{
    "invoiceNumber": 2915,
    "invoiceDate": "2014-06-01T00:00:00Z",
    "invoiceType": "order",
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "order": {
        "link": {
            "url":
"https://api.neteller.com/v1/orders/ORD_7915d463-ccc8-4305-9d33-9e5c9310f12e",
            "rel": "order",
            "method": "GET"
        }
    },
    "status": "paid",
    "totalAmount": 2495,
    "currency": "USD",
    "links": [
        "url":
"https://api.neteller.com/v1/orders/ORD_7915d463-ccc8-4305-9d33-9e5c9310f12e/invoice",
        "rel": "self",
        "method": "GET"
    ]
}
```

Plans

Subscription plans allow you to establish a fixed recurring payment schedule with your customers. Subscription Plans allow you define the pricing information and billing cycle for a subscription. You must first establish a subscription plan before you can offer it to your customers.

You can then enroll your customers by creating a new 'Subscription' request and NETELLER will automatically transfer recurring payments to your merchant account every period on their behalf.

Sub Sections:

- Create a plan
- Lookup plan
- Cancel a plan
- Delete a plan
- List plans

Resource URIs

Create a plan
POST https://api.neteller.com/v1/plans

Lookup plan
GET https://api.neteller.com/v1/plans/YOUR_PLAN_ID

Cancel a plan
POST https://api.neteller.com/v1/plans/YOUR_PLAN_ID/cancel

Delete a plan
DELETE https://api.neteller.com/v1/plans/YOUR_PLAN_ID

List plans
GET https://api.neteller.com/v1/plans

Create a plan

Creating a subscription plan allows you to define a plan for having automatic payments transferred to your merchant account on a fixed schedule.

POST https://api.neteller.com/v1/plans

Scope required to initiate request: none (default)

Scope(s) impacting response: none (default)

Expandable Resources: none

Related Resources (if applicable): none

This API request accepts a **plan** object.

Request Structure

Field Name	Field Type	Constraints	Required?	Description
planId	string	length <= 50	**required**	The source of funds for the transfer into your merchant account.
planName	string	length <= 200	**required**	The display name for the plan.
interval	number	length <= 3	**required**	The number of intervals between each billing attempt
intervalType	string	length <= 10	**required**	The frequency that the plan member will be billed (**daily**, **weekly**, **monthly**, **yearly**)
intervalCount	number	length <= 3	**required**	The length of the contract in intervals. (ie: to establish a 1 year subscription plan with quarterly charges, then interval = 3, intervalType = 'monthly' and intervalCount = 4.
amount	number			The amount to bill for each re-occurrence. Amount should be in the smallest unit of currency with no decimals. (ie: 14.99 EUR would be 1499)
currency	string	length = 3	**required**	The currency of the amount to be billed.

Sample Request

```
curl -X POST https://api.neteller.com/v1/plans \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
     -d '
{
    "planId": "MONTHLYGREENPLAN",
    "planName": "Sample Premier Monthly Membership",
    "interval": 3,
    "intervalType": "monthly",
    "intervalCount": 4,
    "amount": 2995,
    "currency": "EUR"
}'
```

Response Structure

Returns a **subscriptionPlan** object.

Sample Response (existing NETELLER account holder)

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
    "planId": "MONTHLYGREENPLAN",
    "planName": "Sample Premier Monthly Membership",
    "interval": 3,
    "intervalType": "monthly",
    "intervalCount": 4,
    "amount": 2995,
    "currency": "EUR",
    "status": "active",
    "links": [
        {
            "url": "https://api.neteller.com/v1/plans/MONTHLYGREENPLAN",
            "rel": "self",
            "method": "GET"
        }
    ]
 }
```

Lookup plan

GET https://api.neteller.com/v1/plans/YOUR_PLAN_ID

Scope required to initiate request: none (default)

Scope(s) impacting response: none (default)

Expandable Resources: none

Related Resources (if applicable): none

Request Parameters

Parameter Name	Required?	Description
planId	required	The unique identifier for the subscription plan.

Sample Request

```
curl -X GET https://api.neteller.com/v1/plans/MONTHLYGREENPLAN \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response

```
HTTP/1.1 204 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
    "planId": "MONTHLYGREENPLAN",
    "planName": "Sample Premier Monthly Membership",
    "interval": 3,
    "intervalType": "monthly",
    "intervalCount": 4,
    "amount": 2995,
    "currency": "EUR",
    "status": "active",
    "links": [
        {
            "url": "https://api.neteller.com/v1/plans/MONTHLYGREENPLAN",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

Cancel a plan

POST https://api.neteller.com/v1/plans/YOUR_PLAN_ID/cancel

Scope required to initiate request: none (default)

Scope(s) impacting response: none (default)

Expandable Resources: none

Related Resources (if applicable): none

Request Parameters

Parameter Name	Required?	Description
planId	required	The unique identifier for the subscription plan.

Sample Request

```
curl -X POST https://api.neteller.com/v1/plans/MONTHLYGREENPLAN/cancel \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
    "planId": "SAMPLEMONTHLY",
    "planName": "Sample Premier Monthly Plan",
    "interval": 3,
    "intervalType": "monthly",
    "intervalCount": 4,
    "amount": 2495,
    "currency": "EUR",
    "status": "cancelled",
    "links": [
        {
            "url": "https://api.neteller.com/v1/plans/SAMPLEMONTHLY",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

Delete a plan

You can only delete a subscription plan that has not been offered to any members. Once you have enrolled members in a subscription plan, you may choose to cancel the plan to prevent any further subscriptions from being created. Any existing subscriptions on a cancelled plan will continue to be charged until the end of their term.

DELETE https://api.neteller.com/v1/plans/YOUR_PLAN_ID

Scope required to initiate request: none (default)

Scope(s) impacting response: none (default)

Expandable Resources: none

Related Resources (if applicable): none

Request Parameters

Parameter Name	Required?	Description
planId	required	The unique identifier for the subscription plan.

Sample Request

```
curl -X DELETE https://api.neteller.com/v1/plans/MONTHLYGREENPLAN \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response

```
HTTP/1.1 204 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
```

List plans

GET https://api.neteller.com/v1/plans

Scope required to initiate request: none (default)

Scope(s) impacting response: none (default)

Expandable Resources: none

Related Resources (if applicable): none

Request Parameters

Parameter Name	Required?	Description
limit	**optional**	The number of records to be returned per page. Default = 10, Max = 100
offset	**optional**	Allows you to fetch the next set of resources

Sample Request

```
curl -X GET https://api.neteller.com/v1/plans \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response

```
HTTP/1.1 204 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
    "list": [
        {
            "planId": "MONTHLYGREENPLAN",
            "planName": "Sample Premier Monthly Membership",
            "interval": 3,
            "intervalType": "monthly",
            "intervalCount": 4,
            "amount": 2995,
            "currency": "EUR",
            "status": "active"
        },
        {...},
        {...},
    ],
    "links": [
        {
            "url":
"https://api.neteller.com/v1/plans?sessionid=234243242&limit=10&offset=10",
            "rel": "self",
            "method": "GET"
        },
        {
            "url":
"https://api.neteller.com/v1/plans?sessionid=234243242&limit=10&offset=0",
            "rel": "prev_offset",
            "method": "GET"
        },
        {
            "url":
"https://api.neteller.com/v1/plans?sessionid=234243242&limit=10&offset=20",
            "rel": "next_offset",
            "method": "GET"
        }
    ]
}
```

Subscriptions

Members must be subscribed to a subscription plan in order for NETELLER to issue recurring charges on your behalf. The "subscription" object represents a single enrollment of a member in a subscription plan.

Sub Sections:

- Create a subscription
- Lookup subscription
- Cancel subscription
- List subscriptions
- Lookup invoice for a subscription
- List all invoices for a subscription

Resource URIs:

Create a subscription
POST https://api.neteller.com/v1/subscriptions

Lookup subscription
GET https://api.neteller.com/v1/subscriptions/SUBSCRIPTION_ID

Cancel subscription
POST https://api.neteller.com/v1/subscriptions/SUBSCRIPTION_ID/cancel

List subscriptions
GET https://api.neteller.com/v1/subscriptions

Create a subscription

Creating a subscription is the act of enroiling an existing NETELLER account holder in one of your subscription plans. Once a subscription has been created, payments will be automatically distributed to your merchant account on every recurring billing cycle. In the event that a payment cannot be made (ie: the NETELLER account holder has insufficient funds available), we will retry to collect payment up to 3 times over a 7 day period. If the payment is still unsuccessful and all retry attempts have been exhausted, a final notification of the failed charge will sent via email to both you and the subscriber.   You can optionally configure to be notified via webhook from your merchant account portal.

In order to create a subscription you must first obtain member authorization for the required '**subscription_payment**' scope. See Obtain an access token using the auth_code grant type. Once authorized, you can then exchange the authorization code for an access token that will allow you to access this API.

POST https://api.neteller.com/v1/subscriptions

Scope required to initiate request: subscription_payment

Scope(s) impacting response: account_enhanced_profile, account_contact

Expandable Resources: plan**,** customer

Related Resources (if applicable): payment

This API request accepts a **subscription** object.

Request Structure

Field Name	Field Type	Constraints	Required?	Description
planId	string	length <= 50	**required**	
accountProfile	accountProfile		**required**	
startDate	string	length <= 20	*optional*	An optional start date for the subscription. If the subscription is to start immediately, this should be left blank and enrollment will immediately attempt to charge the initial payment. If the initial payment fails for any reason, then an error will be returned and the subscription will not be started. ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ

Sample Request

