



NEOCURRENCY®

---

# API SPECIFICATIONS

**VERSION 1.8.1**

UPDATED 3.15.2022

**CONTENTS**

SUMMARY .....	5
STANDARD USER FLOW .....	5
DIFFERENCES BETWEEN SANDBOX AND PRODUCTION .....	5
DISCUSSION: EXPIRATION DATES .....	5
DISCUSSION: CREATEORDERNOW vs CREATECODES .....	6
EXAMPLE: CREATING YOUR FIRST ORDER .....	7
EXAMPLE: CREATING YOUR FIRST ORDER AND SENDING YOUR FIRST EMAIL .....	8
AUTHENTICATE .....	10
Request .....	10
Response .....	11
1. GET USER .....	13
Request .....	13
Response .....	13
Response example: .....	14
2. GET BRANDS .....	15
Request .....	15
Response .....	15
Response example: .....	16
3. GET FUNDS .....	18
Request .....	18
Response .....	18
Response example: .....	19
4. GET CURRENT BALANCE .....	20
Request .....	20
Response .....	20
Response example: .....	21
5. GET PROJECTS .....	22
Request .....	22
Response .....	22

Response example: .....	23
6. CREATE CAMPAIGN .....	24
Request .....	24
Response.....	25
Response example: .....	26
7. CREATE CODES .....	27
Request .....	27
Response.....	29
Response example: .....	29
8. GET CAMPAIGNS .....	31
Request .....	31
Response.....	31
Response example: .....	31
9. ACTIVATE CAMPAIGN .....	33
Request .....	33
Response.....	33
Response example: .....	34
10. ORDER NOW .....	35
Request .....	35
Response.....	35
Response example: .....	36
11. CREATE ORDER NOW .....	37
Request .....	37
Response.....	39
Response example: .....	39
12. ORDER NOW ADD EMAILS .....	41
Request .....	41
Response.....	42
Response example: .....	43
13. ORDER NOW SEND EMAILS .....	44
Request .....	44
Response.....	45
Response example: .....	45

14. ORDER NOW LOAD PHYSICAL CARDS .....	46
Request .....	46
Response.....	47
Response example: .....	48
15. ORDER NOW REDEEM PHYSICAL CARDS .....	49
Request .....	49
Response.....	50
Response example: .....	50
Dynamic Fields .....	51
Client_ID Discussion.....	52
Cross Environment Testing .....	52
Conventions .....	53
Status Codes .....	53

# SUMMARY

## STANDARD USER FLOW

The Production Environment base URL is: <https://redeem.yourdigitalreward.com>

The standard user flow for using the Production Environment is to first Authenticate using the Authentication endpoint. After which to create a new campaign using the Create Campaign endpoint. After a new Campaign has been created, the user should activate the campaign using the Activate Campaign endpoint before being able to create codes using the Create Codes endpoint. After a campaign has been created and activated additional codes can be added at a later time using the Create Codes endpoint.

Before starting with the Production Environment, we highly recommend using our Sandbox Environment which mimics the functionality of the Production Environment and allows for easy and safe testing of endpoints and user flow. Please ask about access to our Sandbox Environment.

## DIFFERENCES BETWEEN SANDBOX AND PRODUCTION

There are some differences between the Production Environment and the Sandbox Environment. The two major differences between the Sandbox Environment and the Production Environment are as follows.

Firstly, when calling Sandbox Environment methods there is a /sandbox/ prefix attached to the method declarations. For example, the Sandbox Environment brands method is called via `api/sandbox/brands` while the Production Environment brands method is called via `api/brands`.

Secondly, the code redemption in the Sandbox Environment is simulated. So when attempting to redeem codes via <https://redeem.yourdigitalreward.com/sandbox/activate-code/{code}> user will receive a message that the code has been redeemed but that is the extent of the code redemption in the Sandbox Environment.

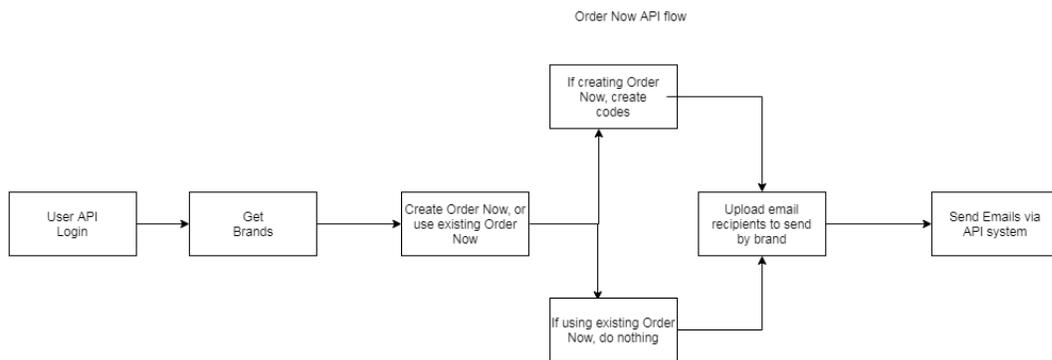
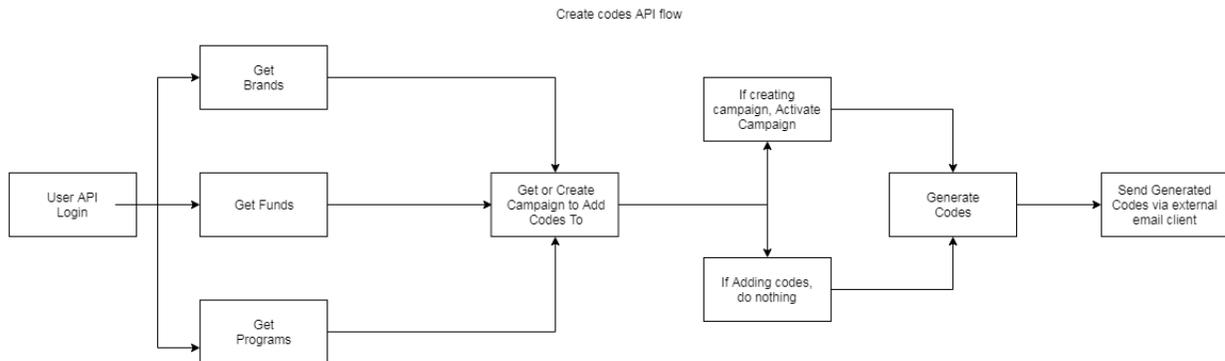
## DISCUSSION: EXPIRATION DATES

