

## Introduction

Our platform is the primary way for customers to order, track, and manage stored value products. Our API endpoints are only accessible to customers who have been set up by our team and supplied with credentials. Customers with the proper credentials can access the API endpoints by the use of Basic Authentication.

There are several endpoints for use:

- Get catalog
- Get accounts
- Place an order
- Get order

## Authentication

- We use Basic Auth flow [https://en.wikipedia.org/wiki/Basic\\_access\\_authentication](https://en.wikipedia.org/wiki/Basic_access_authentication)
- We create an account for you in the system and generate credentials. We send the credentials to you
- Credentials do not expire
- Credentials can be regenerated by our team. There is no way for you to regenerate credentials yourself. When credentials are regenerating the ClientId remains the same and only ClientSecret is changed
- IP addresses do not need to be whitelisted

## Get catalog

GET /api/integration/v1.0/catalog

You have to include valid credentials in the Authorization header of each GET request. The GET /api/integration/v1.0/catalog endpoint lets you retrieve the list of brands and products available for you. A successful GET request returns a list of available brands in the response body with the information for each brand.

Example request:

```
curl -X GET {host}/api/integration/v1.0/catalog
-H 'Accept: */*'
-H 'Authorization: Basic abc123'
```

Example response:

```
{
  "brands": [
    {
      "name": "string",
      "countryCode": "string",
      "currencyCode": "string",
      "description": "string",
      "disclaimer": "string",
      "redemptionInstructions": "string",
      "terms": "string",
      "logoUrl": "string",
      "modifiedDate": "2020-09-23T15:01:00.8409786",
      "products": [
        {
          "id": 1,
          "name": "string",
          "minFaceValue": 1.00,
          "maxFaceValue": 1000.00,
          "count": 1000,
          "modifiedDate": "2020-09-23T15:01:01.00000"
        }
      ]
    }
  ],
}
```

Response Remarks:

- Description can be empty
- Disclaimer can be empty
- RedemptionInstructions can be empty
- Terms can be empty
- Products[].MinFaceValue indicates a minimum face value of the card if product has the range Products[].denomination
- Products[].MaxFaceValue indicates a maximum face value of the card if product has the range denomination

- For products with the fixed value MinFaceValue and MaxFaceValue are equal
- Products[].Count indicates the number of cards available for the purchase. Id Count is null then cards amount is not defined

#### Remarks:

- Normally a sandbox catalog may contain from 10 to 1000 products. Product list can be discussed with your manager.

## Test Products

1. [Waterstones](#) ( insufficient fund)
2. [Out of Stock](#) (Controlled Out of Stock)
3. [Controlled Test Long Processing](#) (Controlled Test Long Running)
  - **Controller Test Long Running successful after some time. It depends on number of codes requested. The logic there is to process half of remaining codes every 10 seconds. For example, if you ordered 10 codes then at first reprocessing there will be 5 successful codes, then there will be 3 successful out of 5 remaining codes, then 1 and then 1 code. So it will take around 40-50 seconds to complete**
4. [Controlled Test 1 Hour](#) (Controlled Test One Hour)
5. [Controlled Failed Test](#) (Controlled Failed with error message)
6. [Amazon 10 Eur](#) (Success and will generate codes)

## Get accounts

GET /api/integration/v1.0/accounts

You must include valid credentials in the Authorization header of each GET request. The GET /api/integration/v1.0/accounts endpoint lets you retrieve the list of your accounts. A successful GET request returns a list of available accounts in the response body with the information for each account. Use one of the account ids in order to say what account to use when placing the order.

Example request:

```
curl -X GET {host}/api/integration/v1.0/accounts
-H 'Accept: */*'
-H 'Authorization: Basic abc123'
```

Example response:

```
{
  "accounts": [
    {
      "id": 1,
      "currency": "string",
```

```
        "balance": 100.00,  
        "isActive": true  
    }  
  ]  
}
```

## Get exchange rates

GET /api/integration/v1.0/exchange-rates?baseCurrency={baseCurrency}&currency={currency}

You must include valid credentials in the Authorization header of each GET request. The GET /api/integration/v1.0/exchange-rates endpoint lets you retrieve the list of exchange rates that we are using.

If a request is sent without query parameters the endpoint returns a list of all exchange rates based on USD..