```
curl -X POST https://api.neteller.com/v1/subscriptions \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
     -d '
{
    "planId": "SAMPLEMONTHLY",
    "accountProfile": {
        "email": "johndoe@email.com"
    },
    "startDate": "2014-06-01T00:00:00Z"
}'
```

Response Structure

Returns a **subscription** object.

Sample Response (existing NETELLER account holder)

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
    "subscriptionId": "588",
    "plan": {
        "link": {
            "url": "https://api.neteller.com/v1/plans/DEMOGOLDPLAN",
            "rel": "plan",
            "method": "GET"
        }
    },
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUS_FCBB852E-CAF6-4109-8AE1-40100D3ECE12",
            "rel": "customer",
            "method": "GET"
        }
    },
    "status": "active",
    "startDate": "2014-11-20T23:52:37Z",
    "endDate": "2015-11-19T23:52:37Z",
    "currentPeriodStart": "2014-11-20T23:52:35Z",
    "currentPeriodEnd": "2014-11-27T23:52:35Z",
    "cancelAtPeriodEnd": "false",
    "cancelDate": null,
    "links": [
        {
            "url": "https://api.neteller.com/v1/subscriptions/588",
            "rel": "self",
            "method": "GET"
        },
        {
            "url": "https://api.neteller.com/v1/payments/138416527555633",
            "rel": "payment",
            "method": "GET"
        }
    ]
}
```

Lookup subscription

Retrieve information about a previously created subscription by passing the relevant subscription identifier.

GET https://api.neteller.com/v1/subscriptions/SUBSCRIPTION_ID

Scope required to initiate request: none (default)

Scope(s) impacting response: account_enhanced_profile, account_contact

Expandable Resources: plan**,** customer

Related Resources (if applicable): none

Sample Request

```
curl -X GET https://api.neteller.com/v1/subscriptions/234234224 \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response

```
{
    "subscriptionId": "234234224",
    "plan": {
        "link": {           {
                "url": "https://api.neteller.com/v1/plans/MONTHLYGREENPLAN",
                "rel": "plan",
                "method": "GET"
        }
    },
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUST_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "status": "active",
    "startDate": "2014-06-01T00:00:00Z",
    "endDate": "2014-12-31T00:00:00Z",
    "termEndDate": "2014-12-31",
    "currentPeriodStart": "2014-07-01T00:00:00Z",
    "currentPeriodEnd": "2014-07-31T00:00:00Z",
    "cancelAtPeriodEnd": false,
    "cancelDate": null,
    "lastCompletedPaymentDate": "2014-06-31T00:00:00Z",
    "links": [
        {
            "url": "https: //api.neteller.com/v1/subscriptions/234234224",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

Sample Request with expanded plan

```
curl -X GET https://api.neteller.com/v1/subscriptions/234234224?expand=plan\
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response with expanded plan

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
    "subscriptionId": "234234224",
    "plan": {
        "planId": "MONTHLYGREENPLAN",
        "planName": "Sample Premier Monthly Plan",
        "interval": 3,
        "intervalType": "monthly",
        "intervalCount": 4,
        "amount": 2495,
        "currency": "EUR",
        "status": "active"
    },
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUST_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "status": "active",
    "startDate": "2014-06-01T00:00:00Z",
    "endDate": "2014-12-31T00:00:00Z",
    "termEndDate": "2014-12-31",
    "currentPeriodStart": "2014-07-01T00:00:00Z",
    "currentPeriodEnd": "2014-07-31T00:00:00Z",
    "cancelAtPeriodEnd": false,
    "cancelDate": null,
    "lastCompletedPaymentDate": "2014-06-31T00:00:00Z",
    "links": [
        {
            "url": "https: //api.neteller.com/v1/subscriptions/234234224",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

Cancel subscription

By default when you cancel a subscription, the NETELLER subscription status is updated immediately to reflect a cancelled state. If you wish to delay cancellation until the end of the current period, pass in the optional cancelAtPeriod end flag and the subscription will be cancelled just before the start of the next billing cycle.

POST https://api.neteller.com/v1/subscriptions/SUBSCRIPTION_ID/cancel

Scope required to initiate request: none (default)

Scope(s) impacting response: account_enhanced_profile, account_contact

Expandable resources: plan**,** customer

Related resources (if applicable): none

Request Parameters

Parameter Name	Required?	Description
subscriptionId	required	The unique identifier for the subscription.
cancelAtPeriodEnd	optional	An optional parameter that you can set to **true** if you want to delay the cancellation of the subscription until the end of the current period. By default, if this is not passed the subscription will be cancelled immediately. As soon as you cancel a subscription, the member will no longer be billed.

Sample Request

```
curl -X POST https://api.neteller.com/v1/subscriptions/234234224/cancel \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
    "subscriptionId": "234234224",
    "plan": {
        "link": {
            "url": "https://api.neteller.com/v1/plans/MONTHLYGREENPLAN",
            "rel": "plan",
            "method": "GET"
        }
    },
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUST_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "status": "cancelled",
    "startDate": "2014-06-01T00:00:00Z",
    "endDate": "2014-12-31T00:00:00Z",
    "currentPeriodStart": "2014-07-01T00:00:00Z",
    "currentPeriodEnd": "2014-07-31T00:00:00Z",
    "cancelAtPeriodEnd": false,
    "cancelDate": "2014-06-04T12:24:02Z",
    "links": [
        {
            "url": "https://api.neteller.com/v1/subscriptions/234234224",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

List subscriptions

You can obtain a list of all your active subscriptions. Newest subscriptions will be returned at the top of the list (sorted newest -> oldest).

GET https://api.neteller.com/v1/subscriptions

Scope required to initiate request: none (default)

Scope(s) impacting response: account_enhanced_profile, account_contact

Expandable resources: none

Related resources (if applicable): none

Request Parameters

Parameter Name	Required?	Description
limit	**optional**	The number of records to be returned. Default = 10, Max = 100
offset	**optional**	Allows you to fetch the next set of resources

Sample Request

```
curl -X GET https://api.neteller.com/v1/subscriptions \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
    "list": [
        {
            "subscriptionId": "234234224",
            "plan": {
                "link": {
                    "url": "https://api.neteller.com/v1/plans/MONTHLYGREENPLAN",
                    "rel": "plan",
                    "method": "GET"
                }
            },
            "customer": {
                "link": {
                    "url":
"https://api.neteller.com/v1/customers/CUST_0d676b4b-0eb8-4d78-af25-e41ab431e325",
                    "rel": "customer",
                    "method": "GET"
                }
            },
            "status": "active",
            "startDate": "2014-06-01T00: 00: 00Z",
            "endDate": "2014-12-31T00: 00: 00Z",
            "termEndDate": "2014-12-31",
            "currentPeriodStart": "2014-07-01T00: 00: 00Z",
            "currentPeriodEnd": "2014-07-31T00: 00: 00Z",
            "cancelAtPeriodEnd": false,
            "cancelDate": null,
            "lastCompletedPaymentDate": null
        },
        {},
        {}
    ],
    "links": [
        {
            "url":
"https://api.neteller.com/v1/subscriptions?session=1212&limit=10&offset=10",
            "rel": "self",
            "method": "GET"
        },
        {
            "url":
"https://api.neteller.com/v1/subscriptions?session=1212&limit=10&offset=0",
            "rel": "prev_offset",
            "method": "GET"
        },
        {
            "url":
"https://api.neteller.com/v1/subscriptions?session=1212&limit=10&offset=20",
            "rel": "next_offset",
            "method": "GET"
        }
    ]
}
```