The NeoCurrency System Campaigns have an end date associated with each campaign created in the NeoCurrency System. The end date of a campaign is the date after which all codes generated for this campaign become expired or invalid. This is done so that the system is

more flexible in handling both limited duration and unlimited duration campaigns. In the case of a client wanting to run a campaign only for a limited amount of time, we suggest entering a campaign end date when the client wants the system generated codes to expire. If on the other hand the client would like to have an unlimited length campaign, we recommend setting a campaign end date five or ten years in the future. As a recap, NeoCurrency system generated codes are only valid while the campaign which has generated those codes has not passed its end date.

## DISCUSSION: CREATEORDERNOW vs CREATECODES

In the NeoCurrency System, there is a distinction between Order Now orders and creating additional codes for an already existing campaign. The main distinction is the Order Now orders allow the user to upload and send emails for the codes created with Order Now orders while this is not allowed for create/codes standard campaigns. If user wishes to use the email functionality, user should use the createordernow order endpoint and flow. Below is a flow diagram comparing the ordering process between using Order Now and using the standard code creation.



## EXAMPLE: CREATING YOUR FIRST ORDER

In this example, we will go over the needed endpoints to create your first order.

The overall flow is as follows:

POST /api/sandbox/createordernow

Below is an example of the requests and responses from this transaction flow:

POST /api/sandbox/createordernow

Request:

```
{
  "custom1": "this is my first order", ← This is a custom field that can be used for things like
  custom data which will not be shown to the end customer.
  "brands": [ {
    "id": 623, ← This is the brand you are looking to order
    "denomination": 0.05,
    "quantity": 1
  } ]
}
```

Response:

```
{
  "success": {
    "order_id": 6208,
    "custom1": "this is my first order",
    "data": [
      {
        "campaign_brand_id": 10031,
        "denomination": "0.05",
        "currency": "USD",
        "codes": [
          "c613e814692e94906be3d8f592551894e5bd1f85"
        ],
        "uuids": {
          "c613e814692e94906be3d8f592551894e5bd1f85": "3520f5a13757d26d717ac35c5c1f0
          d7f202103111028291"
        }
      }
    ]
  }
}
```

## EXAMPLE: CREATING YOUR FIRST ORDER AND SENDING YOUR FIRST EMAIL

In this example, we will go over the needed endpoints to create your first order and send your first email.

The overall flow is as follows:

POST /api/sandbox/createordernow

POST /api/sandbox/addemails

POST /api/sandbox/sendemails

Below is an example of the requests and responses from this transaction flow:

POST /api/sandbox/createordernow

Request:

```
{
  "custom1": "this is my first order", ← This is a custom field that can be used for things like
  custom data which will not be shown to the end customer.
  "brands": [ {
    "id": 623, ← This is the brand you are looking to order
    "denomination": 0.05,
    "quantity": 1
  } ]
}
```

Response:

```
{
  "success": {
    "order_id": 6208, ← This will be needed when adding and sending emails
    "custom1": "this is my first order",
    "data": [
      {
        "campaign_brand_id": 10031, ← This will be needed when adding and sending
        emails
        "denomination": "0.05",
        "currency": "USD",
        "codes": [
          "c613e814692e94906be3d8f592551894e5bd1f85"
        ],
        "uuids": {
```

```
      "c613e814692e94906be3d8f592551894e5bd1f85": "3520f5a13757d26d717ac35c5c1f0
d7f202103111028291"
    }
  }
]
}
}
```

POST /api/sandbox/addemails

Request:

```
{
  "order_id": 6208,
  "campaign_brand_id": 10031,
  "emails": [
    {
      "email": "johndoe@test.com",
      "first_name": "John",
      "last_name": "Doe",
      "subject": " This is a test"
    }
  ]
}
```

Response:

```
{
  "success": "You have successfully added one email!"
}
```

POST api/sandbox/sendemails

Request:

```
{
  "order_id": 6208,
  "campaign_brand_id": 10031
}
```

Response:

```
{
  "send": 1
}
```

## HEADERS

In order to work properly, the NeoCurrency REST API needs the header “Content-Type” with the value “application/json” to be included in each request.