If a request contains baseCurrency parameter the endpoint returns a list of all exchange rates based on base currency parameter.

If a request contains baseCurrency and currency parameters the endpoint will return the exchange rate between base currency and currency.

Example request:

```
curl -X GET {host}/api/integration/v1.0/exchange-rates -d "baseCurrency=USD" -d  
"currency=EUR"
```

```
-H 'Accept: */*'
```

```
-H 'Authorization: Basic abc123'
```

Example response:

```
{  
  "baseCurrencyCode": "USD"  
  "rates": [  
    {  
      "value": 0.8714830000,  
      "currencyCode": "EUR"  
    }  
  ]  
}
```

## Place an order

POST /api/integration/v1.0/orders/checkout

You have to include valid credentials in the Authorization header of each POST request. The POST /api/integration/v1.0/orders/checkout endpoint lets you place an order for gift cards. A successful POST request returns requestId, that should be used to get cards

( endpoint - /api/integration/v1.0/orders/{requestId})

Example request:

```
curl -X POST {host}/api/integration/v1.0/orders/checkout
-H 'Accept: */*'
-H 'Authorization: Basic abc123'
```

Body:

```
{
  "RequestId": "97018cf7-98f5-40fc-a142-731d43a72e17",
  "AccountId": 1,
  "Products": [
    {
      "ProductId": 1,
      "Quantity": 1,
      "Value": 1
    }
  ]
}
```

- RequestId should be unique per client
- AccountId must be the id of active account
- ProductId must be the id of product from the catalog
- Value has to be between MinFaceValue and MaxFaceValue inclusively

Example response:

```
{
  "97018cf7-98f5-40fc-a142-711d43a72e17"
}
```

Remarks:

- AccountId has to be retrieved via Get Accounts API call. Use one of the account ids provided by Get Accounts response.
- In case order with specified RequestId already exists, there will be an error saying that order already exists.
- When the request is submitted, the order will be processed immediately. There is no way to create order but not process it.

Limits:

- 1 Create Order requests per 1 second. Otherwise response is 429
- Maximum number of cards in 1 order is 500. Otherwise response is 400

## Get Order

GET/api/integration/v1.0/orders/{requestId}

You have to include valid credentials in the Authorization header of each GET request. The GET/api/integration/v1.0/orders/checkout/{requestId} endpoint lets you get the order. A successful GET request returns order information with cards requested in the response body with the information for each product requested.

```
curl -X GET {host}/api/integration/v1.0/orders/{requestId}
-H 'Accept: */*'
-H 'Authorization: Basic abc123'
```

Body:

```
{
}
```

Example response:

```
{
  "orderId": 1,
  "requestId": "97018cf7-98f5-40fc-a142-711d43a72e17",
  "items": [
    {
      "brandCode": "string",
      "productId": 1,
      "productFaceValue": 1,
      "quantity": 1,
      "pictureUrl": "string",
      "countryCode": "string",
      "currencyCode": "string",
      "cards": [
        {
          "serialNumber": "string",
          "cardCode": "string",
          "pin": "string",
          "expirationDate": "string",
          "status": "string",
        }
      ]
    }
  ],
  "status": "string",
  "createdDate": "2020-09-25T14:04:50.00000",
  "total": 1,
  "errorMessage": "string"
}
```

```
]
}
```

#### Order Status explanation:

**Created, Processed, Succeeded, Failed, PartialFailed**

#### Remarks:

- If you are getting status Pending, this means that order is currently being processed, wait until you get another status.

#### Response Remarks:

- Sometimes, not all cards are bought. In this case a failure message is returned for such cards
- order was Succeeded (successfully processed), Created (but not processed), Failed or PartialFailed (that means that some cards were bought but some were not)
- Total indicates the amount of money in account currency charged from the account

#### Card Status explanation:

**Paused** => We are waiting for these cards from different channels.

**Sold** => Cards that were successfully purchased;

**Canceled** or **Failed** => Cards that were Failed and we won't charge for them;

**Created** => Order processing has not started yet;

**Processing** => We are processing your card in the system;

#### Http status codes:

**503:** Means that API is under maintenance and you need to try again in 10 minutes

**429:** We accept one request per second and return a 429 status code if the limits are exceeded. The limitation only affects the **checkout** function.

#### Before going live

##### Remarks:

- Please, find the appropriate Product Ids in your catalog