Lookup invoice for a subscription

GET https://api.neteller.com/v1/subscriptions/SUBSCRIPTION_ID/invoices/INVOICE_NUMBER

Scope required to initiate request: none (default)

Scope(s) impacting response: none (default)

Expandable resources: customer, subscription

Related resources (if applicable): none

Sample Request

```
curl -X GET https://api.neteller.com/v1/subscriptions/234234224/invoices/3020 \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Sample Response

```
 {
    "invoiceNumber": 3020,
    "invoiceDate": "2014-06-01T00:00:00Z",
    "invoiceType": "subscription",
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUST_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "subscription": {
        "link": {
            "url": "https://api.neteller.com/v1/subscription/234234224",
            "rel": "subscription",
            "method": "GET"
        }
    },
    "status": "paid",
    "periodStartDate": "2014-06-01T00:00:00Z",
    "periodEndDate": "2014-06-31T00:00:00Z",
    "totalAmount": 2495,
    "currency": "USD",
    "retryCount": 1,
    "nextRetryDate": null,
    "links": [
        {
            "url": "https://api.neteller.com/v1/subscriptions/234234224/invoices/3020",
            "rel": "self",
            "method": "GET"
        }
    ]
 }
```

List all invoices for a subscription

GET https://api.neteller.com/v1/subscriptions/SUBSCRIPTION_ID/invoices

Scope required to initiate request: none (default)

Scope(s) impacting response: none (default)

Expandable resources: none

Related resources (if applicable): none

Sample Request

```
curl -X GET https://api.neteller.com/v1/subscriptions/SUBSCRIPTION_ID/invoices \
     -H "Content-Type:application/json" \
     -H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Request Parameters

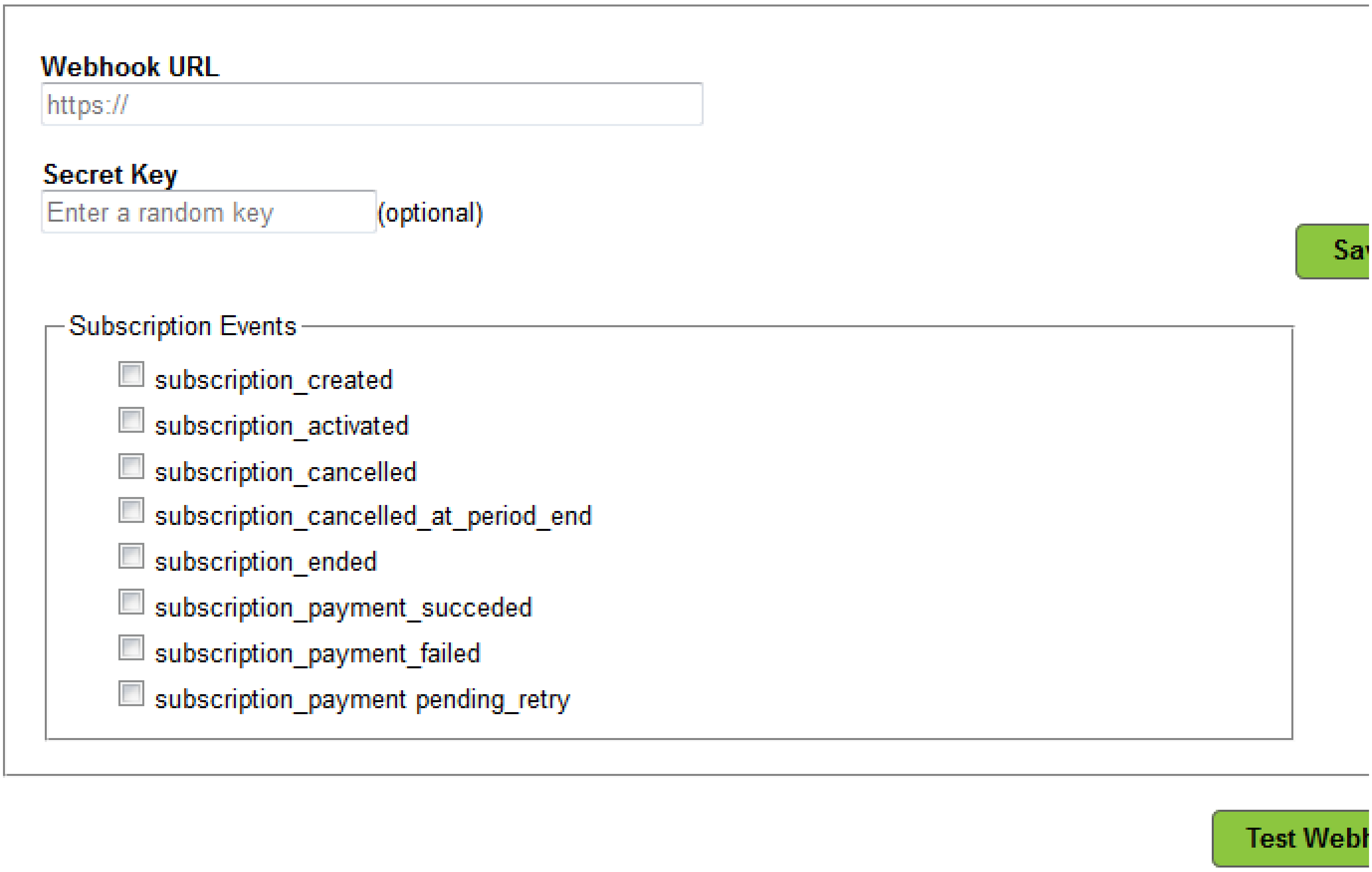
Parameter Name	Required?	Description
limit	optional	The number of records to be returned. Default = 10, Max = 100
offset	optional	Allows you to fetch the next set of resources

Sample Response

```
{
    "list": [
        {
            "invoiceNumber": 2915,
            "invoiceDate": "2014-06-01T00:00:00Z",
            "invoiceType": "subscription",
            "customer": {
                "link": {
                    "url":
"https://api.neteller.com/v1/customers/21EC2020-3AEA-4069-A2DD-08002B30309D",
                    "rel": "customer",
                    "method": "GET"
                }
            },
            "subscription": {
                "link": {
                    "url": "https://api.neteller.com/v1/subscription/234234224",
                    "rel": "subscription",
                    "method": "GET"
                }
            },
            "status": "paid",
            "periodStartDate": "2014-06-01T00:00:00Z",
            "periodEndDate": "2014-06-31T00:00:00Z",
            "amountDue": 2495,
            "currency": "USD",
            "retryCount": 1,
            "nextRetryDate": null
        },
        {},
        {}
    ],
    "links": [
        {
            "url":
"https://api.neteller.com/v1/subscriptions/234234224/invoices?session=1212&limit=10&off
set=10",
            "rel": "self",
            "method": "GET"
        },
        {
            "url":
"https://api.neteller.com/v1/subscriptions/234234224/invoices?session=1212&limit=10&&of
fset=0",
            "rel": "prev_offset",
            "method": "GET"
        },
        {
            "url":
"https://api.neteller.com/v1/subscriptions/234234224/invoices?session=1212&limit=10&off
set=20",
            "rel": "next_offset",
            "method": "GET"
        }
    ]
}
```

Webhooks

A webhook allows you to define an HTTP callback that will be POST to when an event occurs that you would like to be notified of. You can define the notification URL and select which events you wish to be notified of from your merchant account portal.



Webhook URL

https://

Secret Key

Enter a random key (optional)

Subscription Events

- □ subscription_created
- □ subscription_activated
- □ subscription_cancelled
- □ subscription_cancelled_at_period_end
- □ subscription_ended
- □ subscription_payment_succeded
- □ subscription_payment_failed
- □ subscription_payment pending_retry

Responding to a webhook

To acknowledge that you received the webhook without any problem, your server should return a 200 HTTP status code. Any other information you return in the request headers or request body will be ignored. Any response code outside the 200 range, including 3xx codes, will indicate to NETELLER that you did not receive the webhook.