## AUTHENTICATION

### AUTHENTICATE

The NeoCurrency REST API uses OAuth2 authentication with Password Grant. In order to successfully authenticate and send requests to the REST API, the user sets the request Authorization header to Bearer and transmits the access\_token received as part of a successful authentication response.

#### Request

Method	URL
POST	api/get-token

Headers	Value
Content-Type	application/json

Type	Params	Values
POST	client_id	string
POST	client_secret	string
POST	email	string
POST	password	string

**client\_id**

Client ID provided by the NeoCurrency REST API

**client\_secret**

Companion to the client\_id that is provided by the NeoCurrency REST API

**email**

The email of the user that wishes to authenticate with the API.

**password**

The password of the user that wishes to authenticate with the API

**Request example:**

```
{  
  "client_id": "JOI8EX4J3K8706L3HQ0KIL3H",  
  "client_secret": "Q0JBQEHWT1J899BG3UM83GNMWG53OW4OQB5GMCNYH7DYDN",  
  "email": "api@test.com",  
  "password": "123456"  
}
```

**Response**

Status	Response
200	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- access_token: The access token. This expires in 1 day</li></ul>
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "Unauthenticated."</li></ul>
500	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "500 Internal Server Error."</li></ul>

**Response example:**

```
{  
  "access_token": "alphanumeric code here"  
}
```

## 1. GET USER

Get information for the authenticated user

### Request

Method	URL
GET	api/user

Type	Params	Values
Headers	Authorization: Bearer	access_token

### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

### Response

Status	Response
200	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- id: Integer</li><li>- name: String</li><li>- email: String</li><li>- role_id: Integer</li><li>- created_at: Datetime</li><li>- updated_at: Datetime</li></ul>

	<ul style="list-style-type: none"><li>- active: Integer 1-Active, 0-Not Active</li></ul>
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "Unauthenticated."</li></ul>
500	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "500 Internal Server Error."</li></ul>

### Response example:

```
{
  "id": 3,
  "name": "Api test",
  "email": "api@neocurrency.com",
  "role_id": "5",
  "created_at": "2017-08-13 19:46:45",
  "updated_at": "2017-12-07 13:46:24",
  "active": "1"
}
```

## 2. GET BRANDS

Getting all the brands that the authenticated user has access to.

### Request

Method	URL
GET	api/brands

Type	Params	Values
Header	Authorization: Bearer	access_token

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

### Response

Status	Response
200	The rest API will respond with an array of JSON objects, each of them containing the following properties: <ul style="list-style-type: none"> <li>- id: Integer</li> <li>- name: String</li> <li>- active: Integer 1-Active, 0-Not Active</li> <li>- created_at: Datetime</li> <li>- updated_at: Datetime</li> <li>- user_id: Integer - ID of authenticated user</li> </ul>
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: error: "Unauthenticated."
500	The rest API will respond with a JSON object containing the following properties:

	- error: "500 Internal Server Error."
--	---------------------------------------

## Response example:

```
[
  {
    "id": 1,
    "name": "Test V1",
    "program_id": 46595,
    "catalog_id": null,
    "sku": "UVC-V-V6M",
    "value": 0,
    "min": 1,
    "max": 4900,
    "logo":
      "https://redeem.yourdigitalreward.com/storage/upload/eJfbzICpvtT0w4oXkng2DuGsjspcAzUcr
      O6dIASP.png",
    "description": "The Virtual Visa&#174; Reward can be used to buy what you want,
    when you want it. Because it is so flexible and convenient, the Visa Virtual Reward makes it
    easy to treat yourself to something special or to help cover your everyday expenses. The
    decision is yours.",
    "terms": "This virtual reward is issued by Sutton Bank, member FDIC, pursuant to license
    by Visa U.S.A. Inc.",
    "reward_gateways_id": 2,
    "currency_id": 1,
    "created_at": null,
    "updated_at": null,
    "active": 1
  },
  {
    "id": 4,
    "name": "eBay-5-200",
    "program_id": 1185,
    "catalog_id": 1,
    "sku": "EBY-D-V-00",
    "value": 0,
    "min": 5,
    "max": 200,
    "logo": null,
  }
]
```

```
"description": "",  
"terms": null,  
"reward_gateways_id": 3,  
"currency_id": 1,  
"created_at": null,  
"updated_at": null,  
"active": 1  
}  
]
```

### 3. GET FUNDS

Getting all the funds that the authenticated user has access to.

#### Request

Method	URL
GET	api/funds

Type	Params	Values
Header	Authorization: Bearer	access_token

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

#### Response

Status	Response
200	The rest API will respond with an array of JSON objects
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: error: "Unauthenticated."
500	The rest API will respond with a JSON object containing the following properties: - error: "500 Internal Server Error."

### Response example:

```
[
  {
    "id": 1,
    "client_id": 3,
    "currency_id": "USD",
    "value": 1000
  },
  {
    "id": 9,
    "client_id": 3,
    "currency_id": "EUR",
    "value": 500
  }
]
```

## 4. GET CURRENT BALANCE

Getting the total of the current funds and any balance distributed over any Projects the user has access to. This endpoint shows the current total balance that the client has access to, to create and redeem codes with. This endpoint will provided more up to date information than the /funds endpoint which only shows the current fund balance for the user.

### Request

Method	URL
GET	api/currentbalances

Type	Params	Values
Header	Authorization: Bearer	access_token

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

### Response

Status	Response
200	The rest API will respond with an array of JSON objects
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: - error: "Unauthenticated."
500	The rest API will respond with a JSON object containing the following properties: - error: "500 Internal Server Error."

### Response example:

```
[
  {
    "id": 1,
    "client_id": 3,
    "currency_id": "USD",
    "value": 1469
  },
  {
    "id": 5,
    "client_id": 3,
    "currency_id": "EUR",
    "value": 100
  }
]
```

## 5. GET PROJECTS

Getting all the projects that the authenticated user has access to.

### Request

Method	URL
GET	api/projects

Type	Params	Values
Header	Authorization: Bearer	access_token

### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

### Response

Status	Response
200	The rest API will respond with an array of JSON objects
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: - error: "Unauthenticated."
500	The rest API will respond with a JSON object containing the following properties: - error: "500 Internal Server Error."

## Response example:

```
[
  {
    "id": 1,
    "p_group_id": 3,
    "p_name": "Summer Campaigns",
    "p_description": "Summer Campaigns Project",
    "p_active": 1,
    "created_at": "2017-09-20 14:52:15",
    "updated_at": "2017-09-30 12:14:15"
  },
  {
    "id": 2,
    "p_group_id": 3,
    "p_name": "Winter Campaigns",
    "p_description": "Winter Campaign Project",
    "p_active": 1,
    "created_at": "2017-09-20 20:12:23",
    "updated_at": "2017-12-07 10:02:37"
  }
]
```

## 6. CREATE CAMPAIGN

Method called when creating a campaign. Once this method is successfully executed a campaign with pending status will be created for the user.

### Request

Method	URL
POST	api/campaigns/create

Type	Params	Values
Header	Authorization: Bearer	access_token

Type	Params	Values
POST	name	string
POST	start_date	Date: datetime format(MM/DD/YYYY HH:ii)
POST	end_date	Date: datetime format (MM/DD/YYYY HH:ii)
POST	timezone	string
POST	project_id	integer
POST	fund_id	integer

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

#### name

The name of the campaign, there are no restrictions on name and multiple campaigns can have the same name

#### start\_date

The date, down to the hour and minute combination of when the campaign will start

#### end\_date

The date, down to the hour and minute combination of when the campaign will end.

**timezone**

The timezone for the start and end date of the campaign. We use PHP timezones (<http://php.net/manual/en/timezones.php>) as these automatically take into account daylight savings time. The possible values for timezone are: “Eastern Standard Time”, “Central Standard Time”, “Mountain Standard Time”, “Pacific Standard Time”, “Hawaii”.

**project\_id**

The ID of the Project attached to this Campaign. See get all projects by user.

**fund\_id**

The ID of the Fund attached to this Campaign. See get all funds by user.

**Request example:**

```
{
  "name": "api campaign 1",
  "start_date": "12/27/2017 13:00",
  "end_date": "12/31/2017 13:00",
  "timezone": "Eastern Standard Time",
  "project_id": 1,
  "fund_id": 1
}
```

**Response**

Status	Response
200	The rest API will respond with a JSON objects containing the following properties: - success
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
422	The rest API will respond with an array of JSON objects, each of them, containing the following properties: - name_of_the_problem_field: array of validation error messages
401	The rest API will respond with a JSON object containing the following properties: - error: “Unauthenticated.”

500	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "500 Internal Server Error."</li></ul>
-----	---

### Response example:

```
{
  "success": [
    {
      "create_campaign": "You have successfully created a campaign with ID: 45!"
    }
  ]
}

{
  "errors": [
    {
      "timezone": "Time Zone is not correct!"
    },
    {
      "start_date": "A start date must be before end date!"
    }
  ]
}
```

## 7. CREATE CODES

This method is used to get the generated codes for the campaign with the specified campaign ID. The endpoint returns both the codes in the codes array as well as UUID pair which can be used to distinguish individual codes on client end. The UUID has no monetary value and can be used to refer to the reward code without having to show it directly.

### Request

Method	URL
POST	api/codes/create

Type	Params	Values
Header	Authorization: Bearer	access_token

Type	Params	Values
POST	campaign_id	integer
POST	brands	array
	id	integer
	denomination	float
	quantity	integer
	secret (optional)	boolean (true/false)
	secret_activation_time (optional)	double
	client_id (optional)	integer
	expiration_days (optional)	double

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

#### campaign\_id

The ID of the campaign for which these codes are being generated for.

**brands**

The array of brands used in the campaign for which codes need to be generated. For each brand an `id` for the brand, a `denomination` and `quantity` are required.

**secret (optional)**

Secret is a true or false value which determines if the code generated will have an additional 2 factor authentication code when redeemed. This is an optional parameter.