> ⚠ If your system did not respond with a valid HTTP 200 status, NETELLER will continue to retry the request with an escalating time delay for up to 48 hours

Secure webhooks (recommended)

NETELLER recommends that you use one of the following forms of authentication on your webhook URLs

HTTP Basic Authentication - Use basic authentication syntax in your webhook URL:

Format : https://{username}:{password}@{webhook_url}
Example : https://ntwhuser:NT1p2dsl@myapp.com/netellerwebhook

Secret Key - Provide a secret key that is passed back as part of the webhook event notification body.

Format : https://{webhook_url}
Example : https://myapp.com/netellerwebhook
Body:

The secret key should not be shared with anyone.

Event Types	
subscription_activated	Subscription has moved from an 'pending' to 'active' state
subscription_cancelled	Subscription was cancelled
subscription_cancelled_at_period_end	Subscription is set to be cancelled at the end of the current billing period
subscription_created	Subscription was created
subscription_ended	Subscription has lapsed and is now ended
subscription_payment_failed	Attempts to collect payment for the subscription for the current billing period have failed
subscription_payment_pending_retry	Member has insufficient funds available, NETELLER will continue to try to collect payment for this period
subscription_payment_succeeded	Payment was successful

Sample webhook response

```
{
     "mode": "live",
     "id": "ebecd052-757f-4991-9f19-469d21e6c065",
     "eventDate": "2014-01-01T00:00:00Z",
     "eventType": "subscription_payment_succeeded",
     "attemptNumber": 1,
     "key": "23haJ20opHJ2ks38aGEnw",
     "links": [
         {
             "url":
"https://api.neteller.com/v1/subscriptions/234234224/invoices/99102",
             "rel": "invoice",
             "method": "GET"
         }
     ]
}
```

When sending a Test Webhook request from the merchant portal, the links section will be excluded from the response as their is no valid resource reference to send.

JSON Objects

- accountPreferences
- accountProfile
- attribute
- balance
- contactDetail
- customer
- date
- error
- event
- fee
- fieldError
- invoice
- item
- link
- order
- payment
- paymentMethod
- plan
- redirect
- subscription
- tax
- token
- transaction

Understanding the JSON object structure
Depending on the API endpoint you are calling, your request may consist of standard HTTP parameters, or a JSON object passed in the request body. By default, the root of a JSON object is not enveloped unless required.

For example, when looking up a payment with the url /v1/payments/3001, the payment object returned in the response will look like this:

```
{ "id": "3001", ..... }
```

as opposed to:

```
{ "payment" :
    { "id": "3001", ..... }
}
```

Field Legend
✓ Valid / Required
✓ Optional
✗ Invalid / Not Returned

accountPreferences

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
lang	string	length = 5	✓	✓	default	The preferred language of the member. see Languages for complete list
currency	string	length = 3	✓	✓	default	The preferred wallet currency of the member. Member can optionally modify this to whatever they desire during the registration process. see Currencies for complete list

```
{
    "lang": "en_US",
    "currency": "EUR"
}
```

accountProfile

****NOTE:** Most accountProfile values are OPTIONAL inputs for payments to non-registered members (transferOut) as these values are only used if the recipient is not already a NETELLER account holder. Every field you can supply will reduce the amount of information the recipient is required to re-enter during setup of their NETELLER account.

If the recipient email is already found in the NETELLER system, the supplied profile data will be ignored. Future lookups for this transfer will return relevant recipientProfile data for correlation with your external system.

Field Name	Field Type	Constraints	Request Field	Response Field	**Response Field Scope**	Description
email	string	length <= 100	✓	✓	default	The accour address.
accountId	string	length <= 15	✓	✓	default	The 12 digit NETEL account nu
firstName	string	length <= 25	✓	✓	default	The clients name, or gi name.
lastName	string	length <= 25	✓	✓	default	The clients name, or fa name.
address1	string	length <= 35	✓	✓	account_enhanced_profile	The clients residential, address.
address2	string	length <= 35	✓	✓	account_enhanced_profile	Continuatio clients resic street addre
address3	string	length <= 35	✓	✓	account_enhanced_profile	Continuatio clients resic street addre
city	string	length <= 50	✓	✓	account_enhanced_profile	The clients residence.
countrySubdivisionCode	string		✓	✓	account_enhanced_profile	The ISO 31 code indica state/provin or other val denoting th country sub (e.g. BE=Be 13=Tôkyô [AB=Alberta
country	string	length <= 2	✓	✓	default	The ISO 31 Alpha 2-coc clients cour residence (Germany = JP=Japan. CA=Canad
postCode	string	length <= 10	✓	✓	account_enhanced_profile	The zip cod postal code clients resic

contactDetails	array of contactDetail		✓	✓	account_contact	An array of numbers. C accept up t numbers, th number will considered Phone num should cont special cha
gender	string	length <= 1	✓	✓	account_enhanced_profile	m - male, f
dateOfBirth	date		✓	✓	default	
accountPreferences	accountPreferences		✓	✓	account_enhanced_profile	

```
{
    "email": "jsmith@email.com",
    "accountId": "451234567890",
    "firstName": "John",
    "lastName": "Smith",
    "address1": "addressline1",
    "address2": "addressline2",
    "address3": "addressline3",
    "city": "Calgary",
    "countrySubdivisionCode": "",
    "postCode": "T8A22J",
    "contactDetails": [
        {
            "type": "landLine",
            "value": "14032332333"
        },
        {
            "type": "mobile",
            "value": "14035552333"
        }
    ],
    "gender": "m",
    "dateOfBirth": {
        "year": "1975",
        "month": "01",
        "day": "31"
    },
    "accountPreferences": {
        "lang": "en",
        "currency": "EUR"
    }
}
```

attribute

Attributes allows you to specify additional data (e.g., affiliate codes, session IDs, etc.) to be stored against the request and returned to in redirection urls, webhooks, status calls, reports, etc.

Up to 10 addendum data key/value pairs are enabled per order. They do not need to be defined in advance and can be different on each transaction.

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
key	string	length <= 50	✓	✓	default	This is a unique identifier for an attribute.
value	string	length <= 100	✓	✓	default	The value for this attribute.

```
{
    "key": "affiliate_code",
    "value": "test12345"
}
```

balance

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
amount	string	length <= 100	✗	✓	default	The amount in the smallest unit of currency.
currency	string	length <= 15	✗	✓	default	The associated currency.

```
{
    "amount": 3242,
    "currency": "EUR"
}
```

contactDetail

The contactDetail object includes an array of phone numbers. Only 2 numbers are permitted at this time. The first number will be considered the primary contact and the second the alternate.

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
type	string	length <= 10	✓	✗	default	landLine, mobile ** The values are case sensitive.
value	string	length <= 30	✓	✗	default	The dialing number or recipients id. Do not include the country code in the dialing number.

```
[
    {
        "type": "landLine",
        "value": "4039971234"
    }
]
```

customer

Field Name	Field Type	Constraints	Request Field	Response Field	**Response Field Scope**	Description
customerId	string	length <=100	✓	✓	default	Unique identifier for this customer. ie. CUS_0d676b4b-0eb8-4d78-af25-e
accountProfile	accountProfile	length <= 100	✓	✓	default	The associated account profile.
verificationLevel	string	length = 2	✗	✓	default	00 - Member has not been verified registered payment instruments 01 - Member has not been verified more registered payment instrume 10 - Member is verified, no registe instruments 11 - Member is verified, has one o registered payment instruments
availableBalance	balance		✗	✓	account_available _balance	The members currently available b

```
{
    "customerId": "CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
    "accountProfile": {
        "email": "jsmith@email.com",
        "accountId": "451234567890",
        "firstName": "John",
        "lastName": "Smith",
        "country": "US",
        "dateOfBirth": {
            "year": "1975",
            "month": "01",
            "day": "31"
        }
    },
    "verificationLevel": "01"
}
```

date

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
year	string	length <= 4	✓	✓	default	4 digit year, YYYY
month	string	length <= 2	✓	✓	default	2 digit month, MM
day	string	length <= 2	✓	✓	default	2 digit day, DD

```
{
   "year": "1975",
   "month": "01",
   "day": "31"
}
```

error

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
code	string	100	✗	✓	default	The error code reflecting the reason for the failed request.
message	string	250	✗	✓	default	Short description of the status.
fieldErrors	array of fieldErrors		✗	✓	default	If applicable, details around particular fields that reported errors with the API request.

```
{
    "code": "5018",
    "message": "FieldErrors",
    "fieldErrors": [
        {
            "field": "fieldA",
            "error": "missing required value"
        },
        {
            "field": "fieldC",
            "error": "missing required value"
        }
    ]
}
```

event

Events are used in webhook responses to notify your registered callback uri of activities that have occurred relating to your account.