**secret\_activation\_time (optional)**

If the brand has `secret` set to true, then the `secret_activation_time` is the time (in hours) of when the redeemed code will be activated after the end user has requested a redemption of the code. An example where a code is created today with a secret code and a `secret_activation_time` of 24 (hours) means the following. If the code is created today and accessed by the end client tomorrow, the 24 hours will begin counting down from tomorrow (the time end user activated the reward). After the 24 hours wait time, the end client will be able to claim their reward. The minimum time is 0.25 of an hour or 15 minutes while there is no maximum.

**expiration\_days (optional)**

This optional parameter controls how many days after redemption a code for a given brand will expire. For example if a brand is ordered with an `expiration_days` value of 1, then the code will expire 1 day after being redeemed. Expiration in this context means that the end client will have a limited amount of time after redeeming the code to use said code.

**client\_id**

This optional parameter is something which is required to place orders for some brands. The value here will be provided at Account Setup. In most cases the value will start from “1” and increase as more `client_id`'s are added for a specific brand.

**Request example:**

```
{
  "campaign_id":9,
  "brands":[
    {
      "id" : 999,
      "denomination": 1.01,
      "quantity": 2
      "secret" : true,
      "secret_activation_time": 0.25,
      "client_id": 29,
      "expiration_days": 30
    }
  ]
}
```

```

    }
  ]
}

```

## Response

Status	Response
200	The rest API will respond with a JSON objects containing the following properties: - success
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: - error: "Unauthenticated."
500	The rest API will respond with a JSON object containing the following properties: - error: "500 Internal Server Error."

## Response example:

```

{
  "success": [
    {
      "denomination": "0.11",
      "currency": "USD",
      "codes": [
        "80cbdeae755c97e61e47918addb1a268f36aca6d",
        "df7a458d8314f8bcac2803d47647bcd711e8a43d"
      ],
      "uuids": {
        "80cbdeae755c97e61e47918addb1a268f36aca6d": "462e14a70135493e89b39fa6cd1364c3202102051036531",
        "df7a458d8314f8bcac2803d47647bcd711e8a43d": "83cab43947d05f43267e3fb47c337834202102051036532"
      },
      "codeSecretOffset": {
        "80cbdeae755c97e61e47918addb1a268f36aca6d": 0.25,
        "80cbdeae755c97e61e47918addb1a268f36aca6d": 0.25
      },
    }
  ]
}

```

```
    "codeSecret": {  
      "80cbdeae755c97e61e47918addb1a268f36aca6d ": "REXWJ6n0fJ",  
      "df7a458d8314f8bcac2803d47647bcd711e8a43d": "AER1JbO6f9"  
    },  
    "order_id": 6149  
  }  
]  
}
```

## 8. GET CAMPAIGNS

Gets all the Campaigns that the authenticated user has access to.

### Request

Method	URL
GET	api/campaigns

Type	Params	Values
Header	Authorization: Bearer	access_token

### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

### Response

Status	Response
200	The rest API will respond with an array of JSON objects
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "Unauthenticated."</li></ul>
500	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "500 Internal Server Error."</li></ul>

### Response example:

```
[  
  {
```



```
"id": 34,  
"name": "api campaign 1",  
"description": null,  
"created_at": {  
  "date": "2017-12-27 12:58:53.000000",  
  "timezone_type": 3,  
  "timezone": "UTC"  
},  
"status": "pending",  
"start_date": "2017-12-27 13:00:00",  
"end_date": "2017-12-31 13:00:00",  
"timezone": "Eastern Standard Time",  
"project_id": 1,  
"fund_id": 1  
},  
{  
  "id": 35,  
  "name": "api campaign 1",  
  "description": null,  
  "created_at": {  
    "date": "2017-12-27 12:59:11.000000",  
    "timezone_type": 3,  
    "timezone": "UTC"  
  },  
  "status": "pending",  
  "start_date": "2017-12-27 13:00:00",  
  "end_date": "2017-12-11 13:00:00",  
  "timezone": "Eastern Standard Time",  
  "project_id": 1,  
  "fund_id": 1  
}  
]
```

## 9. ACTIVATE CAMPAIGN

Activate a campaign so that codes for this campaign can be generated.

### Request

Method	URL
POST	api/campaign/active

Type	Params	Values
Header	Authorization: Bearer	access_token

Type	Params	Values
POST	campaign_id	integer

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

#### campaign\_id

campaign\_id is the ID of the campaign that needs to be activated. A campaign needs to be activated before any codes can be generated for this campaign.

### Request example:

```
{  
  "campaign_id":46  
}
```

### Response

Status	Response
200	The rest API will respond with a JSON objects containing the following properties: <ul style="list-style-type: none"><li>- success</li></ul>

308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "Unauthenticated."</li></ul>
500	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "500 Internal Server Error."</li></ul>

### Response example:

```
{
  "success": {
    "id": 46,
    "name": "SANDBOX api campaign 1",
    "description": null,
    "expiration_days": 331,
    "created_at": "2018-01-04 15:18:09",
    "updated_at": "2018-01-05 10:36:18",
    "status": "active",
    "start_date": "2018-01-04 13:00:00",
    "end_date": "2018-12-01 13:00:00",
    "timezone": "Eastern Standard Time",
    "project_id": 1,
    "fund_id": 1
  }
}
```

## 10. ORDER NOW

Lists all the orders from Order Now functionality. The Order Now type orders are those which are eligible to be used with the email functionality available in the API. To create an Order Now type order, use the `/api/createordernow` endpoint.

### Request

Method	URL
GET	api/ordernow

Type	Params	Values
Header	Authorization: Bearer	access_token

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

### Response

Status	Response
200	The rest API will respond with a JSON objects containing the following properties: <ul style="list-style-type: none"> <li>- success</li> </ul>
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"> <li>- error: "Unauthenticated."</li> </ul>
500	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"> <li>- error: "500 Internal Server Error."</li> </ul>

## Response example:

```
[
  {
    "id": 1,
    "total_cost": "$5.50",
    "payment_method": "Draw-Down Account",
    "purchase_order_number": "66",
    "brands": [
      {
        "brand_id": 2,
        "brand_name": "Amazon",
        "amount": "$1.00",
        "total_rewards": "2",
        "rewards_available_to_send": 2,
        "rewards_available_to_add_email": 2
      },
      {
        "brand_id": 23,
        "brand_name": "CVS",
        "amount": "$10.00",
        "total_rewards": "1",
        "rewards_available_to_send": 1,
        "rewards_available_to_add_email": 1
      }
    ]
  }
]
```

## 11. CREATE ORDER NOW

This method is used to create an order which can use the email functionality enabled in the API.

### Request

Method	URL
POST	api/createordernow

Type	Params	Values
Header	Authorization: Bearer	access_token

Type	Params	Values
POST	purchase_order_number	string
POST	custom1	string
POST	brands: id: denomination: quantity: secret (optional): secret_activation_time (optional): expiration_days: (optional)	Array integer double integer boolean (true/false) double double
POST	client_id:	integer (optional)

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

#### purchase\_order\_number

purchase\_order\_number is an optional integer value to pass when creating an order now which can help the user track the different order now orders in the system.

#### custom1

custom1 is an optional string field where user can enter any data they wish to pass to the

NeoCurrency system about an order now order. This information can help the user track a specific order now order in the system.

### **brands**

The array of brands used in the campaign for which codes need to be generated. For each brand an `id` for the brand, a `denomination` and `quantity` are required. The `id` is from the `/brands` endpoint, the possible values for `denomination` and `quantity` are governed by the brand `id` and what is available for this brand via `/brands` endpoint.

### **secret (optional)**

Secret is a true or false value which determines if the code generated will have an additional 2 factor authentication code when redeemed. This is an optional parameter.

### **secret\_activation\_time (optional)**

If the brand has `secret` set to true, then the `secret_activation_time` is the time (in hours) of when the redeemed code will be activated after the end user has requested a redemption of the code. An example where a code is created today with a secret code and a `secret_activation_time` of 24 (hours) means the following. If the code is created today and accessed by the end client tomorrow, the 24 hours will begin counting down from tomorrow (the time end user activated the reward). After the 24 hours wait time, the end client will be able to claim their reward. The minimum time is 0.25 of an hour or 15 minutes while there is no maximum.

### **expiration\_days (optional)**

This optional parameter controls how many days after redemption a code for a given brand will expire. For example if a brand is ordered with an `expiration_days` value of 1, then the code will expire 1 day after being redeemed. Expiration in this context means that the end client will have a limited amount of time after redeeming the code to use said code.

### **client\_id**

This optional parameter is something which is required to place order now orders for some brands. The value here will be provided at Account Setup. In most cases the value will start from "1" and increase as more `client_id`'s are added for a specific brand.

## **Request example:**

```
{
  "purchase_order_number": "66",
  "custom1": "something",
  "brands": [
    {
      "id": 2,
      "denomination": 1.00,
      "quantity": 2,
      "expiration_days": 20.5,
    },
    {
```

```

    "id": 23,
    "denomination": 10.00,
    "quantity": 1
  }
]
}

```

### Request example 2:

```

{
  "brands" : [ {
    "id" : 623,
    "denomination" : 0.05,
    "quantity" : 1,
    "secret" : true,
    "secret_activation_time": 0.5
  } ]
}

```

### Response

Status	Response
200	The rest API will respond with a JSON objects containing the following properties: <ul style="list-style-type: none"> <li>- success</li> </ul>
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"> <li>- error: "Unauthenticated."</li> </ul>
500	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"> <li>- error: "500 Internal Server Error."</li> </ul>

### Response example:

```

{
  "success": {
    "order_id": 6208,
    "custom1": null,
    "data": [

```

```
{
  "campaign_brand_id": 10031,
  "denomination": "0.05",
  "currency": "USD",
  "codes": [
    "c613e814692e94906be3d8f592551894e5bd1f85"
  ],
  "uuids": {
    "c613e814692e94906be3d8f592551894e5bd1f85": "3520f5a13757d26d717ac35c5
c1f0d7f202103111028291"
  },
  "codeSecretOffset": {
    "c613e814692e94906be3d8f592551894e5bd1f85": 0.5
  },
  "codeSecret": {
    "c613e814692e94906be3d8f592551894e5bd1f85": "REXWJ6n0fJ"
  }
}
]
```

## 12. ORDER NOW ADD EMAILS

This method is used to create an order which can use the email functionality enabled in the API.