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
mode	string	length <= 10	✗	✓	default	test - denotes this is a test webhook notification that was manually triggered live - this is a live production webhook notification
eventDate	string	length <= 50	✗	✓	default	The date the event occurred. ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ
eventType	string	length <= 50	✗	✓	default	The specific type of event that occurred.
attemptNumber	string	length <= 50	✗	✓	default	The current attemptNumber for this notification. This will increment until we receive a HTTP 200 response on send, or until all retries have been exhausted.
key	string	length <= 55	✗	✓	default	The optional secret key that you defined with your webhook. This should be validated within your integration for added security.
links	array of link		✗	✓	default	Associated HATEOAS links for this event. Use these links to lookup the details for this event.

```
{
     "mode": "test",
     "eventDate": "2014-01-01T00:00:00Z",
     "eventType": "subscription_payment_succeeded",
     "attemptNumber": 1,
     "links": [
         {
             "url": "https://api.neteller.com/v1/payments/434632223423",
             "rel": "payment",
             "method": "GET"
         },
         {
             "url":
"https://api.neteller.com/v1/subscriptions/234234224/invoices/99102",
             "rel": "invoice",
             "method": "GET"
         }
     ]
 }
```

fee

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
feeName	string	length <= 50	✓	✓	default	The name that will display in history and invoice details for this fee item. (ie: Setup Fee, Shipping & Handling Fee...etc)
feeType	string	length <= 100	✗	✓	default	The type of fee. (Service_fee)
feeAmount	number	length <= 10	✓	✓	default	The fee amount that was deducted. Amount fields reflect the smallest unit of currency with no decimals. Eg. $25.00 USD should be formatted as 2500.
feeCurrency	string	length = 3	✗	✓	default	The currency the fee is in. see Currencies for complete list

* fields will only be returned if the data is available at time of request. Transfers that are in a pending state will not have fee's calculated until they are accepted.

```
{
    "feeName": "Setup Fee",
    "feeType": "service_fee",
    "feeAmount": 50,
    "feeCurrency": "USD"
}
```

fieldError

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
field	string	length <= 100	✗	✓	default	Identifies the JSON request field
error	string	length <= 250	✗	✓	default	The problem associated with field

```
{
    "field": "transaction.accountProfile.email",
    "error": "Required field"
}
```

invoice

Invoices are statements containing charges for a specific term.

- When a customer pays for an order, a invoice will be created detailing the purchase details for the order.
- When you create a subscription a invoice is used to track each re-curring charge to the customer.

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
invoiceNumber	string	length <= 30	✗	✓	default	Unique identifier for this invoice.
invoiceDate	string	length <= 20	✗	✓	default	Date the invoice was created (first issued). ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ
invoiceType	string	string <= 15	✗	✓	default	**subscription** - invoice is auto created from subscription billing **order** - invoice is manually created through a create order api request
customer	customer		✗	✓	default	The associated customer for this invoice. An order invoice may not yet be associated to a customer if it hasn't been paid.
subscription	subscription		✗	✓	default	The corresponding subscription resource this invoice belongs to ** Only applicable for subscription invoices
order	order		✓	✓	default	The corresponding subscription resource this invoice belongs to ** Only applicable for order invoices
status	string	length <= 15	✗	✓	default	payment_due - Indicates that a due payment has not yet been collected and will be re-tried paid - Indicates the payment was successful not_paid - Indicates the payment was not collected and all attempts to collect payment failed
periodStartDate	string	length <= 20	✗	✓	default	Date the period for this invoice started ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ ** Only applicable for subscription invoices
periodEndDate	string	length <= 20	✗	✓	default	Date the period for this invoice ended ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ ** Only applicable for subscription invoices

totalAmount	number		✗	✓	default	The total amount due reported in the smallest unit of currency (no decimals).
currency	string	length = 3	✗	✓	default	The currency for the amount due. see Currencies for complete list
retryCount	number		✗	✓	default	The total number of additional attempts made to try to collect payment for this invoice. If charge failed due to NSF, the system will automatically retry the transaction up to 3 times. ** Only applicable for subscription invoices
nextRetryDate	string	length <= 20	✗	✓	default	The date the charge will be re-attempted. ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ ** Only applicable for subscription invoices

Sample subscription invoice

```
{
    "invoiceNumber": 2915,
    "invoiceDate": "2014-06-01T00:00:00Z",
    "invoiceType": "subscription",
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "subscription": {
        "link": {
            "url": "https://api.neteller.com/v1/subscription/234234224",
            "rel": "subscription",
            "method": "GET"
        }
    },
    "status": "paid",
    "periodStartDate": "2014-06-01T00:00:00Z",
    "periodEndDate": "2014-06-31T00:00:00Z",
    "totalAmount": 2495,
    "currency": "USD",
    "retryCount": 1,
    "nextRetryDate": null
 }
```

Sample order invoice

```
{
    "invoiceNumber": 2915,
    "invoiceDate": "2014-06-01T00:00:00Z",
    "invoiceType": "order",
    "merchantRefId": "1231412412421",
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "order": {
        "items": [
            {
                "quantity": 1,
                "name": "Item A",
                "description": "Lorem ipsum nunc eget venenatis diam. Integer euismod
magna dui, a accumsan elit interdum.",
                "sku": "XYZPART1",
                "amount": 2500
            },
            {
                "quantity": 2,
                "name": "Item B",
                "description": "Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Sed sit amet congue orci, ac euismod.",
                "amount": 200
            }
        ],
        "fees": [
            {
                "feeName": "Setup Fee",
                "feeAmount": 500
            }
        ],
        "taxes": [
            {
                "taxName": "VAT",
                "taxAmount": 199
            }
        ]
    },
    "status": "paid",
    "totalAmount": 2495,
    "currency": "USD"
}
```

item

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
amount	string	length <= 15	✓	✓	default	
quantity	number	length <= 3	✓	✓	default	
sku	string	length <= 50	✓	✓	default	
name	string	length <= 50	✓	✓	default	
description	string	length <= 150	✓	✓	default	

```
{
    "amount": 500,
    "quantity": 2,
    "sku": "id/sku-1",
    "name": "Product A",
    "description": "This is a brief description of Product A."
}
```

link

Every API call response may include an array of HATEOAS (Hypermedia as the Engine of Application State) links. You can use these links within your own application to control your own page flow and minimize the need to change your integration in the future.

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
url	string	length <= 500	✗	✓	default	This is the URL to which to send the link. If this is an HTTPS address, please ensure that your certificate is valid – otherwise, the link will cause the browser to display a certificate warning.
rel	string	length <= 100	✗	✓	default	This is the link type, allowing different endpoints to be targeted depending on the end state of the transaction. The link relation describes how this link relates to the previous call. - self will get details of the current call
method	string	length <= 6	✗	✓	default	The HTTP method required for the related call

```
{
    "url": "https://api.neteller.com/v1/payments/342342334",
    "rel": "self",
    "method": "GET"
}
```

order

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
orderId	string	length <= 100	✗	✔	default	Unique NETELLER reference for the ORD_7915d463-ccc8-4305-9d33-9e5 SUB_c46645ea-b7b0-4a91-87f7-88c(
merchantRefId	string	length <= 50	✔	✔	default	The associated merchant reference ic initial request.
totalAmount	number		✔	✔	default	The total amount due for this order, in items, fees, taxes.
currency	string	length <= 3	✔	✔	default	The currency of the total amount due. must have the same currency for all it and taxes.
lang	string	length <= 5	✔	✔	default	The language that the Quick Checkou should be displayed in. see Languages for complete list
items	Array of Item		✓	✓	default	Specific line items (products/fees or s that you are requesting payment for.
fees	Array of fee		✓	✓	default	Specific fees (setup/shipping&handlir that are to be included in the payment
taxes	Array of tax		✓	✓	default	Specific taxes(VAT, GST, PST, etc... to be included in the payment.
customerIp	string	length <= 20	✓	✗		The value should be the incoming IP a your customer, which will restrict the p page to being viewable only from this address.
✓ Though the *customerIp* parameter is not mandatory, NETELLER recommends including it for additional security.						
paymentMethods	array of paymentMethod		✓	✓	default	This allows you to define which paym you would like to use
redirects	array of redirect		✓	✓	default	This allows you to add redirects to the Redirects cause a connection back to merchant system via the customer's b