### Request

Method	URL
POST	api/addemails

Type	Params	Values
Header	Authorization: Bearer	access_token

Type	Params	Values
POST	order_id	integer
POST	campaign_brand_id	integer
POST	emails	array with first name, last name, email, subject, custom message. Subject and Custom Message are optional

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

#### order\_id

order\_id is the ID of the Order Now order for which user wishes to send emails for. To get all eligible Order Now order\_ids use /api/ordernow. A single Order Now order maybe have multiple brands attached, so emails are loaded and sent by brand.

#### campaign\_brand\_id

campaign\_brand\_id is the campaign brand ID received from the success message of the createordernow endpoint. The campaign brand ID is an amalgamation which signifies a particular brand and denomination pair. These integer values are unique and non-repeating.

### emails

emails is an array which holds first name, last name, email, subject and custom message for each code. The subject and custom message parameters are optional. If they are used, they will overwrite the email subject and default message to user. For custom message, we only allow text messages and do not, in this version, allow html or links in the message. For an example of usage, please see the Request example below.

### Request example:

```
{
  "order_id": 1,
  "campaign_brand_id": 2,
  "emails": [
    {
      "email": "john@test.com",
      "first_name": "John",
      "last_name": "Doe",
      "subject": "Congratulations Winner",
      "message": "This will be some string text message that will be displayed."
    }
  ]
}
```

### Response

Status	Response
200	The rest API will respond with a JSON objects containing the following properties: <ul style="list-style-type: none"><li>- success</li></ul>
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "Unauthenticated."</li></ul>
500	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "500 Internal Server Error."</li></ul>

**Response example:**

```
{  
  "success": "You have successfully added one email!"  
}
```

## 13. ORDER NOW SEND EMAILS

This method is used to create an order which can use the email functionality enabled in the API.

### Request

Method	URL
POST	api/sendemails

Type	Params	Values
Header	Authorization: Bearer	access_token

Type	Params	Values
POST	order_id	integer
POST	campaign_brand_id	integer
POST	language	String (optional)

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

#### order\_id

order\_id is the ID of the Order Now order for which user wishes to send emails for. To get all eligible Order Now order\_ids use /api/ordernow. A single Order Now order maybe have multiple brands attached, so emails are loaded and sent by brand.

#### campaign\_brand\_id

campaign\_brand\_id is the campaign brand ID received from the success message of the createordernow endpoint. The campaign brand ID is an amalgamation which signifies a particular brand and denomination pair. These integer values are unique and non-repeating.

#### language

The optional language parameter allows emails to be sent in different languages. Examples include “en” for English and “fr” for French. Languages supported is based on Account Setup.

### Request example:

```
{
  "order_id": 5041,
  "campaign_brand_id": 297
}
```

### Request example 2:

```
{
  "order_id": 5106,
  "campaign_brand_id": 250,
  "language": "fr"
}
```

### Response

Status	Response
200	The rest API will respond with a JSON objects containing the following properties: - success
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: - error: "Unauthenticated."
500	The rest API will respond with a JSON object containing the following properties: - error: "500 Internal Server Error."

### Response example:

```
{
  "send": 2
}
```

## 14. ORDER NOW LOAD PHYSICAL CARDS

This method is used to create an order which can use the email functionality enabled in the API.

### Request

Method	URL
POST	api/bulkuploadaddresses

Type	Params	Values
Header	Authorization: Bearer	access_token

Type	Params	Values
POST	order_id	integer
POST	campaign_brand_id	integer
POST	people first_name last_name address_1 address_2 city state zip carrier_line_1 carrier_line_2 fourth_line	Array String String String String String String String String String String

#### access\_token

access\_token must be sent with all client requests. The access\_token helps the server to validate the request source.

#### order\_id

order\_id is the ID of the Order Now order for which user wishes to send emails for. To get all eligible Order Now order\_ids use /api/ordernow. A single Order Now order maybe have multiple brands attached, so emails are loaded and sent by brand.

#### campaign\_brand\_id

campaign\_brand\_id is the campaign brand ID received from the success message of the createordernow endpoint. The campaign brand ID is an amalgamation which signifies a particular brand and denomination pair. These integer values are unique and non-repeating.

### people

people is the array of contact information which will be used to create and send the physical cards. The information required includes contact first and last names and full address. Carrier\_line\_1 and carrier\_line\_2 are two customizable fields which will display on the card carrier letter. Fourth\_line is a customizable message which displays on the card itself under the customer's name.

### Request example:

```
{
  "orderId":"5963",
  "campaign_brand_id":"9429",
  "people": [
    {
      "first_name":"John",
      "last_name": "Doe",
      "address_1": "123 Street",
      "address_2": "Apt 1a",
      "city": "New York",
      "state": "NY",
      "zip": "00001",
      "carrier_line_1":"message 1",
      "carrier_line_2":"message 2",
      "fourth_line":"text under name on card"
    }
  ]
}
```

### Response

Status	Response
200	The rest API will respond with a JSON objects containing the following properties: - success
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.