links	array of link		✗	✓	default	Array of HATEOAS links (if applicable • hosted_payment – URI for the pa page. Customers should be redire this URI to see their invoice and c payment; or this should be used a action on the merchant payment p using a silent post. • self – This URI can be called to re JSON object about the current sta order, e.g., to see if the payment settled. • resend_callback – If a callback UI specified in the initial request, cal URI will re-queue the callback and (which is useful if your callback re run out)

```
{
    "orderId": "ORD_0d676b4b-0eb8-4d78-af25-e41ab431e325",
    "merchantRefId": "124124910192",
    "totalAmount": 3599,
    "currency": "EUR",
    "lang": "en_US",
    "items": [
        {
            "quantity": 1,
            "name": "Item A",
            "description": "Lorem ipsum nunc eget venenatis diam. Integer euismod magna
dui, a accumsan elit interdum.",
            "sku": "XYZPART1",
            "amount": 2500
        },
        {
            "quantity": 2,
            "name": "Item B",
            "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Sed sit amet congue orci, ac euismod.",
            "amount": 200
        }
    ],
    "fees": [
        {
            "feeName": "Setup Fee",
            "feeAmount": 500
        }
    ],
    "taxes": [
        {
            "taxName": "VAT",
            "taxAmount": 199
        }
    ]
    "paymentMethods": [
       {
           "type": "voucher",
           "value": "ukash"
       }
    ],
    "redirects": [
        {
            "rel": "on_success",
            "returnKeys": [
                "id"
            ],
            "uri": "https://example.com/success.html"
        },
        {
            "rel": "on_cancel",
            "returnKeys": [
                "id"
            ],
            "uri": "https://example.com/cancel.html"
        }
    ],
    "links": [
        {
            "url":
"https://api.neteller.com/v1/checkout/ORD_959b1148-704d-4d92-9a7d-14a18990a648",
            "rel": "hosted_payment",
            "method": "GET"
        },
        {
            "url":
"https://api.neteller.com/v1/orders/ORD_959b1148-704d-4d92-9a7d-14a18990a648",
            "rel": "self",
            "method": "GET"
        }
    ]
}
```

payment

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
customer	customer		✓	✓	default	Member account details for this transaction.
transaction	transaction		✓	✓	default	Associated transaction details.

```
{
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUS_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "transaction": {
        "merchantRefId": "9876543210",
        "amount": 2500,
        "currency": "USD",
        "id": "288382647322027",
        "createDate": "2013-06-22T14:33:12",
        "updateDate": "2013-06-22T16:21:52",
        "status": "approved",
        "fees": [
            {
                "feeType": "service_fee",
                "feeAmount": 50,
                "feeCurrency": "USD"
            }
        ]
    }
}
```

paymentMethod

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
type	string	length <= 15	✓	✗	default	The source of funds for this payment.
value	string	length <= 100	✓	✗	default	The associated value for this payment type.

Permitted Combinations

Type	Value
neteller	The account holders 12 digit NETELLER Account ID.**OR** The NETELLER account holders registered email address. ⚠ **You can only use NETELLER Account ID if the recipient already has a registered NETELLER account.**
voucher	ukash
onlinebanking	sofortbanking
onlinebanking	poli
onlinebanking	giropay

```
{
    "type": "neteller",
    "value": "jsmith@email.com"
}

{
    "type": "onlinebanking",
    "value": "sofort"
}
```

plan

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
planId	string	length <= 50	✓	✓	default	A unique identifier for the plan that you will use to identify which plan you want to enroll your customer in.
planName	string	length <= 200	✓	✓	default	The display name for the plan.
interval	number	length <=3	✓	✓	default	The number of intervals between each billing attempt. (ie: to bill every 15 days, you would select an interval of 15 and a type of daily)
intervalType	string	length <=10	✓	✓	default	The type of interval you wish want to define. daily weekly monthly yearly
intervalCount	number	length <=3	✓	✓	default	The length of the contract in intervals, or how many recurring billings will be attempted. intervalCount * interval = Total Subscription Term (duration) (ie: For a yearly subscription billing every 3 months, this would be 4)
amount	number		✓	✓	default	The amount to bill for each re-ocurrence.
currency	string	length = 3	✓	✓	default	The currency of the amount to be billed.
status	string	length <=15	✗	✓	default	pending - The subscription has been requested and is pending activation (start date has not yet elapsed). active - The subscription plan is currently active cancelled - The subscription plan was cancelled and can no longer be subscribed to

```
{
    "planId": "SAMPLEMONTHLY",
    "planName": "Sample Premier Monthly Plan",
    "interval": 3
    "intervalType": "monthly",
    "intervalCount": 4,
    "amount": 2495,
    "currency": "EUR".
    "status": "active"
}
```

redirect

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
rel	string	length <= 15	✓	✗	default	This is the redirect type, allowing different endpoints to be targeted depending on the end state of the transaction. Possible values are: • on_success • on_error • on_decline • on_cancel • on_timeout
returnKeys	Array of string	length <= 100	✓	✗	default	.
uri	string		✓	✗	default	
delimiter	string		✓	✗	default	This specifies the delimiter to use when parameters are sent to the redirect URI (e.g, “;” “&” “$”). By default an ampersand (&) is used

⚠ **Never rely on a redirection to determine the end state of a transaction, as they can fail or be tampered with. Always use a callback to obtain the transaction status.**

```
{
    "rel": "on_success",
    "returnKeys": [
        "id"
    ],
    "uri": "https://example.com/success.html"
}
```

subscription

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
subscriptionId	string	length <= 50	✗	✓	default	A system generated value that uniquely identifies this subscription.
plan	plan	length <= 50	✓	✓	default	The plan identifier you want to subscribe the customer to.
customer	customer		✓	✓	default	The associated customer for this subscription.
status	string	length <= 15	✗	✓	default	pending - A subscription has been requested and is pending activation upon the requested start date active - The subscription is currently active cancelled - The subscription was cancelled by the subscriber or by the merchant application ended - The subscription has lapsed failed - The subscription request failed. This status will only be set if the subscription was created with no startDate supplied and the immediate billing failed.
startDate	string	length <= 20 > current date	✓	✓	default	Date the subscription should start. If startDate is not supplied, the customer will be billed immediately. If supplying a subscription startDate, the date must be greater then the current date. ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ
endDate	string	length <= 20	✗	✓	default	If the subscription has ended (either because it was canceled or because the customer was switched to a subscription to a new plan), the date the subscription ended ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ

currentPeriodStart	string	length <= 20	✗	✓	default	Start of the current period that the subscription has been invoiced for ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ
currentPeriodEnd	string	length <= 20	✗	✓	default	End of the current period that the subscription has been invoiced for. At the end of this period, a new invoice will be created ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ
cancelAtPeriodEnd	boolean	true false	✗	✓	default	If the subscription has been canceled with the atPeriodEnd flag set to true, then this value on the subscription will be true. You can use this attribute to determine whether a subscription that has a status of active is scheduled to be canceled at the end of the current period.
canceledDate	string	length <= 20	✗	✓	default	If the subscription has been canceled, the date of that cancellation. If the subscription was canceled with cancelAtPeriodEnd, canceledDate will still reflect the date of the initial cancellation request, not the end of the subscription period when the subscription is automatically moved to a canceled state. ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ
lastCompletedPaymentDate	string	length <= 20	✗	✓	default	The date the last successful payment was processed for this subscription ISO 8601 format (UTC) YYYY-MM-DDThh:mm:ssZ

```
{
    "subscriptionId": "234234224",
    "plan": {
        "link": {           {
                "url": "https://api.neteller.com/v1/plans/MONTHLYGREENPLAN",
                "rel": "subscriptionPlan",
                "method": "GET"
        }
    },
    "customer": {
        "link": {
            "url":
"https://api.neteller.com/v1/customers/CUST_0d676b4b-0eb8-4d78-af25-e41ab431e325",
            "rel": "customer",
            "method": "GET"
        }
    },
    "status": "active",
    "startDate": "2014-06-01T00:00:00Z",
    "endDate": "2014-12-31T00:00:00Z",
    "currentPeriodStart": "2014-07-01T00:00:00Z",
    "currentPeriodEnd": "2014-07-31T00:00:00Z",
    "cancelAtPeriodEnd": false,
    "cancelDate":  null,
    "lastCompletedPaymentDate":  "2014-06-01T00:00:00Z"
}
```

tax

Field Name	Field Type	Constraints	Request Field	Response Field	Response Field Scope	Description
taxName	string	length <= 50	✓	✓	default	The name that will display in history and invoice details for this tax item. (ie: VAT, GST, PST...etc)
taxAmount	number	length <= 10	✓	✓	default	The amount of the tax. Amount fields reflect the smallest unit of currency with no decimals. Eg. $25.00 USD should be formatted as 2500.
taxCurrency	string	length = 3	✗	✓	default	The currency the tax is in. see Currencies for complete list

* fields will only be returned if the data is available at time of request. Transfers that are in a pending state will not have fee's calculated until they are accepted.

```
{
    "taxName": "VAT",
    "taxAmount": 50,
    "taxCurrency": "USD"
}
```

token

Field Name	Field Type	Constraints	Request Field	Response Field	Description
tokenType	string	length <= 50	✗	✓	The type of access token that was issued. Currently only 'Bearer' tokens are supported.
expiresIn	number		✗	✓	How much time in seconds until the token expires.
accessToken	string	length <= 200	✗	✓	The access token to pass in the API call to access the protected resource.
refreshToken	string	length <= 200	✗	✓	The token to use to obtain a new access token with the same permissions (scope) previously authorized.

```
{
    "tokenType": "Bearer",
    "expiresIn": 300,
    "accessToken": "28A8J23k2l3k7322027",
    "refreshToken" "28A8J23k2l3k73220273245334"
}
```

transaction

Field Name	Field Type	Constraints	Request Field	Response Field	Description
merchantRefId	string	length <= 50	✓	✓	The external merchant transaction id that uniquely identifies this transaction within the merchant system.
amount	number	length <= 15	✓	✓	The amount of the transaction.. Amount fields reflect the smallest unit of currency with no decimals. Eg. $25.00 USD should be formatted as 2500.
currency	string	length <= 3	✓	✓	The currency of the transaction. see Currencies for complete list
id	number		✗	✓	The transaction id that identifies this transaction within the NETELLER system. The returned id is considered a transaction group id and can be used to identify both sides of a transfer.
createDate	string	length <= 20	✗	✓	ISO 8601 format (UTC) The timestamp when the transaction was requested. YYYY-MM-DDThh:mm:ssZ
updateDate	string	length <= 20	✗	✓	ISO 8601 format (UTC) The timestamp when the transaction was last updated. YYYY-MM-DDThh:mm:ssZ
fees	array of fee		✗	✓	List of fees associated with this transaction.
status	string	length <= 20	✗	✓	accepted pending declined cancelled
errorCode	string	length <= 100	✗	✓	If applicable, the corresponding code for why this transaction was not accepted. see REST API Errors
errorMessage	string	length <= 250	✗	✓	If applicable, the corresponding reason for why this transaction was not accepted. see REST API Errors

```
{
    "merchantRefId": "9876543210",
    "amount": 2500,
    "currency": "USD",
    "id": "288382647322027",
    "createDate": "2013-06-22T14:33:12",
    "updateDate": "2013-06-22T16:21:52",
    "status": "pending",
    "fees": [
        {
            "feeType": "service_fee",
            "feeAmount": 50,
            "feeCurrency": "USD"
        }
    ]
}
```

⚠ The fields in the response will vary depending on the state of the transaction. If the original request failed, only a subset of information will be available.

REST API Errors

HTTP response codes are used to indicate the status of a request.To ease integration and support for developers integrating to our APIs, the error messages that are returned should be verbose enough to correct the problem. The error structure contains a 4–5 digit code as well as a human readable message. Additionally, the application error code will be returned in the header so it can be easily used by an application without parsing the body.

HTTP Status Code Summary

Category Description

1xx: Informational – Communicates transfer protocol–level information.
2xx: Success – Indicates that the client's request was accepted.
3xx: Redirection – Indicates that the client must take some additional action in order to complete the request.
4xx: Client Error – Indicates that the client has made an error with the request.
5xx: Server Error – Indicates that an error occurred on the server side.

Common HTTP Response Status Codes

200 OK – Everything worked as expected.

201 CREATED – The request was successful. NETBANX created a new resource and the response body contains the representation.

202 ACCEPTED – This indicates that the client's request will be handled asynchronously. It tells the client that the request appears valid, but it still may have problems once it is processed.

204 NO CONTENT – This is usually returned in response to a PUT, POST, or DELETE request, when the REST API declines to send back any status message or representation in the body of the response message.

304 NOT MODIFIED – The client's cached version of the representation is still up to date.

400 BAD REQUEST – This often indicates that a required parameter is missing or that a parameter is invalid. This is a generic client-side error status, used when no other 4xx error code is appropriate.

401 UNAUTHORIZED – This indicates that the client tried to operate on a protected resource without providing the proper authorization. They may have provided the wrong credentials or none at all.

402 PAYMENT REQUIRED – The parameters were valid but the request failed.

404 NOT FOUND – The requested resource does not exist.

405 METHOD NOT ALLOWED – The client tried to POST or PUT to a resource that would not accept it.

415 UNSUPPORTED MEDIA TYPE – The request is in a format not supported by the requested resource for the requested method.

429 TOO MANY REQUESTS – The application is sending too many simultaneous requests.

500 INTERNAL SERVER ERROR – An error occurred with an internal server.

502 EXTERNAL SERVER ERROR – We received an invalid response from the upstream gateway in attempting to fulfill the request.

Error Attributes

Element	Type	Description
code	String	This is the error code.
message	String	This is a description of the error.
details	Array	This is a detailed description.
fieldErrors	Array	This is a list of fields that have problems.

Error message example with details

```
{
    "error": {
        "code": "5023",
        "message": "Bad request",
        "details": [
            "Illegal unquoted character ((CTRL-CHAR, code 10)): has to be escaped
using backslash to be included in string value at  [line: 2, column: 61]"
        ]
    }
}
```

Error message example with field errors

```
{
   "error":{
      "code":5068,
      "message":"Field Error(s)",
      "fieldErrors":[
         {
            "field":"recipientEmail",
            "error":"john@hi@you.c.c.c.c is not a valid e-mail address."
         },
         {
            "field":"transaction.amount",
            "error":"A transaction amount is required."
         },
         {
            "field":"transaction.currency",
            "error":"Currency with value [USD] exceeds the maximum size of [3]."
         }
      ],
      "details":[
           "There are invalid or missing fields in your request."
        ]
   }
}
```

Common Errors

These are the error codes common to all subsystems

HTTP Status	Code	Message	Description
400	5001	Invalid currency	The submitted an currency code is invalid or your account does not support this currency.
	5010	Invalid country	The submitted country code is invalid.
	5016	Account not found	The account you provided cannot be found.

	5017	Account disabled	The account you provided is disabled.
	5023	Bad request	The request is not parseable.
	5042	Invalid merchant reference	This error usually indicates that a transaction was attempted with the merchant reference field missing. This value is mandatory and must be included with your request.
	5068	Invalid field	Either you submitted a request that is missing a mandatory field or the value of a field does not match the format expected.
401	5275	Authentication credentials expired	The authentication credentials provided with the request have expired.
	5276	Authentication credentials disabled	The authentication credentials provided with the request provided have been disabled.
	5277	Authentication credentials locked	The authentication credentials provided with the request have been locked due to multiple authentication failures.
	5278	Cannot authenticate	The authentication credentials provided with the request were not accepted for an unknown reason.
	5279	Authentication credentials are invalid	The authentication credentials are invalid.
403	5270	Unauthorized access	The credentials provided with the request do not have permission to access the requested data.
404	5269	Entity not found	The ID(s) specified in the URL do not correspond to the values in the system.
	5273	URI not found	Your client reached our application but we were unable to service your request due to an invalid URL.
406	5271	Unsupported 'Accept' header	You requested a response in the 'Accept' header that is in an unsupported format.
415	5272	Unsupported 'Content-Type'	The 'Content-Type' you specified in request header was submitted in an unsupported format.
429	1200	API call rate exceeded	The API call has been denied as it has exceeded the permissible call rate limit.
500	1000	Internal error	An internal error occurred.
	1002	Database Error	
	1003	Decryption Error	
	1007	Gateway Error	

These errors are specific to the NETELLER REST API.

HTTP Status	Code	Message	Description
401	20000	Unsupported API request	You are not registered to use this API
401	20001	Account closed	The account is closed
400	20002	Merchant account inactive	Your merchant account is currently inactive.

400	20003	Invalid merchant account currency	Your NETELLER Merchant Account does not support this currency.
400	20005	Duplicate transaction reference	The provided transaction reference has already been used for another transaction. You must provide a unique value for each request.
400	20006	Invalid amount format	Amount must be numeric and cannot contain decimals or any alpha characters.
400	20007	Invalid accountId or email	The specified accountId or email is invalid.
400	20008	Invalid verification code	The supplied verification code was not valid for this request.
400	20010	Merchant blocked geolocation	Member is residing in a blocked country/state/region (merchant has chosen to block members from transacting in that location)
400	20011	NETELLER blocked geolocation	Member is residing in a NETELLER blocked country/state/region.
400	20015	Member is not entitled.	Member is not entitled to receive this type of transaction.
400	20016	Merchant is not entitled.	Merchant is not entitled to initiate this type of transaction.
402	20020	Insufficient balance	Insufficient balance to complete the transaction.
400	20021	Below member min transfer amount	The specified amount is below defined minimum transfer limits.
400	20022	Above max transfer out amount	The specified amount is above defined maximum transfer out limits.
400	20025	Below merchant min transfer amount	The specified amount is below your defined minimum transfer limits.
400	20026	Above max transfer in amount	The specified amount is above defined maximum transfer in limits.
400	20030	Below min transaction limit	The specified amount is too low. You must specify an amount greater than or equal to 1 unit of currency.
400	20031	Above max transaction limit	The specified amount is too high. You must specify an amount within your transactional limit.
400	20035	Above max account limit	The transaction exceeds allowed account limits.
400	20040	Expired	The transaction was not completed in the permitted time and has expired.
400	20045	Merchant declined	Transaction was manually cancelled or declined by the merchant.
400	20050	Legacy transaction details not available from this API, use NetCheck	The transaction you are trying to retrieve was submitted using the legacy NetDirect or Instant Payout API. Please use NetCheck to retrieve the associated error detail for the transaction.
400	20100	Plan already exists	The requested planId has already been used. Please specify a unique planId with your request.
400	20101	Cannot delete a subscribed to plan	You cannot delete a subscription plan that has already been subscribed to. If you wish to prevent further enrollment in this plan then you must cancel the plan .
400	20102	Invalid term length	The term length for this subscription plan is invalid. Please ensure the interval, interval type and interval count combination is less than 5 years.
400	20200	Existing member email	The supplied email is already registered with NETELLER.
401	20900	Request not authorized	Your request was not authorized as the API requires member authorization. Please use the authorization_grant flow to obtain member permission for the required scope.

Currencies

- AUD – Australian Dollar
- GBP – British Pound
- BGN – Bulgarian Lev
- CAD – Canadian Dollar
- DKK – Danish Krone
- EUR – Euro
- HUF – Hungarian Forint
- INR – Indian Rupee
- JPY – Japanese Yen
- LTL – Lithuanian Litus
- MYR - Malaysian Ringgit
- MAD- Moroccan Dirham
- MXN – Mexican Peso
- NOK – Norwegian Kroner
- PLN – Polish Zloty
- RON – Romanian New Leu
- RUB - Russian Ruble
- SGD - Singapore Dollar
- SEK – Swedish Krona
- CHF - Swiss Franc
- TND - Tunisia Dinar
- USD – United States Dollar

Languages

- da_DK - Danish
- de_DE - German
- el_GR - Greek
- en_US - English
- es_ES - Spanish
- fr_FR - French
- it_IT - Italian
- ja_JP - Japanese
- ko_KR - Korean
- no_NO - Norwegian
- pl_PL - Polish
- pt_PT - Portuguese
- ru_RU - Russian
- sv_SE - Swedish
- tr_TR - Turkish

Test Member Accounts

⚠ The following member accounts are available in the sandbox environment for testing purposes. These accounts should never be used in production and are only intended for use in testing with the test.api.neteller.com endpoint. The password supplied here can be used for testing the Authorization flows for a particular account.

Currency	*Account ID*	*Email Address*	*Secure ID*	Password
AUD	451823760529	netellertest_AUD@neteller.com	521652	NTt3st1!
BGN	450424149137	netellertest_BGN@neteller.com	354380	NTt3st1!
CAD	455781454840	netellertest_CAD@neteller.com	755608	NTt3st1!
CHF	452324249609	netellertest_CHF@neteller.com	372993	NTt3st1!
DKK	459734233011	netellertest_DKK@neteller.com	856751	NTt3st1!
EUR	453501020503	netellertest_EUR@neteller.com	908379	NTt3st1!
GBP	458591047553	netellertest_GBP@neteller.com	411392	NTt3st1!
HUF	450824149649	netellertest_HUF@neteller.com	363552	NTt3st1!
INR	450824016049	netellertest_INR@neteller.com	332880	NTt3st1!
JPY	452604251512	netellertest_JPY@neteller.com	490055	NTt3st1!
LTL	451623758513	netellertest_LTL@neteller.com	750338	NTt3st1!
LVL	450224147121	netellertest_LVL@neteller.com	728615	NTt3st1!
MAD	453123727913	netellertest_MAD@neteller.com	796289	NTt3st1!
MXN	456444237546	netellertest_MXN@neteller.com	878408	NTt3st1!
MYR	452724116521	netellertest_MYR@neteller.com	108145	NTt3st1!
NOK	455394172769	netellertest_NOK@neteller.com	418852	NTt3st1!
PLN	451823629489	netellertest_PLN@neteller.com	654091	NTt3st1!
RON	450424018097	netellertest_RON@neteller.com	860647	NTt3st1!
RUB	455121038904	netellertest_RUB@neteller.com	888470	NTt3st1!
SGD	451523741861	netellertest_SGD@neteller.com	316938	NTt3st1!
SEK	453313818311	netellertest_SEK@neteller.com	173419	NTt3st1!
TND	453523858985	netellertest_TND@neteller.com	588931	NTt3st1!
USD	454651018446	netellertest_USD@neteller.com	270955	NTt3st1!

Glossary

[Access Token] [Authorization Code] [Authorization Grant] [Invoice] [Refresh Token] [REST (REpresentational State Transfer)] [Transaction]

Access Token

Access tokens are credentials used to access protected resources. An access token is a string representing an authorization issued to the client. The string is usually opaque to the client. Tokens represent specific scopes and durations of access, granted by the resource owner, and enforced by the resource server and authorization server.

The token may denote an identifier used to retrieve the authorization information, or self-contain the authorization information in a verifiable manner (i.e. a token string consisting of some data and a signature).

The access token provides an abstraction layer, replacing different authorization constructs (e.g. username and password) with a single token understood by the resource server. This abstraction enables issuing access tokens more restrictive than the authorization grant used to obtain them, as well as removing the resource server's need to understand a wide range of authentication methods.

Authorization Code

The authorization code is obtained by using an authorization server as an intermediary between the client and resource owner. Instead of requesting authorization directly from the resource owner, the client directs the resource owner to an authorization server (via its user-agent as defined in [RFC2616]), which in turn directs the resource owner back to the client with the authorization code.

Before directing the resource owner back to the client with the authorization code, the authorization server authenticates the resource owner and obtains authorization. Because the resource owner only authenticates with the authorization server, the resource owner's credentials are never shared with the client.

The authorization code provides a few important security benefits such as the ability to authenticate the client and issuing the access token directly to the client without potentially exposing it to others, including the resource owner.

Authorization Grant

An authorization grant is a general term used to describe the intermediate credentials representing the resource owner authorization (to access its protected resources), and serves as an abstraction layer. An authorization grant is used by the client to obtain an access token.

The OAUTH specification defines four grant types: authorization code, implicit, resource owner password credentials, and client credentials, as well as an extensibility mechanism for defining additional types.

Invoice

An invoice is a collection of purchase items that rolls up to produce a single monetary amount, or amount due, that needs to be collected from the buyer.   The invoice, when paid, will have a related **transaction** that details the collection of payment.

Refresh Token

Refresh tokens are credentials used to obtain access tokens. Refresh tokens are issued to the client by the authorization server and are used to obtain a new access token when the current access token becomes invalid or expires.

A refresh token is a string representing the authorization granted to the client by the resource owner. The string is usually opaque to the client. The token denotes an identifier used to retrieve the authorization information. Unlike access tokens, refresh tokens are intended for use only with authorization servers and are never sent to resource servers.

REST (REpresentational State Transfer)

Representational state transfer (**REST**) is a software architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed hypermedia system. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements.

Transaction

A transaction represents a financial settlement between two parties. The buyer and seller are separate entities, often involving the exchange of items of value, such as information, goods, services, and money.