401	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "Unauthenticated."</li></ul>
500	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "500 Internal Server Error."</li></ul>

**Response example:**

```
{  
  "success"  
}
```

## 15. ORDER NOW REDEEM PHYSICAL CARDS

This method is used to redeem and send the physical cards to the physical addresses loaded in the system using `/api/bulkuploadaddresses`.

### Request

Method	URL
POST	<code>api/redeemphysicalcards</code>

Type	Params	Values
Header	Authorization: Bearer	<code>access_token</code>

Type	Params	Values
POST	<code>order_id</code>	integer
POST	<code>campaign_brand_id</code>	integer

#### **access\_token**

`access_token` must be sent with all client requests. The `access_token` helps the server to validate the request source.

#### **order\_id**

`order_id` is the ID of the Order Now order for which user wishes to send emails for. To get all eligible Order Now `order_ids` use `/api/ordernow`. A single Order Now order maybe have multiple brands attached, so emails are loaded and sent by brand. Do note that only orders which have physical brands are eligible for this endpoint. If the order does not contain a physical brand and error will be returned when hitting this endpoint.

#### **campaign\_brand\_id**

`campaign_brand_id` is the campaign brand ID received from the success message of the `createordernow` endpoint. The campaign brand ID is an amalgamation which signifies a particular brand and denomination pair. These integer values are unique and non-repeating.

### Request example:

```
{  
  "order_id": 5041,  
  "campaign_brand_id": 297  
}
```

## Response

Status	Response
200	The rest API will respond with a JSON objects containing the following properties: <ul style="list-style-type: none"><li>- success</li></ul>
308	The rest API will respond with such a code in the event of attempting to connect to the rest API via http instead of https.
401	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "Unauthenticated."</li></ul>
500	The rest API will respond with a JSON object containing the following properties: <ul style="list-style-type: none"><li>- error: "500 Internal Server Error."</li></ul>

## Response example:

```
{  
  "success"  
}
```

# CODE ACTIVATION

Codes from the sandbox NeoCurrency system can be activated much in the same way that the live system can activate codes. The endpoint for code activation while not directly an API endpoint is <https://redeem.yourdigitalreward.com/sandbox/activate-code/{code}> .

# GLOSSARY

## Dynamic Fields

This a notation which allows the user to pass specific dynamic reward data into the email templates used for sending email rewards in the NeoCurrency system. These fields are denoted by using hash signs (#) so for example #firstname# will dynamically pull in the first name of the customer. This way it's possibly to dynamically send out personalized email templates without having to create multiple email templates for different customers/denominations etc. A list of acceptable dynamic fields is below:

#firstname# - the first name of the customer if a name has been uploaded.

#fullname# - the full name of the customer, this is equivalent to #firstname# #lastname#.

#lastname# - the last name of the customer if a name has been uploaded.

#rewardbrand# - the name of the brand the customer is receiving.

#currency# - the currency symbol of the reward the customer is receiving (example \$ or € or £).

#rewardvalue# - the denomination value of the reward customer is receiving.

An example of API usage using the /api/sendemails endpoint would be as follows

```
{
  "order_id": 1234,
  "campaign_brand_id": 111,
  "subject": "Thank you #firstname#, here is a #currency##rewardvalue# gift",
  "under_header": "#fullname# thank you for being awesome",
  "under_button": "Mr. #lastname#, please visit us again soon!",
  "language": "de"
}
```

## Client\_ID Discussion

When placing orders, some brands, require an additional parameter called `client_id`. This is provided to the user ahead of time. The `client_id` does not change and is tied to the specific user account. Furthermore, brands have affected brands have different `client_id`'s between sandbox and live API environments. The `client_id` is used when making requests for rewards from specific brands and should be passed to both `create_codes` or `createordernow` endpoints. Failure to pass this parameter will result in an error stating that this parameter is required for the specific brand. In most cases usually Visa and Mastercard reward cards are the brands which will need this additional parameter. Below are all current brands which require `client_id` (note a single account may not have access to all these brands. We will try to keep the list amassed here up to date).

Brand				
608	671	672	673	674
674	676	677	678	682
695	804	857	860	861
862	863	864	865	866
867	868			

## Cross Environment Testing

For some clients, we have the option to enable Cross Environment Testing. This means that the client can create campaigns in their live environment and when a campaign is created a duplicate campaign is created in their Sandbox environment. Furthermore, when placing sandbox testing codes, the client can use their live Campaign ID in the sandbox codes/create endpoint and the client will receive sandbox codes while using a live ID (for testing). The idea behind this feature is to ease campaign creation and testing for clients between their live and sandbox accounts. Do note that this functionality will ONLY work for newly created live campaigns. To enable the feature, please contact your reward manager.

## Conventions

- **Client** - Client application.
- **Status** - HTTP status code of response.
- All the possible responses are listed under 'Responses' for each method. Only one of them is issued per request server.
- All responses are in JSON format.
- All request parameters are mandatory unless explicitly marked as [optional]

## Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

3XX - Redirect of some kind

4XX - Error occurred in client's part

5XX - Error occurred in server's part

Status Code	Description
200	OK
201	Created
202	Accepted (Request accepted, and queued for execution)
308	Permanently moved
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable