

---

# Open Loyalty

Apr 27, 2021



---

## Contents

---

<b>1</b>	<b>The User Guide</b>	<b>3</b>
1.1	The User Guide . . . . .	3
<b>2</b>	<b>The REST API Reference</b>	<b>7</b>
2.1	The REST API Reference . . . . .	7
<b>3</b>	<b>Developer Documentation</b>	<b>233</b>
3.1	The Developer's Guide . . . . .	233
<b>4</b>	<b>The Cookbook</b>	<b>237</b>
4.1	The Cookbook . . . . .	237
<b>5</b>	<b>Changelog</b>	<b>239</b>
5.1	Changelog . . . . .	239
	<b>Index</b>	<b>241</b>



Open Loyalty is a technology for loyalty solutions. It is geared towards starting new loyalty projects based on the [Symfony Framework](#).

---

**Note:** This documentation assumes you have a working knowledge of the Symfony Framework. If you're not familiar with Symfony, please start by reading the [Quick Tour](#) from the Symfony documentation.

---

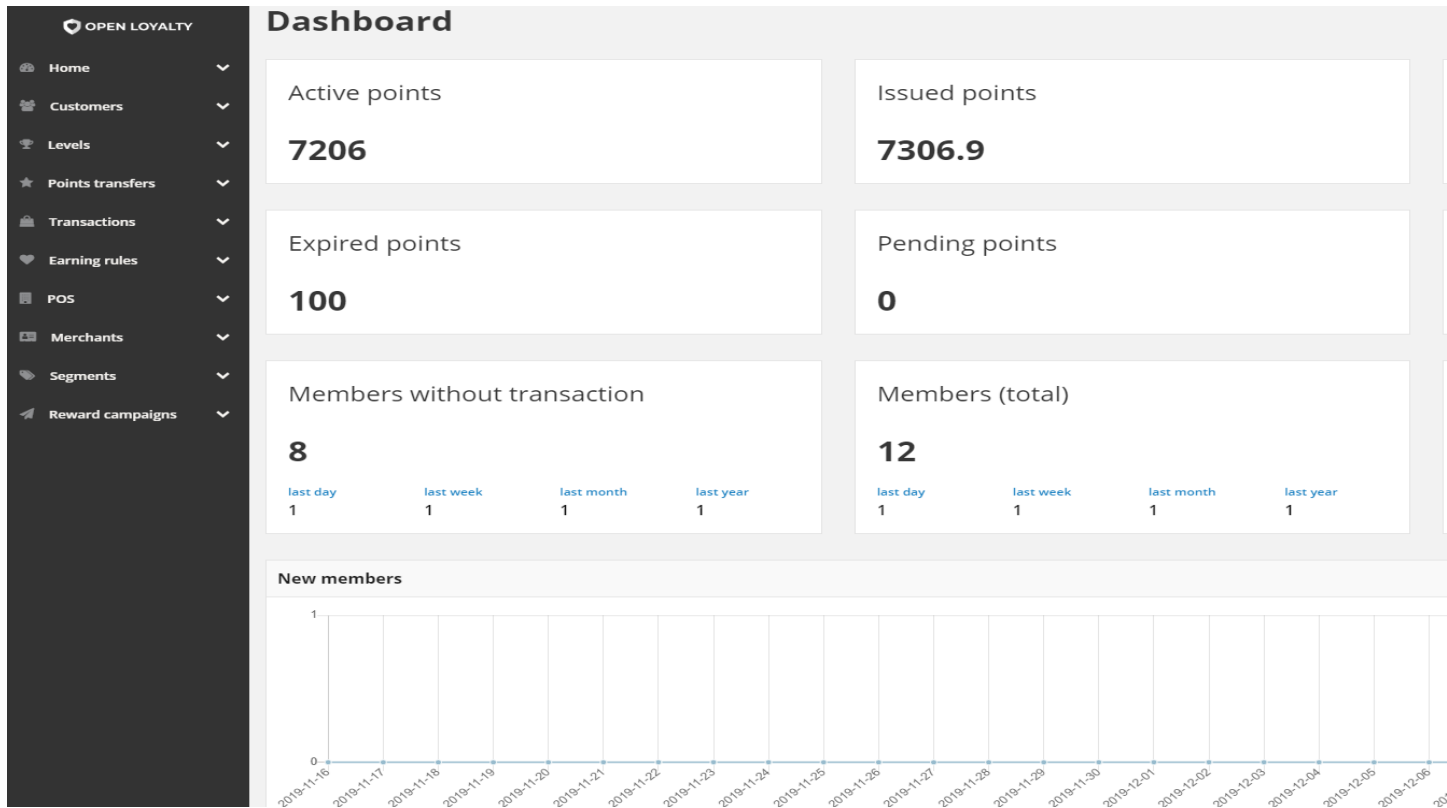


*The Open Loyalty User Guide* is a complete documentation for users.

### 1.1 The User Guide

**Congratulations on your new Loyalty Platform. This manual is intended to help you get the most out of your Loyalty program in your day-to-day use.**

This guide answers the “why, where, and how” questions that most users have when learning to use Open Loyalty platform. You’ll find lots of step-by-step instructions, screenshots, and examples.



Revel Systems offers businesses the ability to customize their loyalty and rewards programs. This feature gives businesses the power to create engaging programs that generate customer loyalty and increase sales. Through this Platform, you can easily manage the rewards and loyalty points to be provided to your customers. Thereafter, the customers can earn or redeem the points as per the rules defined by you.

### 1.1.1 Getting started

The Getting started section introduces your Loyalty Platform Admin, and walks you through the basic configuration settings. You'll get an overview of the resources that are available to you as an Admin of the Open Loyalty and how to log into your Admin account. Finally, you'll learn the concepts of loyalty platform and configuration scope, and establish best practices for project standards and requirements.

### 1.1.2 Customers

The Customers section becomes you familiar with the customer account menu, and learn to manage customer accounts according to adding, deleting and modifying data. You will also learn how to create customer account that can be referenced in customer levels and segments.

### 1.1.3 Levels

In the Levels section of the guide, you will learn how to create and use customer levels to create opportunities for customer engagement and how to set up targeted discounts and rewards based on a variety of conditions. The more points customers receive, the higher level they'll reach. And, the higher level of loyalty, the more rewards customers will get.

You can use levels to offer customer incentives, such as:



- assigned a fixed reward to the particular level. The higher level – the better reward.
- offer limited in time special rewards for customer assigned to particular level

### 1.1.4 Points transfers

The points transfers section of the guide walks you through the basic points transfer information. You will learn how to add and manage transfer of loyalty points records and finally better understand all terms related to points transfer.

### 1.1.5 Transactions

In the Transaction section of the guide, you'll learn how to manage all aspects of the transaction, including matching transaction with customer and better understanding of terms and transaction process.

### 1.1.6 Earning rules

The Earning rules section of the guide provides an overview of the ways for customers to earn points – the engine of your Loyalty Program. You'll learn how to create and manage Earning Rule to accomplish many things, from rewarding high-value customers, to stopping points earning all together.

### 1.1.7 POS

In the POS section of the guide, you'll learn how to set up a stores – online and offline, and manage their data.

### 1.1.8 Merchants

The Merchants section of the guide provides an overview of the all merchants working in yours stores and involve with Loyalty Program. You'll learn how to add and manage merchant data, including assigning to particular store.

### 1.1.9 Segments

In the Segments section of the guide, you'll become familiar with the customer segmentation feature, and learn to configure customer segments according to your preference. You will also learn how to create and assigned customers to segments, that can be referenced in rewards, earning rules, levels etc.

### 1.1.10 Reward Campaigns

In the Reward campaigns section of the guide, you'll familiar with creating and managing rewards available within your Loyalty Program. You will learn what type of rewards can be choose, how to assigned reward to specify customers, define activity time and manage reward details. You will also learn how to verify which rewards have been already redeemed and by which customers.

### Indices and tables

- genindex



---

## The REST API Reference

---

*The API Guide* covers the REST API of the Open Loyalty platform.

### 2.1 The REST API Reference

#### 2.1.1 Introduction to the Open Loyalty REST API

This part of the documentation is about the RESTful JSON API for the Open Loyalty platform.

---

**Note:** To use this documentation, you should have some basic knowledge of [REST APIs](#).

---

#### Get current version

To retrieve the current version of OpenLoyalty, you need to call the `/api/` endpoint with the `GET` method.

#### Definition

```
GET /api/
```

#### Example

```
curl http://localhost:8181/api/ \  
-X "GET" \  
-H "Accept: application/json" \  
-H "Content-type: application/json" \  

```

### Example Response

```
STATUS: 200 OK
```

```
{
  "name": "OpenLoyalty",
  "version": "4.2.0"
}
```

### Health check endpoint

The OpenLoyalty API provides a special endpoint for checking application status. The endpoint verifies the status of critical services like database, Elasticsearch, etc. Calling the `/api/healthcheck` endpoint with the GET method returns 204 code when all services are working correctly. Otherwise, it returns 503 with details about the status of each service.

### Definition

```
GET /api/healthcheck
```

### Example

```
curl http://localhost:8181/api/healthcheck \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/json" \
```

### Example Response

```
STATUS: 503 Unavailable
```

```
[
  {
    "name": "PostgreSQL",
    "status": "Problem",
    "message": "An exception occurred in driver: SQLSTATE[08006] [7] could not
↳translate host name \"db\" to address: Name or service not known"
  },
  {
    "name": "Elasticsearch",
    "status": "OK"
  },
  {
    "name": "RabbitMQ",
    "status": "OK"
  }
]
```

## 2.1.2 Authorization

This part of the documentation is about the authorization process in the Open Loyalty platform through the API. Open Loyalty uses two types of authorization: JSON Web Tokens and permanent API Tokens. In order to check this configuration, please set up your local copy of the Open Loyalty platform and change *localhost* to your address.

### LDAP

By default Open Loyalty authenticates admin users using database. This can be changed by set environment `ADMIN_LDAP_AUTHORIZATION_ENABLED` to true.

---

**Note:** You can enable two authorization methods at the same time but this is not recommended.

---

### JSON Web Token

Open Loyalty has the JWT authorization configured.

To learn what a JSON Web Token is and how it works, check out *Introduction to JSON Web Tokens* <<https://jwt.io/introduction/>>

---

**Note:** The JWT authorization process is used by frontend applications.

---

### Obtain an access token

Send a request with the following parameters:

#### Definition

```
POST /api/<user_type>/login_check
```

Parameter	Parameter type	Description
<code>_username</code>	request	For <code>&lt;user_type&gt;=admin</code> use username, for <code>&lt;user_type&gt;=customer</code> use e-mail address or loyalty card number or phone number, for <code>&lt;user_type&gt;=seller</code> use e-mail address
<code>_password</code>	request	User password
<code>&lt;user_type&gt;</code>	query	Use one of: admin, customer, seller

---

**Note:** Each user type has different permissions to call API methods.

---

### Example

```
curl http://localhost:8181/api/admin/login_check
-H 'Content-Type: application/json;charset=UTF-8'
-H 'Accept: application/json, text/plain, */*'
--data-binary '{"_username":"admin","_password":"open"}'
```

### Example Response

```
{
  "token": "eyJhbGciOiJSUzI1NiIsInR5cCI6I..",
  "refresh_token": "0558f8bb29948c4e54c443f..."
}
```

---

**Note:** Token and refresh token have been shortened for the documentation purpose by suspension points.

---

### Using JSON Web Token

Add authorization header to each request

```
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I..
```

You can now access any API method you want under the /api prefix.

### Example

```
curl http://localhost:8181/api/admin/analytics/customers \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

### Permanent token

A permanent token is a constant string value assigned to the admin account in Open Loyalty or a constant value which is not related to a real user and is stored in the configuration.

### Creating a permanent token in the configuration

In order to activate a configuration access token, you need to add to a Symfony config value

```
parameters:
  master_api_key: 371BBCF483524FD5A837B4095F7FBE96AFD46B678C0F025D5EED0316FD5D7762
```

### Creating a permanent user token

Send a request with the following parameters

## Definition

POST /api/admin/data

Parameter	Parameter type	Description
admin[firstName]	request	First name
admin[lastName]	request	Last name
admin[phone]	request	Phone number
admin[email]	request	E-mail address (required)
admin[plainPassword]	request	Plain password (required if admin[external]=0)
admin[external]	request	Allows to define permanent token. Set 1 if true, otherwise 0
admin[apiKey]	request	Permanent token (required if admin[external]=1)
admin[isActive]	request	Set account active. Set 1 if active, otherwise 0

## Example

```
curl http://localhost:8181/api/admin/data \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IiI6..." \
-d "admin[email]=administrator@example.com" \
-d "admin[external]=1" \
-d "admin[apiKey]=customPermanentToken" \
-d "admin[isActive]=1"
```

## Example Response

STATUS: 200 OK

## Example Fail Response

STATUS: 400 Bad Request

```
{
  "form": {
    "children": {
      "firstName": {},
      "lastName": {},
      "phone": {},
      "email": {
        "errors": [
          "This value is already used."
        ]
      }
    },
    "plainPassword": {},
    "external": {},
    "apiKey": {
```

(continues on next page)

(continued from previous page)

```
    "errors": [
      "This value should not be blank."
    ],
    "isActive": {}
  },
  "errors": []
}
```

### Create a permanent user token using the Admin Cockpit

Create a new account in the administration panel.

---

**Note:** The administration panel is available at <http://localhost:8182/>. To log in, use the standard username “admin” and password “open”.

---

Mark a new account as “external” and provide an “Api key”.

The screenshot shows a web form titled "Add user" with a sub-section "Create user". The form contains the following fields and values:

Field	Value
Name	test
Surname	test
Active	Active
Phone	
E-mail*	test@openloyalty.io
External	<input checked="" type="checkbox"/>
Api key*	123456789

At the bottom of the form are two buttons: "SAVE" (highlighted in red) and "CANCEL".

### How to use a permanent token

A permanent token can be provided using headers or a query parameter.

#### Using headers

```
curl http://localhost:8181/api/admin \
-X "GET" -H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "X-AUTH-TOKEN: customPermanentToken"
```



## Using a query parameter

```
curl http://localhost:8181/api/admin?auth_token=customPermanentToken \
-X "GET" -H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
```

### 2.1.3 Admin Users API

These endpoints will allow you to easily manage admin users.

#### Creating an Admin User

To create a new admin user, you need to call the `/api/admin/data` endpoint with the `POST` method.

#### Definition

```
POST /api/admin/data
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
admin[firstName]	request	First name
admin[lastName]	request	Last name
admin[phone]	request	Phone number
admin[email]	request	E-mail address (required)
admin[plainPassword]	request	Plain password (required if admin[external]=0)
admin[external]	request	Allows to define permanent token. Set 1 if true, otherwise 0
admin[apiKey]	request	Permanent token (required if admin[external]=1)
admin[isActive]	request	Set account active. Set 1 if active, otherwise 0
admin[roles][]	request	Role IDs assigned to this administrator (max 1 role)

#### Example

To create a new admin user, use the method below:

```
curl http://localhost:8181/api/admin/data \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "admin[email]=administrator@example.com" \
-d "admin[external]=0" \
-d "admin[plainPassword]=password1234" \
-d "admin[isActive]=1" \
-d "admin[roles][0]=37"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).



## Example

To see the details of the admin user with `admin = 22200000-0000-474c-b092-b0dd880c07e2`, use the method below:

```
curl http://localhost:8181/api/admin/data/22200000-0000-474c-b092-b0dd880c07e2 \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

STATUS: 200 OK

```
{
  "id": "22200000-0000-474c-b092-b0dd880c07e2",
  "username": "admin",
  "isActive": true,
  "createAt": "2017-09-21T13:54:04+0200",
  "email": "admin@example.com",
  "external": false,
  "dtype": "admin"
}
```

**Note:** The `22200000-0000-474c-b092-b0dd880c07e2` id is an example value. Your value may be different. Check the list of all admin users if you are not sure which id should be used.

## Collection of Admin Users

To retrieve a paginated list of admin users, you need to call the `/api/admin` endpoint with the GET method.

## Definition

GET `/api/admin`

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name, by default = firstName
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC

To see the first page of all admin users, use the method below:

### Example

```
curl http://localhost:8181/api/admin \  
-X "GET" \  
-H "Accept: application/json" \  
-H "Content-type: application/x-www-form-urlencoded" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```
{  
  "users": [  
    {  
      "id": "01b7a62a-640a-4c24-b182-c9f2852cae71",  
      "username": "01b7a62a-640a-4c24-b182-c9f2852cae71",  
      "isActive": true,  
      "createAt": "2017-09-25T09:28:49+0200",  
      "email": "administrator@example.com",  
      "external": true,  
      "apiKey": "customPernamentToken",  
      "dtype": "admin",  
      "roles": [  
        {  
          "id": 38,  
          "name": "Reporter admin",  
          "role": "ROLE_ADMIN",  
          "master": false  
        }  
      ],  
    },  
    {  
      "id": "22200000-0000-474c-b092-b0dd880c07e2",  
      "username": "admin",  
      "isActive": true,  
      "createAt": "2017-09-21T13:54:04+0200",  
      "email": "admin@example.com",  
      "external": false,  
      "dtype": "admin",  
      "roles": [  
        {  
          "id": 38,  
          "name": "Reporter admin",  
          "role": "ROLE_ADMIN",  
          "master": false  
        }  
      ],  
    },  
  ],  
}
```

(continues on next page)

(continued from previous page)

```

    "id": "4383c58e-ff64-4e03-8364-5b716cb3e9e5",
    "username": "4383c58e-ff64-4e03-8364-5b716cb3e9e5",
    "isActive": true,
    "createAt": "2017-09-25T09:33:45+0200",
    "email": "administrato123r@example.com",
    "external": true,
    "apiKey": "customPernamentToken123",
    "dtype": "admin",
    "roles": [
      {
        "id": 38,
        "name": "Reporter admin",
        "role": "ROLE_ADMIN",
        "master": false
      }
    ],
  },
],
"total": 3
}

```

## Updating an Admin User

To fully update an admin user, you need to call the `/api/admin/data/<admin>` endpoint with the PUT method.

### Definition

```
PUT /api/admin/data/<admin>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
admin[firstName]	request	First name
admin[lastName]	request	Last name
admin[phone]	request	Phone number
admin[email]	request	E-mail address (required)
admin[plainPassword]	request	Plain password (required if admin[external]=0)
admin[external]	request	Allows to define permanent token. Set 1 if true, otherwise 0
admin[apiKey]	request	Permanent token (required if admin[external]=1)
admin[isActive]	request	Set account active. Set 1 if active, otherwise 0
admin[roles][]	request	Role IDs assigned to this administrator (max 1 role)

### Example

To fully update the admin user with `id = 22200000-0000-474c-b092-b0dd880c07e2`, use the method below:

```

curl http://localhost:8181/api/admin/data/01b7a62a-640a-4c24-b182-c9f2852cae71 \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \

```

(continues on next page)



(continued from previous page)

```
"phone": {},
"email": {},
"plainPassword": {},
"external": {},
"apiKey": {},
"isActive": {}
}
},
"errors": []
}
```

## 2.1.4 Analytics API

These endpoints will allow you to easily analyze your data in Open Loyalty.

### Getting the number of registered customers

To get the number of registered customers in a loyalty program, you need to call the `/api/<storeCode>/admin/analytics/customers` endpoint with the GET method. Additionally, the method returns the number of registered customers for each of the following past intervals (day, week, month, year).

#### Definition

```
GET /api/<storeCode>/admin/analytics/customers
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the analytics of.

#### Example

```
curl http://localhost:8181/api/DEFAULT/admin/analytics/customers \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

#### Example Response

```
STATUS: 200 OK
```

```
{
  "total": 10,
  "intervals": {
    "in_1_days": 0,
    "in_7_days": 0,
    "in_30_days": 0,
    "in_365_days": 0
  }
}
```

### Getting the number of spent and transferred points

To retrieve the number of spent and transferred points, you need to call the `/api/<storeCode>/admin/analytics/points` endpoint with the GET method.

#### Definition

```
GET /api/<storeCode>/admin/analytics/points
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the analytics of.

#### Example

```
curl http://localhost:8181/api/DEFAULT/admin/analytics/points \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

#### Example Response

```
STATUS: 200 OK
```

```
{
  "totalSpendingTransfers": 1,
  "totalPointsSpent": 100
}
```









### Getting log

To retrieve the action log, you need to call the `/api/audit/log` endpoint with the GET method.

### Definition

```
GET /api/audit/log
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
entityType	query	(optional) Narrow results to given entity type for example: <code>customer</code>
eventType	query	(optional) Narrow results to given event type for example: <code>RegisterCustomer</code>
entityId	query	(optional) Narrow results to given entity ID
username	query	(optional) Narrow results to given username
auditLogId	query	(optional) Narrow results to given audit log ID
createdAt-From	query	(optional) For example <code>2017-09-27</code>
createdAtTo	query	(optional) For example <code>2017-09-27</code>
page	query	(optional) Start from page, by default 1
perPage	query	(optional) Number of items to display per page, by default = 10
sort	query	(optional) Sort by column name, by default = <code>firstName</code>
direction	query	(optional) Direction of sorting [ASC, DESC], by default = ASC

### Example

```
curl http://localhost:8181/api/audit/log \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```
{
  "logs": [
    {
      "auditLogId": {
        "auditLogId": "916e963e-dd14-4ef8-849a-e5b54779657d"
      },
      "createdAt": "2017-09-21T13:54:05+0200",
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    "eventType": "MoveCustomerToLevel",
    "entityType": "customer",
    "entityId": "00000000-0000-474c-b092-b0dd880c07e1",
    "username": "<notlogged>",
    "data": [
      "000096cf-32a3-43bd-9034-4df343e5fd93"
    ]
  },
  {
    "auditLogId": {
      "auditLogId": "1efe9c57-c42f-41a1-988c-c4f5b65382d8"
    },
    "createdAt": "2017-09-21T13:54:05+0200",
    "eventType": "RegisterCustomer",
    "entityType": "customer",
    "entityId": "00000000-0000-474c-b092-b0dd880c07e1",
    "username": "<notlogged>",
    "data": {
      "firstName": "John",
      "lastName": "Doe",
      "gender": "male",
      "phone": "11111",
      "email": "user@example.com",
      "birthDate": 653011200,
      "createdAt": 1470646394,
      "company": {
        "name": "test",
        "nip": "nip"
      },
      "loyaltyCardNumber": "000000",
      "address": {
        "street": "Dmowskiego",
        "address1": "21",
        "city": "Wrocław",
        "country": "pl",
        "postal": "50-300",
        "province": "Dolnośląskie"
      }
    }
  }
],
"total": 92
}

```

## Example

```

curl http://localhost:8181/api/audit/log \
  -G \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I..\" \
  -d "entityType=customer" \
  -d "page=2" \
  -d "perPage=2" \

```

(continues on next page)

(continued from previous page)

```
-d "sort=username" \  
-d "direction=DESC"
```

### Example Response

```
STATUS: 200 OK
```

```
{  
  "logs": [  
    {  
      "auditLogId": {  
        "auditLogId": "b6781066-a292-4043-bd14-52998ee10691"  
      },  
      "createdAt": "2017-09-21T13:54:05+0200",  
      "eventType": "ActivateCustomer",  
      "entityType": "customer",  
      "entityId": "00000000-0000-474c-b092-b0dd880c07e1",  
      "username": "<notlogged>",  
      "data": []  
    },  
    {  
      "auditLogId": {  
        "auditLogId": "4574e09b-280c-4e5d-bdd2-327589c714da"  
      },  
      "createdAt": "2017-09-21T13:54:05+0200",  
      "eventType": "MoveCustomerToLevel",  
      "entityType": "customer",  
      "entityId": "00000000-0000-474c-b092-b0dd880c07e2",  
      "username": "<notlogged>",  
      "data": [  
        "000096cf-32a3-43bd-9034-4df343e5fd93"  
      ]  
    }  
  ],  
  "total": 92  
}
```

### Exporting the view

To export the audit logs view you need to call `/api/audit/log/export` endpoint with the GET method and the same parameters. Pagination does not work in this endpoint, you can only sort the exported entries.

### Definition

```
GET /api/audit/log/export
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
entityType	query	<i>(optional)</i> Narrow results to given entity type for example: customer
eventType	query	<i>(optional)</i> Narrow results to given event type for example: RegisterCustomer
entityId	query	<i>(optional)</i> Narrow results to given entity ID
username	query	<i>(optional)</i> Narrow results to given username
auditLogId	query	<i>(optional)</i> Narrow results to given audit log ID
createdAt-From	query	<i>(optional)</i> For example 2017-09-27
createdAtTo	query	<i>(optional)</i> For example 2017-09-27
sort	query	<i>(optional)</i> Sort by column name, by default = firstName
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC

### Example

```
curl http://localhost:8181/api/audit/log/export \
  -G \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..." \
  -d "entityType=user" \
  -d "sort=username" \
  -d "direction=DESC"
```

### Example Response

```
STATUS: 200 OK
```

```
"Log ID",Username,"User type","User ID","Event type","Entity type","Entity ID",
↪ "Created at", "Additional information", IP
ff9817cd-f393-4e12-9319-4fe7207bd80b,admin,admin,22200000-0000-474c-b092-b0dd880c07e2,
↪ AuthenticationSuccess,user,,2020-03-13T12:12:58+01:00, [],172.22.0.1
39e25450-0969-4b5e-82ff-d083e5b9c7e1,,admin,,AuthenticationFailure,user,,2020-03-
↪ 13T12:12:29+01:00, [],172.22.0.1
```

### Creating an archive

To dump all audit log data older than a year counting from today's midnight into an archived file in the server's archives storage, use `/api/audit/log/archive` endpoint with the POST method.

### Definition

```
POST /api/audit/log/archive
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
archive[beforeDate]	request	Date to which logs are archived

### Example

```
curl http://localhost:8181/api/audit/log/archive \  
-X "POST" \  
-H "Accept: application/json" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..." \  
-d "archive[beforeDate]=2019-03-10"
```

### Example Response

```
STATUS: 200 OK
```

```
{  
  "totalArchived": 92,  
  "filename": "audit_log_archive_before_2019_05_20.xml"  
}
```

### Example Response

```
STATUS: 200 OK
```

```
{  
  "totalArchived": 0,  
  'message': "No logs to archive from this time range. The file was not created."  
}
```

### Example Response

```
STATUS: 200 OK
```

```
{  
  "totalArchived": 0,  
  "filename": "audit_log_archive_before_2019_05_20.xml",  
  "message": "Archive for this date range has already been generated.",  
}
```

### Getting an archive list

To retrieve all archived files in the server's archives storage, use `/api/audit/log/archive` endpoint with the GET method.





### Example Response

```
STATUS: 200 OK
```

```
<?xml version="1.0" encoding="UTF-8"?>
<log>
  <entry id="39e25450-0969-4b5e-82ff-d083e5b9c7e1" createdAt="2019-03-06T12:12:29+01:00" >
    <user>admin</user>
    <userId>56a91360-1100-cc5c-83fe-c7199e88c723</userId>
    <userType>admin</userType>
    <event>AuthenticationFailure</event>
    <entityType>user</entityType>
    <entityId/>
    <data>[]</data>
    <origin>8.8.8.8</origin>
  </entry>
</log>
```

### 2.1.6 Reward Campaigns API

These endpoints will allow you to easily manage Reward Campaigns.

---

**Note:** Each role in Open Loyalty has individual endpoints to manage reward campaigns.

---

#### Redeem cashback (admin)

To redeem cashback for a customer, you need to call the `/api/<storeCode>/admin/campaign/cashback/redeem` endpoint with the POST method.

#### Definition

```
POST /api/<storeCode>/admin/campaign/cashback/redeem
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to redeem cashback from.
customerId	request	Customer ID
pointsAmount	request	Number of points to spend
cashbackId	request	(optional) Cashback id

#### Example

```
curl http://localhost:8181/api/DEFAULT/admin/campaign/cashback/redeem \
-X "POST" \
-H "Accept: application/json" \
```

(continues on next page)



## Open Loyalty

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```
{
  "customerId": "6102cef9-d263-46de-974d-ad2e89f6e81d",
  "pointsAmount": 5,
  "pointValue": 10,
  "rewardAmount": 50,
  "cashbackId" : "972012b8-633d-41e8-be5a-5125c1a5be63"
}
```

### Simulate cashback

To simulate cashback, you need to call the `/api/<storeCode>/admin/campaign/cashback/simulate` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/admin/campaign/cashback/simulate
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store
customerId	request	Customer ID
pointsAmount	request	Number of points to spend

### Example

```
curl http://localhost:8181/api/DEFAULT/admin/campaign/cashback/simulate \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I..\" \
-d "customerId=5bfdded09-0931-4eac-baad-0d663cfd8976" \
  -d "pointsAmount=10"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

## Example Response

STATUS: 200 OK

```
{
  "customerId": "5bfded09-0931-4eac-baad-0d663cfd8976",
  "pointsAmount": 10,
  "pointValue": "3.00",
  "rewardAmount": 30
}
```

## Cashback provider callback

To run a cashback provider callback, you need to call the `/api/<storeCode>/campaign/cashback/callback/<provider>` endpoint with the POST method.

## Definition

POST `/api/<storeCode>/campaign/cashback/callback/<provider>`

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to perform cashback on.
<provider>	request	Provider, possible value: paytm

## Example

```
curl http://localhost:8181/api/DEFAULT/campaign/cashback/callback/paytm \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Content: {"type":null,"requestGuid":null,"orderId":"f0b9914e-cd54-
↪4a85-bc3f-d36e0c37edaa_1c50ef53-ab5b-4d24-9d03-f93c7ec042fe","status":null,
↪"statusCode":"ACCEPTED","statusMessage":"ACCEPTED","response":null,"metadata":null}
↪" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The data in the Content is an example value and depends on the [Cashback][PayTM] Requested PayTM cashback message after redeeming cashback.

### Example Response

STATUS: 200 OK

```
{
  "couponId": "1c50ef53-ab5b-4d24-9d03-f93c7ec042fe",
  "id": "f0b9914e-cd54-4a85-bc3f-d36e0c37edaa_1c50ef53-ab5b-4d24-9d03-f93c7ec042fe",
  "customerId": "f0b9914e-cd54-4a85-bc3f-d36e0c37edaa",
  "code": "ACCEPTED",
  "message": "ACCEPTED",
  "provider": "paytm",
  "failed": false
}
```

### Create a new campaign

To create a new campaign, you need to call the `/api/<storeCode>/campaign` endpoint with the POST method.

### Definition

POST `/api/<storeCode>/campaign`

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to create the campaign in.
campaign[reward]	request	Campaign type. Possible types: discount_code, free_delivery
campaign[translations][en][name]	request	Campaign name in given locale.
campaign[translations][en][shortDescription]	request	(optional) A short description in given locale.
campaign[translations][en][conditionsDescription]	request	(optional) A description of required conditions to apply in given locale.
campaign[translations][en][usageInstruction]	request	(optional) A little information about how to use coupons in given locale.
campaign[translations][en][brandDescription]	request	(optional) A little information about brand in given locale.
campaign[active]	request	Set 1 if active, otherwise 0
campaign[categories]	request	(optional) Array of category IDs.
campaign[pushNotificationText]	request	Push message sent to a customer on this campaign becoming active.
campaign[pointValue]	request	Each point will be exchanged for provided value (in current locale)
campaign[cashbackProvider]	request	Cashback campaigns can automatically send funds using the provider
campaign[costInPoints]	request	How many points it costs
campaign[target]	request	Set level to choose target from defined levels. Set segment to choose target from defined segments.
campaign[levels]	request	Array of level IDs. (required only if "target=level")
campaign[segments]	request	Array of segment IDs. (required only if "target=segment")
campaign[labels]	request	(optional) Informational labels in format "key:value;key1:value1"
campaign[unlimited]	request	Set 1 if unlimited, otherwise 0
campaign[singleCoupon]	request	Set 1 if single coupon, otherwise 0
campaign[limit]	request	Global campaign usage limit. (required only if "unlimited=0")
campaign[limitPerUser]	request	Customer campaign usage limit. (required only if "unlimited=0")
campaign[coupons]	request	Array of coupon codes.
campaign[campaignVisibility][allTimeVisible]	request	Set 1 if always visible, otherwise 0
campaign[campaignVisibility][visibleFrom]	request	Campaign visible from YYYY-MM-DD HH:mm, for example 2017-01-01 00:00
campaign[campaignVisibility][visibleTo]	request	Campaign visible to YYYY-MM-DD HH:mm, for example 2017-01-01 00:00

Table 1 – continued from previous page

Parameter	Parameter type	Description
campaign[campaignActivity][allTimeActive]	request	Set 1 if always active, otherwise 0
campaign[campaignActivity][activeFrom]	request	Campaign active from YYYY-MM-DD HH:mm, for example
campaign[campaignActivity][activeTo]	request	Campaign visible to YYYY-MM-DD HH:mm, for example
campaign[daysInactive]	request	Number of days, during which coupon will not be active after
campaign[daysValid]	request	Number of days, during which coupon will be valid, after ac

## Example

```
curl http://localhost:8181/api/DEFAULT/campaign \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
  -d "campaign[translations][en][reward]=discount_code" \
  -d "campaign[translations][en][name]=Discount+Code+Campaign" \
  -d
↪ "campaign[translations][en][shortDescription]=A+short+description+of+discount+code+campaign
↪ " \
  -d
↪ "campaign[translations][en][conditionsDescription]=Discount+code+for+registration" \
  -d "campaign[translations][en][usageInstruction]=Use+discount+code+as+you+like" \
  -d "campaign[translations][en][brandDescription]=Some+brand+description" \
  -d "campaign[active]=1" \
  -d "campaign[costInPoints]=100" \
  -d "campaign[target]=level" \
  -d "campaign[labels]=type:promotion;type:cashback" \
  -d "campaign[levels][0]=e82c96cf-32a3-43bd-9034-4df343e5fd94" \
  -d "campaign[levels][1]=000096cf-32a3-43bd-9034-4df343e5fd94" \
  -d "campaign[unlimited]=0" \
  -d "campaign[singleCoupon]=0" \
  -d "campaign[limit]=10" \
  -d "campaign[limitPerUser]=1" \
  -d "campaign[daysValid]=0" \
  -d "campaign[daysInactive]=0" \
  -d "campaign[coupons][0]=testCoupon" \
  -d "campaign[coupons][1]=DiscountCoupon" \
  -d "campaign[campaignVisibility][allTimeVisible]=0" \
  -d "campaign[campaignVisibility][visibleFrom]=2017-10-05+10:59" \
  -d "campaign[campaignVisibility][visibleTo]=2018-10-05+10:59" \
  -d "campaign[campaignActivity][allTimeActive]=0" \
  -d "campaign[campaignActivity][activeFrom]=2017-09-05+10:59" \
  -d "campaign[campaignActivity][activeTo]=2017-12-05+10:59"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `e82c96cf-32a3-43bd-9034-4df343e5fd94` or `000096cf-32a3-43bd-9034-4df343e5fd94` id are example values. Your value may be different. Check the list of all levels if you are not sure which id should be used.





(continued from previous page)

```

        "brandDescription": {}
      },
      "pl": {
        "children": {
          "name": {},
          "shortDescription": {},
          "conditionsDescription": {},
          "usageInstruction": {},
          "brandDescription": {}
        }
      }
    },
    "active": {},
    "costInPoints": {},
    "target": {},
    "levels": {},
    "segments": {},
    "unlimited": {},
    "singleCoupon": {},
    "limit": {},
    "limitPerUser": {},
    "coupons": {},
    "daysInactive": {},
    "daysValid": {},
    "campaignVisibility": {
      "children": {
        "allTimeVisible": {},
        "visibleFrom": {},
        "visibleTo": {}
      }
    },
    "campaignActivity": {
      "children": {
        "allTimeActive": {},
        "activeFrom": {},
        "activeTo": {}
      }
    }
  },
  "errors": []
}

```

## Get a collection of campaigns

To retrieve a paginated list of campaigns, you need to call the `/api/<storeCode>/campaign` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/campaign
```



(continued from previous page)

```

{
  "levels": [
    "000096cf-32a3-43bd-9034-4df343e5fd94"
  ],
  "segments": [
    "00000000-0000-0000-0000-000000000002"
  ],
  "coupons": [
    "123"
  ],
  "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93",
  "reward": "discount_code",
  "name": "tests",
  "active": true,
  "costInPoints": 10,
  "singleCoupon": false,
  "unlimited": false,
  "limit": 10,
  "limitPerUser": 2,
  "daysValid": 0,
  "daysInactive": 0,
  "campaignActivity": {
    "allTimeActive": false,
    "activeFrom": "2016-01-01T00:00:00+0100",
    "activeTo": "2018-01-01T00:00:00+0100"
  },
  "campaignVisibility": {
    "allTimeVisible": false,
    "visibleFrom": "2016-01-01T00:00:00+0100",
    "visibleTo": "2018-01-01T00:00:00+0100"
  },
  "segmentNames": {
    "00000000-0000-0000-0000-000000000002": "anniversary"
  },
  "levelNames": {
    "000096cf-32a3-43bd-9034-4df343e5fd94": "level2"
  },
  "labels": [
    {
      "key": "type",
      "value": "promotion"
    }
  ],
  "usageLeft": 1,
  "visibleForCustomersCount": 0,
  "usersWhoUsedThisCampaignCount": 0,
  "hasPhoto": false,
  "translations": [
    {
      "name": "Promotion campaign",
      "shortDescription": "_Campaign_ short description",
      "conditionsDescription": "Some conditions description",
      "usageInstruction": "Usage of coupon instruction",
      "brandDescription": "Brand description",
      "id": 32,
      "locale": "en"
    }
  ],
}

```

(continues on next page)

(continued from previous page)

```

    {
      "name": "Promocyjna kampania",
      "shortDescription": "Opis promocyjnej kampanii",
      "id": 33,
      "locale": "pl"
    }
  ]
},
{
  "levels": [
    "000096cf-32a3-43bd-9034-4df343e5fd94"
  ],
  "segments": [
    "00000000-0000-0000-0000-000000000002"
  ],
  "coupons": [
    "123"
  ],
  "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd92",
  "reward": "discount_code",
  "name": "for test",
  "active": false,
  "costInPoints": 10,
  "singleCoupon": false,
  "unlimited": false,
  "limit": 10,
  "limitPerUser": 2,
  "daysValid": 0,
  "daysInactive": 0,
  "campaignActivity": {
    "allTimeActive": false,
    "activeFrom": "2016-01-01T00:00:00+0100",
    "activeTo": "2018-01-01T00:00:00+0100"
  },
  "campaignVisibility": {
    "allTimeVisible": false,
    "visibleFrom": "2016-01-01T00:00:00+0100",
    "visibleTo": "2018-01-01T00:00:00+0100"
  },
  "segmentNames": {
    "00000000-0000-0000-0000-000000000002": "anniversary"
  },
  "levelNames": {
    "000096cf-32a3-43bd-9034-4df343e5fd94": "level2"
  },
  "will_be_active_from": "2016-01-01T00:00:00+0100",
  "will_be_active_to": "2018-01-01T00:00:00+0100",
  "usageLeft": 1,
  "visibleForCustomersCount": 0,
  "usersWhoUsedThisCampaignCount": 0,
  "hasPhoto": false,
  "translations": [
    {
      "name": "tests",
      "shortDescription": "_shortdescription_",
      "conditionsDescription": "_conditionsdescription_",
      "usageInstruction": "_usageinstruction_"
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

        "brandDescription": "_branddescription_",
        "id": 32,
        "locale": "en"
    },
    {
        "name": "tests_pl",
        "shortDescription": "short desc test pl",
        "id": 33,
        "locale": "pl"
    }
]
},
{
    "levels": [
        "e82c96cf-32a3-43bd-9034-4df343e5fd94",
        "000096cf-32a3-43bd-9034-4df343e5fd94"
    ],
    "segments": [],
    "coupons": [
        "testCoupon",
        "DiscountCoupon"
    ],
    "campaignId": "3062c881-93f3-496b-9669-4238c0a62be8",
    "reward": "discount_code",
    "name": "Discount Code Campaign",
    "shortDescription": "A short description of discount code campaign",
    "conditionsDescription": "Discount code for registration",
    "active": true,
    "costInPoints": 100,
    "singleCoupon": false,
    "unlimited": false,
    "limit": 10,
    "limitPerUser": 1,
    "daysValid": 0,
    "daysInactive": 0,
    "campaignActivity": {
        "allTimeActive": false,
        "activeFrom": "2017-09-05T10:59:00+0200",
        "activeTo": "2017-12-05T10:59:00+0100"
    },
    "campaignVisibility": {
        "allTimeVisible": false,
        "visibleFrom": "2017-10-05T10:59:00+0200",
        "visibleTo": "2018-10-05T10:59:00+0200"
    },
    "usageInstruction": "Use discount code as you like",
    "segmentNames": [],
    "levelNames": {
        "e82c96cf-32a3-43bd-9034-4df343e5fd94": "level1",
        "000096cf-32a3-43bd-9034-4df343e5fd94": "level2"
    },
    "usageLeft": 2,
    "visibleForCustomersCount": 0,
    "usersWhoUsedThisCampaignCount": 0,
    "hasPhoto": false,
    "translations": [
        {

```

(continues on next page)

(continued from previous page)

```

        "name": "tests",
        "shortDescription": "_shortdescription_",
        "conditionsDescription": "_conditionsdescription_",
        "usageInstruction": "_usageinstruction_",
        "brandDescription": "_branddescription_",
        "id": 32,
        "locale": "en"
    },
    {
        "name": "tests_pl",
        "shortDescription": "short desc test pl",
        "id": 33,
        "locale": "pl"
    }
]
},
"total": 3
}

```

## Get a collection of active campaigns

To retrieve a paginated list of active campaigns, you need to call the `/api/<storeCode>/campaign/active` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/campaign/active
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the campaigns from.
format	query	If set to html, the descriptions will be in HTML format

### Example

To see the first page of all campaigns, use the method below:

```

curl http://localhost:8181/api/DEFAULT/campaign/active \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. "

```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

STATUS: 200 OK

```
{
  "campaigns": [
    {
      "id": "000096cf-6361-4d70-e169-676e00000001",
      "name": "Test configured campaign"
    },
    {
      "id": "000096cf-6361-4d70-e169-676e00000003",
      "name": "Test reward campaign"
    },
    {
      "id": "000096cf-6361-4d70-e169-676e11111111",
      "name": "cashback"
    },
    {
      "id": "000096cf-6361-4d70-e169-676e22222222",
      "name": "Percentage discount code"
    },
    {
      "id": "000096cf-6361-4d70-e169-676e55555555",
      "name": "Percentage discount code"
    },
    {
      "id": "000096cf-6361-4d70-e169-676e66666666",
      "name": "Percentage discount code"
    },
    {
      "id": "000096cf-6361-4d70-e169-676e44444444",
      "name": "GEO custom campaign"
    },
    {
      "id": "fce61034-a48e-39f5-af3b-c8aa294601f9"
    },
    {
      "id": "a58388e4-bf99-34d7-9d4a-848efd5b6687",
      "name": "2"
    },
    {
      "id": "8500766f-1aa3-3117-9423-70c6851294c7",
      "name": "4"
    },
    {
      "id": "9ea077ae-6d9f-3547-b43f-cb89471ce4d3",
      "name": "6"
    },
    {
      "id": "0c1f68bc-529f-39b5-99df-b5740048a84a",
      "name": "8"
    },
    {
      "id": "1942beff-5375-3455-ad1d-f608c18b0707",
      "name": "10"
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```
{
  "id": "2bca67fd-2ece-47ea-a556-2ec0b3faeba3",
  "name": "tertrt"
},
{
  "id": "5413dff3-47ba-4342-a669-cc9bb54ea1fa",
  "name": "dddddd"
},
{
  "id": "4cd1415d-6c20-4642-a2eb-cd985c1f88aa",
  "name": "testowe"
},
{
  "id": "40d4b8c5-3be4-4f76-8804-d1dc3c9a9732",
  "name": "test"
},
{
  "id": "110d39ce-47ab-4c2c-b0f8-a71c95e0520a",
  "name": "cashback"
}
]}
```

### Get a collection of bought campaigns

To retrieve a paginated list of bought campaigns, you need to call the `/api/<storeCode>/campaign/bought` endpoint with the GET method.

#### Definition

```
GET /api/<storeCode>/campaign/bought
```



Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the entries from.
used	request	<i>(optional)</i> Possible values : true/false
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC
purchasedAtFrom	query	<i>(optional)</i> Purchase date from filter
purchasedAtTo	query	<i>(optional)</i> Purchase date to filter
usageDateFrom	query	<i>(optional)</i> Usage date from filter
usageDateTo	query	<i>(optional)</i> Usage date to filter
activeSinceFrom	query	<i>(optional)</i> Active since date from filter
activeToFrom	query	<i>(optional)</i> Active since date to filter
activeToTo	query	<i>(optional)</i> Active to date to filter
deliveryStatus	query	<i>(optional)</i> <b>Delivery status filter</b> Possible values: ordered, canceled, shipped, delivered

## Example

To see the first page of all bought campaigns, use the method below:

```
curl http://localhost:8181/api/DEFAULT/campaign/bought \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "boughtCampaigns": [
    {
```

(continues on next page)

(continued from previous page)

```

"canBeUsed": true,
"rewardCampaignId": "000096cf-6361-4d70-e169-676e22222222",
"campaignId": "000096cf-6361-4d70-e169-676e22222222",
"customerId": "7ae0712b-f029-4839-9c53-278c37c6fd35",
"purchasedAt": "2019-03-14T10:29:05+0100",
"coupon": {
  "code": "10",
  "id": "d481c4f2-fa88-476a-9e12-a39f728d94d8"
},
"campaignType": "percentage_discount_code",
"campaignName": "Percentage discount code",
"customerEmail": "maxnowacki690711@test.pl",
"customerName": "Max",
"customerLastname": "Nowacki",
"campaignShippingAddress": {},
"costInPoints": 0,
"currentPointsAmount": 100,
"used": false,
"status": "active",
"transactionId": {
  "transactionId": "33fbbedb5-ff71-4a18-9711-4352d3b9e317"
},
"returnedAmount": 0,
"deliveryStatus": {
  "status": ""
}
},
{
  "canBeUsed": true,
"rewardCampaignId": "000096cf-6361-4d70-e169-676e11111111",
"campaignId": "000096cf-6361-4d70-e169-676e11111111",
"customerId": "6102cef9-d263-46de-974d-ad2e89f6e81d",
"purchasedAt": "2019-03-14T13:45:21+0100",
"coupon": {
  "code": "",
  "id": "6797ed0a-65eb-4a75-b1a2-500b18077dc3"
},
"campaignType": "cashback",
"campaignName": "cashback",
"customerEmail": "maxnowacki209528@test.pl",
"customerName": "Max",
"customerLastname": "Nowacki",
"campaignShippingAddress": {},
"costInPoints": 0,
"currentPointsAmount": 100,
"used": false,
"status": "active",
"returnedAmount": 0,
"deliveryStatus": {
  "status": ""
}
},
{
  "canBeUsed": true,
"rewardCampaignId": "000096cf-6361-4d70-e169-676e22222222",
"campaignId": "000096cf-6361-4d70-e169-676e22222222",
"customerId": "79b5c229-5f9a-4c4b-9acc-7620fb95b38a",

```

(continues on next page)

(continued from previous page)

```

"purchasedAt": "2019-03-14T13:48:11+0100",
"coupon": {
  "code": "40",
  "id": "1a4d7e14-fffc-4049-be41-60e824b5102e"
},
"campaignType": "percentage_discount_code",
"campaignName": "Percentage discount code",
"customerEmail": "test@test.pl",
"customerName": "alajna",
"customerLastname": "user",
"campaignShippingAddress": {},
"costInPoints": 0,
"currentPointsAmount": 100,
"used": false,
"status": "active",
"transactionId": {
  "transactionId": "98b15ef5-94ad-43ef-9984-0d41197d14e6"
},
"returnedAmount": 0,
"deliveryStatus": {
  "status": ""
}
}
],
"total": 3
}

```

## Get a collection of campaigns' purchases exported to a CSV file

To retrieve a paginated list of campaign purchases exported to a CSV file, you need to call the `/api/<storeCode>/campaign/bought/export/csv` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/campaign/bought/export/csv
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store from which the entries should be exported.
purchasedAtFrom	query	<i>(optional)</i> Purchase date from filter
purchasedAtTo	query	<i>(optional)</i> Purchase date to filter

### Example

To see the first page of all campaign purchases in CSV file format, use the method below:

```

curl http://localhost:8181/api/DEFAULT/campaign/bought/export/csv \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I..

```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```
0.Name,1.Date,2.Cost,"3.Tax value",4.email,5.phone,6.Firstname,7.Surname,"8.Points_
↪balance","9.Is used"
"Percentage discount code","2019-03-14 10:29:05",0,,maxnowacki690711@test.pl,,Max,
↪Nowacki,100,
"Percentage discount code","2019-03-14 10:30:18",0,,maxnowacki974845@test.pl,,Max,
↪Nowacki,340,
"Percentage discount code","2019-03-14 10:20:01",0,,test@test.pl,,alajna,user,100,
"Percentage discount code","2019-03-14 10:29:32",0,,maxnowacki856039@test.pl,,Max,
↪Nowacki,340,
gift123,"2019-03-15 08:40:24",3,,maxnowacki209528@test.pl,,Max,Nowacki,95,1
test,"2019-03-15 08:15:14",10,,maxnowacki160093@test.pl,,Max,Nowacki,290,
testowe,"2019-03-14 10:28:20",10,,maxnowacki160093@test.pl,,Max,Nowacki,300,
"Percentage discount code","2019-03-14 09:29:50",0,,user-return@example.com,,
↪TestUser,ForCouponTest,2410,
cashback,"2019-03-14 13:45:21",0,,maxnowacki209528@test.pl,,Max,Nowacki,100,
"Percentage discount code","2019-03-14 13:48:11",0,,test@test.pl,,alajna,user,100,
```

### Get a collection of publicly available campaigns

To retrieve a paginated list of campaigns that are publicly available, you need to call the `/api/<storeCode>/campaign/public/available` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/campaign/public/available
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the campaigns from.
labels	request	<i>(optional)</i> Filter by labels
isFeatured	request	<i>(optional)</i> Filter by featured tag
campaignType	request	<i>(optional)</i> Filter by campaign type
name	request	<i>(optional)</i> Filter by campaign name
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC
categoryId[]	query	<i>(optional)</i> Array of category Ids
format	query	<i>(optional)</i> Format of descriptions [html]. Default is RAW.

## Example

To see the first page of all publicly available campaigns, use the method below:

```
curl http://localhost:8181/api/DEFAULT/campaign/public/available \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

## Update a campaign

To fully update a campaign, you need to call the `/api/<storeCode>/campaign/<campaign>` endpoint with the PUT method.

## Definition

```
PUT /api/<storeCode>/campaign/<campaign>
```

## Example

To fully update a campaign with `id = 3062c881-93f3-496b-9669-4238c0a62be8`, use the method below:

```
curl http://localhost:8181/api/DEFAULT/campaign/3062c881-93f3-496b-9669-4238c0a62be8 \
-X "PUT" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..." \
-d "campaign[reward]=discount_code" \
-d "campaign[translations][en][reward]=discount_code" \
-d "campaign[translations][en][name]=Discount+Code+Campaign" \
-d
↪ "campaign[translations][en][shortDescription]=A+short+description+of+discount+code+campaign
↪ " \
-d
↪ "campaign[translations][en][conditionsDescription]=Discount+code+for+registration" \
-d "campaign[translations][en][usageInstruction]=Use+discount+code+as+you+like" \
-d "campaign[translations][en][brandDescription]=Some+brand+description" \
-d "campaign[active]=1" \
-d "campaign[costInPoints]=100" \
```

(continues on next page)

(continued from previous page)

```
-d "campaign[target]=level" \  
-d "campaign[labels]=type:promotion;type:cashback" \  
-d "campaign[levels][0]=e82c96cf-32a3-43bd-9034-4df343e5fd94" \  
-d "campaign[levels][1]=000096cf-32a3-43bd-9034-4df343e5fd94" \  
-d "campaign[unlimited]=0" \  
-d "campaign[singleCoupon]=0" \  
-d "campaign[limit]=10" \  
-d "campaign[limitPerUser]=1" \  
-d "campaign[daysInactive]=0" \  
-d "campaign[daysValid]=1" \  
-d "campaign[coupons][0]=testCoupon" \  
-d "campaign[coupons][1]=DiscountCoupon" \  
-d "campaign[campaignVisibility][allTimeVisible]=0" \  
-d "campaign[campaignVisibility][visibleFrom]=2017-10-05+10:59" \  
-d "campaign[campaignVisibility][visibleTo]=2018-10-05+10:59" \  
-d "campaign[campaignActivity][allTimeActive]=0" \  
-d "campaign[campaignActivity][activeFrom]=2017-09-05+10:59" \  
-d "campaign[campaignActivity][activeTo]=2017-12-05+10:59" \  
-f "campaign[photos][0]=@/FILE_PATH/FILE_NAME"
```

**Warning:** Remember, you must update the whole data of the campaign.

---

**Note:** The *eyJhbGciOiJSUzI1NiIsInR5cCI6I..* authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

---

**Note:** The *e82c96cf-32a3-43bd-9034-4df343e5fd94* or *000096cf-32a3-43bd-9034-4df343e5fd94* id are example values. Your value may be different. Check the list of all levels if you are not sure which id should be used.

---

---

**Note:** The *testCoupon* or *DiscountCoupon* are example values. You can name code coupons as you like.

---

### Example Response

```
STATUS: 200 OK
```

```
{  
  "campaignId": "3062c881-93f3-496b-9669-4238c0a62be8"  
}
```

### Remove campaign's brand icon

To remove a campaign's brand icon from the campaign, you need to call the `/api/<storeCode>/campaign/<campaign>/brand_icon` endpoint with the DELETE method.

## Definition

```
DELETE /api/<storeCode>/campaign/<campaign>/brand_icon
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the campaign belongs to.
<campaign>	query	Campaign ID

## Example

To remove a brand icon from the campaign `campaign = 000096cf-32a3-43bd-9034-4df343e5fd93`, use the method below:

```
curl http://localhost:8181/api/DEFAULT/campaign/000096cf-32a3-43bd-9034-4df343e5fd93/
↪brand_icon \
  -X "DELETE" \
  -H "Accept: application/json" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `campaign = 000096cf-32a3-43bd-9034-4df343e5fd93` id is an example value. Your value may be different. Check the list of all campaigns if you are not sure which id should be used.

## Example Response

```
STATUS: 204 No Content
```

## Get a campaign's brand icon

To get a campaign's brand icon, you need to call the `/api/<storeCode>/campaign/<campaign>/brand_icon` endpoint with the GET method.

## Definition

```
GET /api/<storeCode>/campaign/<campaign>/brand_icon
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the campaign belongs to.
<campaign>	query	Campaign ID







**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii50a62be8` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

**Note:** Translatable fields (name, short description etc.) are returned in given locale.

---

**Note:** The `3062c881-93f3-496b-9669-4238c0a62be8` id is an example value. Your value may be different. Check the list of all admin users if you are not sure which id should be used.

---

### Example Response

STATUS: 200 OK

```
{
  "levels": [
    "e82c96cf-32a3-43bd-9034-4df343e5fd94",
    "000096cf-32a3-43bd-9034-4df343e5fd94"
  ],
  "segments": [],
  "coupons": [
    "testCoupon",
    "DiscountCoupon"
  ],
  "campaignId": "3062c881-93f3-496b-9669-4238c0a62be8",
  "reward": "discount_code",
  "name": "Discount Code Campaign 1",
  "shortDescription": "A short description of discount code campaign",
  "conditionsDescription": "Discount code for registration",
  "active": true,
  "costInPoints": 100,
  "singleCoupon": false,
  "unlimited": false,
  "limit": 10,
  "limitPerUser": 1,
  "daysValid": 1,
  "daysInactive": 0,
  "campaignActivity": {
    "allTimeActive": false,
    "activeFrom": "2017-09-05T10:59:00+0200",
    "activeTo": "2017-12-05T10:59:00+0100"
  },
  "campaignVisibility": {
    "allTimeVisible": false,
    "visibleFrom": "2017-10-05T10:59:00+0200",
    "visibleTo": "2018-10-05T10:59:00+0200"
  },
  "labels": [
    {
      "key": "type",
      "value": "promotion"
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    ],
    "usageInstruction": "Use discount code as you like",
    "segmentNames": [],
    "levelNames": {
      "e82c96cf-32a3-43bd-9034-4df343e5fd94": "level1",
      "000096cf-32a3-43bd-9034-4df343e5fd94": "level2"
    },
    "usageLeft": 2,
    "visibleForCustomersCount": 0,
    "usersWhoUsedThisCampaignCount": 0,
    "hasPhoto": false,
    "translations": [
      {
        "name": "Discount Code Campaign 1",
        "shortDescription": "A short description of discount code campaign",
        "id": 65,
        "locale": "en"
      },
      {
        "name": "Discount Code Campaign 1 in polish",
        "shortDescription": "A short description of discount code campaign in polish
→",
        "id": 66,
        "locale": "pl"
      }
    ],
    "photos" : [
      {
        "photoId" : "e82c96cf-32a3-43bd-9034-4df343e5f23ed",
        "path" : "campaign_photos/e82c96cf-32a3-43bd-9034-4df343e5fd322294",
        "originalName" : "my_image.png",
        "mimeType" : "image/png"
      }
    ]
  }
}

```

## Get available campaigns for a customer

To check which campaigns are available for a specific customer, you need to call the `/api/<storeCode>/admin/customer/<customer>/campaign/available` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/admin/customer/<customer>/campaign/available
```



(continued from previous page)

```

    "000096cf-32a3-43bd-9034-4df343e5fd93",
    "e82c96cf-32a3-43bd-9034-4df343e5fd94",
    "000096cf-32a3-43bd-9034-4df343e5fd94"
  ],
  "segments": [],
  "coupons": [
    "123"
  ],
  "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93",
  "reward": "discount_code",
  "name": "tests",
  "active": true,
  "costInPoints": 10,
  "singleCoupon": false,
  "unlimited": false,
  "limit": 10,
  "limitPerUser": 2,
  "daysValid": 0,
  "daysInactive": 0,
  "campaignActivity": {
    "allTimeActive": false,
    "activeFrom": "2016-01-01T00:00:00+0100",
    "activeTo": "2018-01-01T00:00:00+0100"
  },
  "campaignVisibility": {
    "allTimeVisible": false,
    "visibleFrom": "2016-01-01T00:00:00+0100",
    "visibleTo": "2018-01-01T00:00:00+0100"
  },
  "segmentNames": [],
  "levelNames": {
    "000096cf-32a3-43bd-9034-4df343e5fd93": "level0",
    "e82c96cf-32a3-43bd-9034-4df343e5fd94": "level1",
    "000096cf-32a3-43bd-9034-4df343e5fd94": "level2"
  },
  "usageLeft": 1,
  "usageLeftForCustomer": 1,
  "canBeBoughtByCustomer": true,
  "visibleForCustomersCount": 2,
  "usersWhoUsedThisCampaignCount": 0,
  "hasPhoto": false,
  "labels": [
    {
      "key": "type",
      "value": "promotion"
    }
  ],
  "total": 1
}

```

### Buy reward campaign for a specific customer (admin)

To buy a reward campaign for a specific customer, you need to call the `/api/<storeCode>/admin/customer/<customer>/campaign/<campaign>/buy` endpoint with the POST method.





```
{
  "customers": [
    {
      "customerId": "00000000-0000-474c-b092-b0dd880c07e1",
      "active": true,
      "firstName": "John",
      "lastName": "Doe",
      "gender": "male",
      "email": "user@example.com",
      "phone": "11111",
      "birthDate": "1990-09-11T02:00:00+0200",
      "createdAt": "2016-08-08T10:53:14+0200",
      "levelId": "000096cf-32a3-43bd-9034-4df343e5fd93",
      "agreement1": false,
      "agreement2": false,
      "agreement3": false,
      "updatedAt": "2017-09-21T13:54:04+0200",
      "campaignPurchases": [],
      "transactionsCount": 1,
      "transactionsAmount": 3,
      "transactionsAmountWithoutDeliveryCosts": 3,
      "amountExcludedForLevel": 0,
      "averageTransactionAmount": 3,
      "lastTransactionDate": "2017-09-22T13:54:08+0200",
      "currency": "eur",
      "levelPercent": "14.00%"
    },
    {
      "customerId": "00000000-0000-474c-b092-b0dd880c07e2",
      "active": true,
      "firstName": "Jane",
      "lastName": "Doe",
      "gender": "male",
      "email": "user-temp@example.com",
      "phone": "111112222",
      "birthDate": "1990-09-11T00:00:00+0200",
      "address": {
        "street": "Test",
        "address1": "1",
        "province": "Mazowieckie",
        "city": "Warszawa",
        "postal": "00-000",
        "country": "PL"
      },
      "loyaltyCardNumber": "0000",
      "createdAt": "2016-08-08T10:53:14+0200",
      "levelId": "e82c96cf-32a3-43bd-9034-4df343e5fd94",
      "manuallyAssignedLevelId": {
        "levelId": "e82c96cf-32a3-43bd-9034-4df343e5fd94"
      },
      "agreement1": true,
      "agreement2": false,
      "agreement3": false,
      "updatedAt": "2017-10-02T11:49:25+0200",
      "campaignPurchases": [
        {
          "purchaseAt": "2017-10-02T12:03:34+0200",

```

(continues on next page)





## Open Loyalty

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

**Note:** The `campaign = 000096cf-32a3-43bd-9034-4df343e5fd93` id and `photoId = 08ae48fd-04b0-4a08-a2a7-fcfa3c4caf5` are example values. Your values may be different. Check the list of all campaigns if you are not sure which id should be used.

---

### Example Response

```
STATUS: 200 OK
```

---

**Note:** In the response you will get raw file content with a proper `Content-Type` header, for example: `Content-Type: image/jpeg`.

---

### Example Response

The campaign may not have a photo at all and you will receive the following response.

```
STATUS: 404 Not Found
```

```
{
  "error": {
    "code": 404,
    "message": "Not Found"
  }
}
```

### Remove a campaign's photo

To remove a campaign's photo, you need to call the `/api/<storeCode>/campaign/<campaign>/photo/<photoId>` endpoint with the `DELETE` method.

### Definition

```
DELETE /api/<storeCode>/campaign/<campaign>/photo/<photoId>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the campaign belongs to.
<campaign>	query	Campaign ID
<photoId>	query	Photo ID



## Open Loyalty

---

```
curl http://localhost:8181/api/DEFAULT/campaign/000096cf-32a3-43bd-9034-4df343e5fd93/
↳photo \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
  -d "photo[file]=C:\fakepath\Photo.png"
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

---

**Note:** The `campaign = 000096cf-32a3-43bd-9034-4df343e5fd93` id is an example value. Your value may be different. Check the list of all campaigns if you are not sure which id should be used.

---

---

**Note:** The `photo[file]=C:\fakepathPhoto.png` is an example value. Your value may be different.

---

### Example Response

```
STATUS: 200 OK
```

### Change a campaign's state

To make a campaign active or inactive, you need to call the `/api/<storeCode>/campaign/<campaign>/<active>` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/campaign/<campaign>/<active>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the activated/deactivated campaign belongs to.
<campaign>	query	Campaign ID
<active>	query	Possible values: active, inactive

### Example

To make the campaign `campaign = 000096cf-32a3-43bd-9034-4df343e5fd93` active, use the method below:

```
curl http://localhost:8181/api/DEFAULT/campaign/000096cf-32a3-43bd-9034-4df343e5fd93/
↳active \
  -X "POST" \
```

(continues on next page)

(continued from previous page)

```
-H "Accept: application/json" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `campaign = 000096cf-32a3-43bd-9034-4df343e5fd93` id is an example value. Your value may be different. Check the list of all campaigns if you are not sure which id should be used.

### Example Response

```
STATUS: 200 OK
```

```
{  
  "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93"  
}
```

### Example

To make the campaign `campaign = 000096cf-32a3-43bd-9034-4df343e5fd93` inactive, use the method below:

```
curl http://localhost:8181/api/DEFAULT/campaign/000096cf-32a3-43bd-9034-4df343e5fd93/  
→inactive \  
-X "POST" \  
-H "Accept: application/json" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `campaign = 000096cf-32a3-43bd-9034-4df343e5fd93` id is an example value. Your value may be different. Check the list of all campaigns if you are not sure which id should be used.

### Example Response

```
STATUS: 200 OK
```

```
{  
  "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93"  
}
```

### Example Not Found Response

```
STATUS: 404 Not Found
```

```
{
  "error": {
    "code": 404,
    "message": "Not Found"
  }
}
```

### Get a campaign collection (seller)

To retrieve a paginated list of campaigns, you need to call the `/api/<storeCode>/seller/campaign` endpoint with the GET method.

#### Definition

```
GET /api/<storeCode>/seller/campaign
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the campaigns from.
page	query	(optional) Start from page, by default 1
perPage	query	(optional) Number of items to display per page, by default = 10
sort	query	(optional) Sort by column name
direction	query	(optional) Direction of sorting [ASC, DESC], by default = ASC

To see the first page of all campaigns, use the method below:

#### Example

```
curl http://localhost:8181/api/DEFAULT/seller/campaign \
-X "GET" -H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

---

**Note:** When using endpoints starting with `/api/<storeCode>/seller`, you need to authorize using seller account credentials.

---

---

**Note:** As a seller, you will receive less information about a campaign than an administrator.

---

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

## Example Response

STATUS: 200 OK

```

{
  "campaigns": [
    {
      "levels": [
        "000096cf-32a3-43bd-9034-4df343e5fd93",
        "e82c96cf-32a3-43bd-9034-4df343e5fd94",
        "000096cf-32a3-43bd-9034-4df343e5fd94"
      ],
      "segments": [],
      "coupons": [
        "123"
      ],
      "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93",
      "reward": "discount_code",
      "name": "tests",
      "active": true,
      "costInPoints": 10,
      "singleCoupon": false,
      "unlimited": false,
      "limit": 10,
      "limitPerUser": 2,
      "campaignActivity": {
        "allTimeActive": false,
        "activeFrom": "2016-01-01T00:00:00+0100",
        "activeTo": "2018-01-01T00:00:00+0100"
      },
      "campaignVisibility": {
        "allTimeVisible": false,
        "visibleFrom": "2016-01-01T00:00:00+0100",
        "visibleTo": "2018-01-01T00:00:00+0100"
      },
      "segmentNames": [],
      "levelNames": {
        "000096cf-32a3-43bd-9034-4df343e5fd93": "level0",
        "e82c96cf-32a3-43bd-9034-4df343e5fd94": "level1",
        "000096cf-32a3-43bd-9034-4df343e5fd94": "level2"
      },
      "labels": [
        {
          "key": "type",
          "value": "promotion"
        }
      ],
      "usageLeft": 0,
      "visibleForCustomersCount": 2,
      "usersWhoUsedThisCampaignCount": 1
    },
    {
      "levels": [
        "000096cf-32a3-43bd-9034-4df343e5fd94"
      ],
      "segments": [
        "00000000-0000-0000-0000-000000000002"
      ]
    }
  ]
}

```

(continues on next page)

```

    ],
    "coupons": [
      "123"
    ],
    "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd92",
    "reward": "discount_code",
    "name": "for test",
    "active": true,
    "costInPoints": 10,
    "singleCoupon": false,
    "unlimited": false,
    "limit": 10,
    "limitPerUser": 2,
    "campaignActivity": {
      "allTimeActive": false,
      "activeFrom": "2016-01-01T00:00:00+0100",
      "activeTo": "2018-01-01T00:00:00+0100"
    },
    "campaignVisibility": {
      "allTimeVisible": false,
      "visibleFrom": "2016-01-01T00:00:00+0100",
      "visibleTo": "2018-01-01T00:00:00+0100"
    },
    "segmentNames": {
      "00000000-0000-0000-0000-000000000002": "anniversary"
    },
    "levelNames": {
      "000096cf-32a3-43bd-9034-4df343e5fd94": "level2"
    },
    "usageLeft": 1,
    "visibleForCustomersCount": 0,
    "usersWhoUsedThisCampaignCount": 0
  }
],
"total": 2
}

```

### Get campaign details (seller)

To retrieve the details of a campaign, you need to call the `/api/<storeCode>/seller/campaign/<campaign>` endpoint with the GET method.

#### Definition

```
GET /api/<storeCode>/seller/campaign/<campaign>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the campaign from.
<campaign>	query	Campaign ID





```
    "value": "promotion"
  }
],
"campaignActivity": {
  "allTimeActive": false,
  "activeFrom": "2017-09-05T10:59:00+0200",
  "activeTo": "2017-12-05T10:59:00+0100"
},
"campaignVisibility": {
  "allTimeVisible": false,
  "visibleFrom": "2017-10-05T10:59:00+0200",
  "visibleTo": "2018-10-05T10:59:00+0200"
},
"usageInstruction": "Use discount code as you like",
"segmentNames": [],
"levelNames": {
  "e82c96cf-32a3-43bd-9034-4df343e5fd94": "level1",
  "000096cf-32a3-43bd-9034-4df343e5fd94": "level2"
},
"usageLeft": 2,
"visibleForCustomersCount": 0,
"usersWhoUsedThisCampaignCount": 0
}
```

### Get available campaigns for a customer (seller)

To check which campaigns are available for a specific customer, you need to call the `/api/<storeCode>/seller/customer/<customer>/campaign/available` endpoint with the GET method.

#### Definition

```
GET /api/<storeCode>/seller/customer/<customer>/campaign/available
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<store-Code>	query	Code of the store to get the campaigns from.
<customer>	query	Customer ID
isFeatured	query	<i>(optional)</i> Filter by featured tag
hasSegment	query	<i>(optional)</i> 1 to return only campaigns offered exclusively to some segments, 0 for campaigns offered only to all segments; omit to return all campaigns
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name. Also available to sort by child fields like <i>campaignVisibility.visibleFrom</i>
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC

### Example

To see the list of campaigns for a customer with the ID `customer = 00000000-0000-474c-b092-b0dd880c07e2`, use the method below:

```
curl http://localhost:8181/api/DEFAULT/seller/customer/00000000-0000-474c-b092-
↪b0dd880c07e2/campaign/available \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** When using endpoints starting with `/api/<storeCode>/seller`, you need to authorize using seller account credentials.

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `00000000-0000-474c-b092-b0dd880c07e2` id is an example value. Your value may be different. Check the list of all customers if you are not sure which id should be used.

### Example Response

```
STATUS: 200 OK
```

```
{
  "campaigns": [
    {
      "levels": [
        "000096cf-32a3-43bd-9034-4df343e5fd93",
        "e82c96cf-32a3-43bd-9034-4df343e5fd94",
        "000096cf-32a3-43bd-9034-4df343e5fd94"
      ],
      "segments": [],
      "coupons": [
        "123"
      ],
      "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93",
      "reward": "discount_code",
      "name": "tests",
      "active": true,
      "costInPoints": 10,
      "singleCoupon": false,
      "unlimited": false,
      "limit": 10,
      "limitPerUser": 2,
      "campaignActivity": {
        "allTimeActive": false,
        "activeFrom": "2016-01-01T00:00:00+0100",
        "activeTo": "2018-01-01T00:00:00+0100"
      },
      "campaignVisibility": {
        "allTimeVisible": false,
        "visibleFrom": "2016-01-01T00:00:00+0100",
        "visibleTo": "2018-01-01T00:00:00+0100"
      },
      "labels": [
        {
          "key": "type",
          "value": "promotion"
        }
      ],
      "segmentNames": [],
      "levelNames": {
        "000096cf-32a3-43bd-9034-4df343e5fd93": "level0",
        "e82c96cf-32a3-43bd-9034-4df343e5fd94": "level1",
        "000096cf-32a3-43bd-9034-4df343e5fd94": "level2"
      },
      "usageLeft": 1,
      "usageLeftForCustomer": 1,
      "canBeBoughtByCustomer": true,
      "visibleForCustomersCount": 2,
      "usersWhoUsedThisCampaignCount": 0
    }
  ],
  "total": 1
}
```

### Buy a reward campaign for a specific customer (seller)

To buy a reward campaign for a specific customer, you need to call the `/api/<storeCode>/seller/customer/<customer>/campaign/<campaign>/buy` endpoint with the POST method.

## Definition

```
POST /api/<storeCode>/seller/customer/<customer>/campaign/<campaign>/buy
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<store-Code>	query	Code of the store the campaign belongs to.
<customer>	query	Customer ID
<campaign>	query	Campaign ID
quantity	query	( <i>optional</i> ) default 1 - number of coupons to buy (not valid for cashback and percentage_discount_code)

## Example

To buy the reward campaign `campaign = 000096cf-32a3-43bd-9034-4df343e5fd93` for the customer `customer = 00000000-0000-474c-b092-b0dd880c07e2` use the method below:

```
curl http://localhost:8181/api/DEFAULT/seller/customer/00000000-0000-474c-b092-
→b0dd880c07e2/campaign/000096cf-32a3-43bd-9034-4df343e5fd93/buy
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** When using endpoints starting with `/api/<storeCode>/seller`, you need to authorize using seller account credentials.

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `000096cf-32a3-43bd-9034-4df343e5fd93` id is an example value. Your value may be different. Check the list of all campaigns if you are not sure which id should be used.

**Note:** The `00000000-0000-474c-b092-b0dd880c07e2` id is an example value. Your value may be different. Check the list of all customers if you are not sure which id should be used.

### Example Response

```
STATUS: 200 OK
```

```
{
  "coupons": [{
    "code": "123",
    "id": "ceb169c7-4fe2-4b49-9f2a-5a18634d7236"
  }]
}
```

### Get all campaigns available for a logged-in customer

To get all campaigns available, you need to call the `/api/<storeCode>/customer/campaign/available` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/customer/campaign/available
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the campaign from.
isFeatured	query	<i>(optional)</i> IsFeatured
page	query	<i>(optional)</i> Page
perPage	query	Number of elements per page
sort	query	Field to sort by
direction	query	Sorting direction
categoryId	query	Sorting direction

### Example

Get all campaigns available for logged in customer.

```
curl http://localhost:8181/api/DEFAULT/customer/campaign/available
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

---

**Note:** When using endpoints starting with `/api/<storeCode>/customer`, you need to authorize using customer account credentials.

---

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

## Change delivery status in a campaign bought by a customer

To change delivery status, use the `/api/<storeCode>/admin/customer/<customer>/bought/coupon/<coupon>/changeDeliveryStatus` endpoint with the PUT method.

### Definition

```
PUT /api/<storeCode>/admin/customer/<customer>/bought/coupon/<coupon>/
↪changeDeliveryStatus
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the updated campaign belongs to.
<customer>	query	Customer ID
<coupon>	query	Coupon ID
deliveryStatus[status]	query	Available statuses: ["canceled", "delivered", "ordered", "shipped"] (required)

### Example

To change delivery status for a customer with id = 5bdab759-5b31-48d6-a38b-ba4628ca1a91 and coupon with id = 42d74422-ca0b-46f4-8871-be26f5a0497e, use the method below:

```
curl http://localhost:8181/api/DEFAULT/admin/customer/5bdab759-5b31-48d6-a38b-
↪ba4628ca1a91/bought/coupon/42d74422-ca0b-46f4-8871-be26f5a0497e/changeDeliveryStatus
-X "PUT" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "deliveryStatus[status]=canceled"
```

**Note:** You can get all available statuses via a settings choice request `/api/<storeCode>/settings/choices/deliveryStatus`

**Note:** When using endpoints starting with `/api/<storeCode>/admin/customer/<customer>/bought/coupon/<couponId>/changeDeliveryStatus`, you need to authorize using admin account credentials.

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```
{
  "success": "Delivery status changed!"
}
```

## 2.1.7 Campaigns categories API

These endpoints allow you to easily manage campaign categories. Campaign categories make it possible to group campaigns into categories. One campaign can be assigned to many categories.

### Create a new campaign category

To create a new category, you need to call the `/api/<storeCode>/campaignCategory` endpoint with the POST method.

#### Definition

```
POST /api/<storeCode>/campaignCategory
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the campaign category should belong to.
campaign_category[translations][en][name]	request	Campaign category name in given locale.
campaign_category[active]	request	Set 1 if active, otherwise 0
campaign_category[sortOrder]	request	Sort order key.

#### Example

```
curl http://localhost:8181/api/DEFAULT/campaignCategory \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "campaign_category[translations][en][name]=Category+A" \
-d "campaign_category[active]=1" \
-d "campaign_category[sortOrder]=0"
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---



## Example Response

```
STATUS: 200 OK
```

```
{
  "campaignCategoryId": "3062c881-93f3-496b-9669-4238c0a62be8"
}
```

## Get a collection of campaign categories

To retrieve a paginated list of campaigns categories, you need to call the `/api/<storeCode>/campaignCategory` endpoint with the GET method.

## Definition

```
GET /api/<storeCode>/campaignCategory
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the campaign categories from.
name	request	<i>(optional)</i> Filter by name
active	request	<i>(optional)</i> Filter by activity
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC
format	query	<i>(optional)</i> Format of descriptions [html]. Default is RAW.
_locale	query	<i>(optional)</i> Retrieves data in given locale

To see the first page of all campaigns categories, use the method below:

## Example

```
curl http://localhost:8181/api/DEFAULT/campaignCategory \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** Translatable fields (name) are returned in given locale.

### Example Response

STATUS: 200 OK

```
{
  "categories": [
    {
      "name": "Category A",
      "campaignCategoryId": "000096cf-32a3-43bd-9034-4df343e5fd99",
      "active": true,
      "sortOrder": 0,
      "translations": [
        {
          "name": "Category A",
          "id": 1,
          "locale": "en"
        },
        {
          "name": "Kategoria A",
          "id": 3,
          "locale": "pl"
        }
      ]
    },
    {
      "name": "Category B",
      "campaignCategoryId": "000096cf-32a3-43bd-9034-4df343e5fd98",
      "active": true,
      "sortOrder": 0,
      "translations": [
        {
          "name": "Category B",
          "id": 2,
          "locale": "en"
        },
        {
          "name": "Kategoria B",
          "id": 4,
          "locale": "pl"
        }
      ]
    }
  ],
  "total": 2
}
```

### Update a campaign

To fully update a campaign, you need to call the `/api/<storeCode>/campaignCategory/<campaign>` endpoint with the PUT method.

## Definition

```
PUT /api/<storeCode>/campaignCategory/<campaignCategory>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the updated campaign category belongs to.
<campaignCategory>	query	Id of the campaign category
campaign_category[translations][en][name]	request	Campaign category name in given locale.
campaign_category[active]	request	Set 1 if active, otherwise 0
campaign_category[sortOrder]	request	Sort order key.

## Example

To fully update a campaign category with id = 3062c881-93f3-496b-9669-4238c0a62be8, use the method below:

```
curl http://localhost:8181/api/DEFAULT/campaignCategory/3062c881-93f3-496b-9669-4238c0a62be8 \
  -X "PUT" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii...\" \
  -d "campaign_category[translations][en][name]=Category+A" \
  -d "campaign_category[active]=1" \
  -d "campaign_category[sortOrder]=0"
```

**Warning:** Remember, you must update all data of the campaign category.

## Example Response

```
STATUS: 200 OK
```

```
{
  "campaignCategoryId": "3062c881-93f3-496b-9669-4238c0a62be8"
}
```

## Get campaign category details

To retrieve the details of a campaign category, you need to call the `/api/<storeCode>/campaignCategory/<campaignCategory>` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/campaignCategory/<campaignCategory>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the campaign category from.
<campaignCategory>	query	Id of the campaign category
_locale	query	<i>(optional)</i> Retrieves data in given locale

### Example

To see the details of the campaign category with `campaignCategory = 3062c881-93f3-496b-9669-4238c0a62be8`, use the method below:

```
curl http://localhost:8181/api/DEFAULT/campaignCategory/3062c881-93f3-496b-9669-4238c0a62be8 \
-X "GET" -H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

---

**Note:** Translatable fields (name) are returned in given locale.

---

### Example Response

```
STATUS: 200 OK
```

```
{
  "name": "Category A",
  "campaignCategoryId": "000096cf-32a3-43bd-9034-4df343e5fd99",
  "active": true,
  "sortOrder": 0,
  "translations": [
    {
      "name": "Category A",
      "id": 1,
      "locale": "en"
    },
    {
      "name": "Kategoria A",
      "id": 3,
      "locale": "pl"
    }
  ]
}
```

## Activate or deactivate campaign category

To activate or deactivate a campaign category, you need to call the `/api/<storeCode>/campaignCategory/<campaignCategory>/active` endpoint with the POST method.

### Definition

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the (de)activated campaign category belongs to.
<campaignCategory>	query	Id of the campaign category
active	boolean	True or False

### Example

```
curl http://localhost:8181/api/DEFAULT/campaignCategory/00ca7e90-6361-4465-e76f-
↪727900000001/active \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
  -d "active=1"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```
204 No Content
```

## 2.1.8 Customer API

These endpoints allow you to easily manage customers.

**Note:** Each role in the Open Loyalty has individual endpoints to manage customers.

### Activate a customer

To activate a customer, you need to call the `/api/<storeCode>/admin/customer/<customer>/activate` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/admin/customer/<customer>/activate
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the activated customer belongs to.
<customer>	request	Customer's UUID

### Example

```
curl http://localhost:8181/api/DEFAULT/admin/customer/1bbafb37-b51b-47c5-b3e4-
↪e0a2d028e655/activate \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

---

**Note:** The `customer = 1bbafb37-b51b-47c5-b3e4-e0a2d028e655` id is an example value. Your value may be different. Check the list of all customers if you are not sure which id should be used.

---

### Example Response

```
STATUS: 200 OK
```

```
(no content)
```

### Deactivate a customer

To deactivate a customer, you need to call the `/api/<storeCode>/admin/customer/<customer>/deactivate` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/admin/customer/<customer>/deactivate
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the deactivated customer belongs to.
<customer>	request	Customer's UUID

## Example

```
curl http://localhost:8181/api/DEFAULT/admin/customer/1bbafb37-b51b-47c5-b3e4-
↪e0a2d028e655/deactivate \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `customer = 1bbafb37-b51b-47c5-b3e4-e0a2d028e655` id is an example value. Your value may be different. Check the list of all customers if you are not sure which id should be used.

## Example Response

```
STATUS: 200 OK
```

```
(no content)
```

## Get customer status

To get a customer status, you need to call the `/api/<storeCode>/admin/customer/<customer>/status` endpoint with the GET method.

## Definition

```
GET /api/<storeCode>/admin/customer/<customer>/status
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the customer from.
<customer>	request	Customer's UUID

## Example

```
curl http://localhost:8181/api/DEFAULT/admin/customer/1bbafb37-b51b-47c5-b3e4-
↪e0a2d028e655/status \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

## Open Loyalty

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6IiwiaWF0IjoiMj01ODUyMjE0In0=` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

**Note:** The `customer = 1bbafb37-b51b-47c5-b3e4-e0a2d028e655` id is an example value. Your value may be different. Check the list of all customers if you are not sure which id should be used.

---

### Example Response

STATUS: 200 OK

```
{
  "firstName": "Test",
  "lastName": "test",
  "customerId": "a284f230-c1c0-4a1c-af9a-159b81de1f2b",
  "points": 0,
  "totalEarnedPoints": 0,
  "usedPoints": 0,
  "expiredPoints": 0,
  "lockedPoints": 0,
  "level": "15.00%",
  "levelName": "level1",
  "levelConditionValue": 20,
  "nextLevel": "20.00%",
  "nextLevelName": "level2",
  "nextLevelConditionValue": 200,
  "transactionsAmountToNextLevelWithoutDeliveryCosts": 100,
  "transactionsAmountWithoutDeliveryCosts": 100,
  "averageTransactionsAmount": "50.00",
  "transactionsCount": 2,
  "transactionsAmount": 100,
  "pointsToNextLevel": 200,
  "currency": "eur",
  "levelWillExpireInDays": 100,
  "pointsSinceLastLevelRecalculation": 0,
  "pointsRequiredToRetainLevel": 20
  "pointsExpiringNextMonth": 150
}
```

**Note:** The information in response may vary depends on the loyalty program configuration. Here is an example of all possible information combine.

---

### Get customers

To get a list of customers, you need to call the `/api/<storeCode>/customer/` endpoint with the GET method.



## Definition

```
GET /api/<storeCode>/customer
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the customers from.
firstName	request	<i>(optional)</i> Customer's first name
lastName	request	<i>(optional)</i> Customer's last name
phone	request	<i>(optional)</i> Customer's phone
email	request	<i>(optional)</i> Customer's email address
loyaltyCard-Number	request	<i>(optional)</i> Customer's loyalty card number
transaction-sAmount-From	request	<i>(optional)</i> Customer's transactions amount lower limit
transaction-sAmountTo	request	<i>(optional)</i> Customer's transactions amount upper limit
average-Transaction-AmountFrom	request	<i>(optional)</i> Customer's average transaction amount lower limit
average-Transaction-AmountTo	request	<i>(optional)</i> Customer's average transaction amount upper limit
transaction-sCountFrom	request	<i>(optional)</i> Customer's transactions count lower limit
transaction-sCountTo	request	<i>(optional)</i> Customer's transactions count upper limit
daysFrom-LastTransactionTo	request	<i>(optional)</i> Customers days from last transaction upper limit
hoursFrom-LastUpdateTo	request	<i>(optional)</i> Customer's hours from last update upper limit
strict	query	<i>(optional)</i> If true, search for exact value, otherwise like value For example 1, by default = 0
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name. Note: column names corresponding to parameters ending in From or To do not have this suffix, eg. transactionCount, averageTransactionAmount.
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC]
_locale	query	<i>(optional)</i> Retrieves data in given locale

## Example

```
curl http://localhost:8181/api/DEFAULT/customer \
-X "GET" \
```

(continues on next page)

(continued from previous page)

```
-H "Accept: application/json" \  
-H "Content-type: application/x-www-form-urlencoded" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```
{  
  "customers": [  
    {  
      "customerId": "41fd3247-2069-4677-8904-584f0ed9f6be",  
      "active": true,  
      "firstName": "test",  
      "lastName": "test",  
      "email": "test4@example.com",  
      "address": {},  
      "createdAt": "2018-02-02T11:39:17+0100",  
      "levelId": "000096cf-32a3-43bd-9034-4df343e5fd93",  
      "agreement1": true,  
      "agreement2": false,  
      "agreement3": false,  
      "updatedAt": "2018-02-02T11:39:28+0100",  
      "campaignPurchases": [],  
      "transactionsCount": 0,  
      "transactionsAmount": 0,  
      "transactionsAmountWithoutDeliveryCosts": 0,  
      "amountExcludedForLevel": 0,  
      "averageTransactionAmount": 0,  
      "currency": "eur",  
      "levelPercent": "14.00%"  
    },  
    {  
      "customerId": "142cbe32-da28-42d0-87aa-f93f3e1ebb91",  
      "active": true,  
      "firstName": "test",  
      "lastName": "test",  
      "email": "test3@example.com",  
      "address": {},  
      "createdAt": "2018-02-02T11:38:19+0100",  
      "levelId": "000096cf-32a3-43bd-9034-4df343e5fd93",  
      "agreement1": true,  
      "agreement2": false,  
      "agreement3": false,  
      "updatedAt": "2018-02-02T11:38:20+0100",  
      "campaignPurchases": [],  
      "transactionsCount": 0,  
      "transactionsAmount": 0,  
      "transactionsAmountWithoutDeliveryCosts": 0,  
    }  
  ]  
}
```

(continues on next page)

(continued from previous page)

```
"amountExcludedForLevel": 0,
"averageTransactionAmount": 0,
"currency": "eur",
"levelPercent": "14.00%"
}
],
"total": 2
}
```

## Example

```
curl http://localhost:8181/api/DEFAULT/customer \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. ."
-d "email=example.com" \
-d "strict=0" \
-d "page=1" \
-d "perPage=2" \
-d "sort=customerId" \
-d "direction=asc"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I.. .` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "customers": [
    {
      "customerId": "00000000-0000-474c-b092-b0dd880c07e2",
      "active": true,
      "firstName": "Jane",
      "lastName": "Doe",
      "gender": "male",
      "email": "user-temp@example.com",
      "phone": "111112222",
      "birthDate": "1990-09-11T02:00:00+0200",
      "address": {
        "street": "Bagno",
        "address1": "1",
        "province": "Mazowieckie",
        "city": "Warszawa",
        "postal": "00-000",
        "country": "PL"
      },
      "loyaltyCardNumber": "0000",
    }
  ],
}
```

(continues on next page)

(continued from previous page)

```

    "createdAt": "2016-08-08T10:53:14+0200",
    "levelId": "000096cf-32a3-43bd-9034-4df343e5fd93",
    "agreement1": false,
    "agreement2": false,
    "agreement3": false,
    "updatedAt": "2018-02-02T11:23:18+0100",
    "campaignPurchases": [],
    "transactionsCount": 1,
    "transactionsAmount": 3,
    "transactionsAmountWithoutDeliveryCosts": 3,
    "amountExcludedForLevel": 0,
    "averageTransactionAmount": 3,
    "lastTransactionDate": "2018-02-03T11:23:21+0100",
    "currency": "eur",
    "levelPercent": "14.00%"
  },
  {
    "customerId": "00000000-0000-474c-b092-b0dd880c07e1",
    "active": false,
    "firstName": "John",
    "lastName": "Doe",
    "gender": "male",
    "email": "user@example.com",
    "phone": "11111",
    "birthDate": "1990-09-11T02:00:00+0200",
    "createdAt": "2016-08-08T10:53:14+0200",
    "levelId": "000096cf-32a3-43bd-9034-4df343e5fd93",
    "agreement1": false,
    "agreement2": false,
    "agreement3": false,
    "updatedAt": "2018-02-02T11:23:17+0100",
    "campaignPurchases": [],
    "transactionsCount": 1,
    "transactionsAmount": 3,
    "transactionsAmountWithoutDeliveryCosts": 3,
    "amountExcludedForLevel": 0,
    "averageTransactionAmount": 3,
    "lastTransactionDate": "2018-02-03T11:23:21+0100",
    "currency": "eur",
    "levelPercent": "14.00%"
  }
],
"total": 2
}

```

## Example

```

curl http://localhost:8181/api/DEFAULT/customer \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..." \
-d "email=example.com" \
-d "strict=1" \

```

(continues on next page)

(continued from previous page)

```
-d "page=1" \  
-d "perPage=2" \  
-d "sort=customerId" \  
-d "direction=asc"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```
{  
  "customers": [],  
  "total": 0  
}
```

### Get a customer list as an admin

To get a list of all customers, you need to call the `/api/<storeCode>/admin/customer` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/admin/customer
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the customers from.
firstName	request	<i>(optional)</i> Customer's first name
lastName	request	<i>(optional)</i> Customer's last name
phone	request	<i>(optional)</i> Customer's phone
email	request	<i>(optional)</i> Customer's email address
loyaltyCard-Number	request	<i>(optional)</i> Customer's loyalty card number
transaction-sAmount-From	request	<i>(optional)</i> Customer's transactions amount lower limit
transaction-sAmountTo	request	<i>(optional)</i> Customer's transactions amount upper limit
average-Transaction-Amount-From	request	<i>(optional)</i> Customer's average transaction amount lower limit
average-Transaction-AmountTo	request	<i>(optional)</i> Customer's average transaction amount upper limit
transaction-sCountFrom	request	<i>(optional)</i> Customer's transactions count lower limit
transaction-sCountTo	request	<i>(optional)</i> Customer's transactions count upper limit
daysFrom-LastTransactionTo	request	<i>(optional)</i> Customers days from last transaction upper limit
hoursFrom-LastUpdateTo	request	<i>(optional)</i> Customer's hours from last update upper limit
strict	query	<i>(optional)</i> If true, search for exact value, otherwise like value For example 1, by default = 0
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name. Note: column names corresponding to parameters ending in From or To do not have this suffix, eg. transactionCount, averageTransactionAmount.
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC]
_locale	query	<i>(optional)</i> Retrieves data in given locale

### Example

```
curl http://localhost:8181/api/DEFAULT/admin/customer \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IiI6... " \
-d "email=example.com" \
-d "strict=0" \
-d "page=1" \
```

(continues on next page)

(continued from previous page)

```
-d "perPage=2" \
-d "sort=customerId" \
-d "direction=asc"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6IiwiaWF0Ijoi` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

STATUS: 200 OK

```
{
  "customers": [
    {
      "customerId": "41fd3247-2069-4677-8904-584f0ed9f6be",
      "active": true,
      "firstName": "test",
      "lastName": "test",
      "email": "test4@example.com",
      "address": {},
      "createdAt": "2018-02-02T11:39:17+0100",
      "levelId": "000096cf-32a3-43bd-9034-4df343e5fd93",
      "agreement1": true,
      "agreement2": false,
      "agreement3": false,
      "updatedAt": "2018-02-02T11:39:28+0100",
      "campaignPurchases": [],
      "transactionsCount": 0,
      "transactionsAmount": 0,
      "transactionsAmountWithoutDeliveryCosts": 0,
      "amountExcludedForLevel": 0,
      "averageTransactionAmount": 0,
      "currency": "eur",
      "levelPercent": "14.00%"
    },
    {
      "customerId": "142cbe32-da28-42d0-87aa-f93f3e1ebb91",
      "active": true,
      "firstName": "test",
      "lastName": "test",
      "email": "test3@example.com",
      "address": {},
      "createdAt": "2018-02-02T11:38:19+0100",
      "levelId": "000096cf-32a3-43bd-9034-4df343e5fd93",
      "agreement1": true,
      "agreement2": false,
      "agreement3": false,
      "updatedAt": "2018-02-02T11:38:20+0100",
      "campaignPurchases": [],
      "transactionsCount": 0,
      "transactionsAmount": 0,
      "transactionsAmountWithoutDeliveryCosts": 0,

```

(continues on next page)

(continued from previous page)

```
"amountExcludedForLevel": 0,
"averageTransactionAmount": 0,
"currency": "eur",
"levelPercent": "14.00%"
}
],
"total": 2
}
```

### Activate a customer using SMS activation token

To activate a customer using a token (sms code), you need to call the `/api/<storeCode>/customer/activate-sms/<token>` endpoint with the `POST` method.

#### Definition

```
POST /api/<storeCode>/customer/activate-sms/<token>
```

Parameter	Parameter type	Description
<code>&lt;storeCode&gt;</code>	query	Code of the store the customer to be activated belongs to.
<code>&lt;token&gt;</code>	request	Customer's token, SMS activation code

#### Example

```
curl http://localhost:8181/api/DEFAULT/customer/activate-sms/954604\
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded"
```

---

**Note:** The `token = 954604` is an example value. Your value may be different.

---

#### Example Response

```
STATUS: 204 No Content
```

```
(no content)
```

### Activate a customer using activation token

To activate a customer using a token, you need to call the `/api/<storeCode>/customer/activate/<token>` endpoint with the `POST` method.





### Example

```
curl http://localhost:8181/api/DEFAULT/admin/customer/c9be6a4e-9a21-414a-bbbd-
↪f506d9ffad85/anonymize \
  -X "POST" \
  -H "Accept:\ application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 204
```

```
(no content)
```

### Check if customer with given phone number or email exists

To check if a customer with a given phone number or email exists, you need to call the `/api/<storeCode>/customer/check` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/customer/check
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to check for customers.
emailOrPhone	request	Customer's email or phone

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/check?emailOrPhone=899000333 \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

## Example Response

```
STATUS: 200 OK
```

```
{  
  "total": 1  
}
```

## Create a new customer

To create a new customer, you need to call the `/api/<storeCode>/customer/register` endpoint with the POST method.

---

**Note:** This endpoint allows you to set more customer parameters than `/api/<storeCode>/customer/self_register` and is used when creating a new customer in the admin cockpit. The self register endpoint is used in the client cockpit for registration and has some limitations.

---

## Definition

```
POST /api/<storeCode>/customer/register
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to register the customer in.
customer[firstName]	request	First name
customer[lastName]	request	Last name
customer[gender]	request	<i>(optional)</i> Gender. Possible values male, female, not_disclosed
customer[email]	request	E-mail address <i>(unique)</i>
customer[phone]	request	<i>(optional)</i> A phone number <i>(unique)</i>
customer[birthDate]	request	<i>(optional)</i> Birth date in format YYYY-MM-DD HH:mm, for example 2017-10-05.
customer[createdAt]	request	<i>(optional)</i> Created at in format YYYY-MM-DD HH:mm:ss, for example 2017-01-01 14:15:16.
customer[address][street]	request	<i>(optional)</i> Street name
customer[address][address1]	request	<i>(optional)</i> Building number
customer[address][address2]	request	<i>(optional)</i> Flat/Unit name
customer[address][postal]	request	<i>(optional)</i> Post code
customer[address][city]	request	<i>(optional)</i> City name
customer[address][province]	request	<i>(optional)</i> Province name
customer[address][country]	request	<i>(optional)</i> Country name
customer[company][name]	request	<i>(optional)</i> Company name
customer[company][nip]	request	<i>(optional)</i> Tax ID
customer[loyaltyCardNumber]	request	<i>(optional)</i> Loyalty card number <i>(unique)</i>
customer[labels]	request	<i>(optional)</i> String of labels in form of key1:val1;key2:val2
customer[agreement1]	request	First agreement. Set 1 if true, otherwise 0
customer[agreement2]	request	<i>(optional)</i> Second agreement. Set 1 if true, otherwise 0
customer[agreement3]	request	<i>(optional)</i> Third agreement. Set 1 if true, otherwise 0
customer[invitationToken]	request	<i>(optional)</i> Invitation token

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/register \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "customer[firstName]=John" \
```

(continues on next page)

(continued from previous page)

```
-d "customer[lastName]=Kowalski" \  
-d "customer[email]=john4@example.com" \  
-d "customer[phone]=000000005000" \  
-d "customer[agreement1]=1"
```

**Note:** The *eyJhbGciOiJSUzI1NiIsInR5cCI6...* authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{  
  "customerId": "e0eb0355-8aaa-4fb1-8159-f58e81b7a25c",  
  "email": "john4@example.com"  
}
```

## Example

```
curl http://localhost:8181/api/DEFAULT/customer/register \  
-X "POST" \  
-H "Accept: application/json" \  
-H "Content-type: application/x-www-form-urlencoded" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..." \  
-d "customer[firstName]=John" \  
-d "customer[lastName]=Kowalski" \  
-d "customer[email]=john3@example.com" \  
-d "customer[phone]=000000004000" \  
-d "customer[birthDate]=1990-01-01" \  
-d "customer[address][street]=Street" \  
-d "customer[address][postal]=00-000" \  
-d "customer[address][city]=Wroclaw" \  
-d "customer[address][province]=Dolnoslaskie" \  
-d "customer[address][country]=Poland" \  
-d "customer[company][nip]=111-222-33-44" \  
-d "customer[company][name]=Company+name" \  
-d "customer[loyaltyCardNumber]=000000000000000002" \  
-d "customer[agreement1]=1" \  
-d "customer[agreement2]=1" \  
-d "customer[agreement3]=1"
```

**Note:** The *eyJhbGciOiJSUzI1NiIsInR5cCI6...* authorization token is an example value. Your value may be different. Read more about Authorization [here](#).



(continued from previous page)

```

    },
    "loyaltyCardNumber": {},
    "agreement1": {},
    "agreement2": {},
    "agreement3": {},
    "levelId": {},
    "posId": {},
    "sellerId": {}
  }
},
"errors": []
}

```

## Get customer details

To get details about a customer, you need to call the `/api/<storeCode>/customer/<customer>` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/customer/<customer>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the customer from.
<customer>	query	Customer ID

### Example

```

curl http://localhost:8181/api/DEFAULT/customer/00000000-0000-474c-b092-b0dd880c07e1 \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. "

```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```

{
  "customerId": "00000000-0000-474c-b092-b0dd880c07e1",
  "active": true,

```

(continues on next page)

(continued from previous page)

```
"firstName": "John",
"lastName": "Doe",
"gender": "male",
"email": "user@example.com",
"phone": "+48234234000",
"birthDate": "1990-09-11T02:00:00+0200",
"lastLevelRecalculation": "2019-03-19T12:00:09+0100",
"loyaltyCardNumber": "47834433524",
"createdAt": "2016-08-08T10:53:14+0200",
"id": "e82c96cf-32a3-43bd-9034-4df343e50000",
"levelId": "e82c96cf-32a3-43bd-9034-4df343e50000",
"agreement1": false,
"agreement2": false,
"agreement3": false,
"status": {
  "availableTypes": [
    "new",
    "active",
    "blocked",
    "deleted"
  ],
  "availableStates": [
    "no-card",
    "card-sent",
    "with-card"
  ],
  "type": "active",
  "state": "no-card"
},
"updatedAt": "2019-03-19T11:52:49+0100",
"campaignPurchases": [],
"transactionsCount": 2,
"transactionsAmount": 3,
"transactionsAmountWithoutDeliveryCosts": 3,
"amountExcludedForLevel": 0,
"averageTransactionAmount": 1.5,
"lastTransactionDate": "2019-03-20T11:52:56+0100",
"labels": [],
"level": {
  "levelId": {
    "id": "e82c96cf-32a3-43bd-9034-4df343e50000",
    "levelId": "e82c96cf-32a3-43bd-9034-4df343e50000"
  },
  "name": "level0",
  "translations": {
    "en": {
      "name": "level0"
    },
    "pl": {
      "name": "poziom0"
    }
  }
},
"version": 7,
"currency": "eur",
"segments": [],
"levelPercent": "0.00%"
```

(continues on next page)



(continued from previous page)

```
}
```

## Update a customer

To update an existing customer, you need to call the `/api/<storeCode>/customer/<customer>` endpoint with the `PUT` method.

---

**Note:** The fields you omit will not be affected. The fields you include and leave empty will have their current values removed. Eg. `customer[email]=&customer[loyaltyCardNumber]=000012` will set `loyaltyCardNumber`, erase email and leave all other fields unaffected.

---

---

**Note:** All simple fields can be updated separately, but compound fields (address, company) must be updated as a whole. Any attempt to update only one of the address fields will result in deleting other parts of the address. Any attempt to update only the name or NIP will result in error code 500.

---

## Definition

```
PUT /api/<storeCode>/customer/<customer>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the updated customer belongs to.
<customer>	query	Customer ID
customer[firstName]	request	<i>(optional)</i> First name
customer[lastName]	request	<i>(optional)</i> Last name
customer[gender]	request	<i>(optional)</i> Gender. Possible values <code>male</code> , <code>female</code>
customer[email]	request	<i>(optional)</i> <i>(unique)</i> E-mail address
customer[phone]	request	<i>(optional)</i> A phone number <i>(unique)</i>
customer[birthDate]	request	<i>(optional)</i> Birth date in format YYYY-MM-DD HH:mm, for example 2017-10-05
customer[createdAt]	request	<i>(optional)</i> Created at in format YYYY-MM-DD HH:mm:ss, for example 2017-01-01 14:15:16.
customer[address][street]	request	<i>(optional)</i> Street name
customer[address][address1]	request	<i>(optional)</i> Building number
customer[address][address2]	request	<i>(optional)</i> Flat/Unit name
customer[address][postal]	request	<i>(optional)</i> Post code
customer[address][city]	request	<i>(optional)</i> City name
customer[address][province]	request	<i>(optional)</i> Province name
customer[address][country]	request	<i>(optional)</i> Country name
customer[company][name]	request	<i>(optional)</i> Company name
customer[company][nip]	request	<i>(optional)</i> Tax ID
customer[loyaltyCardNumber]	request	<i>(optional)</i> Loyalty card number <i>(unique)</i>
customer[labels]	request	<i>(optional)</i> String of labels in form of <code>key1:val1;key2:val2</code> .
customer[agreement1]	request	<i>(optional)</i> First agreement. Set 1 if true, otherwise 0
customer[agreement2]	request	<i>(optional)</i> Second agreement. Set 1 if true, otherwise 0
customer[agreement3]	request	<i>(optional)</i> Third agreement. Set 1 if true, otherwise 0

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/e0eb0355-8aaa-4fb1-8159-f58e81b7a25c \
-X "PUT" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "customer[email]=john4@example.com" \
-d "customer[phone]=" \
```

(continues on next page)

(continued from previous page)

```
-d "customer[agreement2]=1"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```
{
  "customerId": "e0eb0355-8aaa-4fb1-8159-f58e81b7a25c"
}
```

**Note:** In earlier versions, this endpoint returned user data after performing an update. This feature was removed because in certain circumstances old data from before the update could be returned. Use `GET /api/customer/<customer>` after the update to always get the up-to-date values instead.

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/e0eb0355-8aaa-4fb1-8159-f58e81b7a25c \
-X "PUT" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I..\" \
-d "customer[phone]=+440000000"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 400 Bad Request
```

```
{
  "form": {
    "children": {
      "firstName": {},
      "lastName": {},
      "gender": {},
      "email": {},
      "phone": {
        "errors": [
          "This value is not a valid phone number."
        ]
      }
    }
  }
}
```

(continues on next page)

```

    ]
  },
  "birthDate": {},
  "createdAt": {},
  "address": {
    "children": {
      "street": {},
      "address1": {},
      "address2": {},
      "postal": {},
      "city": {},
      "province": {},
      "country": {}
    }
  },
  "company": {
    "children": {
      "name": {},
      "nip": {}
    }
  },
  "loyaltyCardNumber": {},
  "labels": {},
  "agreement1": {},
  "agreement2": {},
  "agreement3": {},
  "levelId": {},
  "posId": {},
  "sellerId": {}
},
"errors": []
}

```

### Confirm a change of authentication credential

To activate a change of phone number when it is used as a log in credential, you need to call `/api/<storeCode>/customer/confirm-change/<token>` endpoint with the POST method.

#### Definition

```
POST /api/<storeCode>/customer/confirm-change/<token>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<token>	request	Customer's token, SMS activation code



**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```
{
  "2018-01-06": 0,
  "2018-01-07": 0,
  "2018-01-08": 0,
  "2018-01-09": 0,
  "2018-01-10": 0,
  "2018-01-11": 0,
  "2018-01-12": 0,
  "2018-01-13": 0,
  "2018-01-14": 0,
  "2018-01-15": 0,
  "2018-01-16": 0,
  "2018-01-17": 0,
  "2018-01-18": 0,
  "2018-01-19": 0,
  "2018-01-20": 0,
  "2018-01-21": 0,
  "2018-01-22": 0,
  "2018-01-23": 0,
  "2018-01-24": 0,
  "2018-01-25": 0,
  "2018-01-26": 0,
  "2018-01-27": 0,
  "2018-01-28": 0,
  "2018-01-29": 0,
  "2018-01-30": 0,
  "2018-01-31": 0,
  "2018-02-01": 0,
  "2018-02-02": 5,
  "2018-02-03": 0,
  "2018-02-04": 0
}
```

### Remove customer's avatar

To remove the avatar of a customer using an admin token, you need to call the `/api/<storeCode>/customer/<customer>/avatar` endpoint with the DELETE method.

### Definition

```
DELETE /api/<storeCode>/customer/<customer>/avatar
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer ID

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/1cb6d205-8b77-40e1-a801-052185ed52d9/
↪avatar \
  -X "DELETE" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 204 No Content
```

### Get a customer's avatar

To get a customer's avatar using an admin token, you need to call the `/api/<storeCode>/customer/<customer>/avatar` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/customer/<customer>/avatar
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer ID

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/1cb6d205-8b77-40e1-a801-052185ed52d9/
↪avatar \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

## Open Loyalty

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 204 No Content
```

### Set customer's avatar

To set a customer's avatar using an admin token, you need to call the `/api/<storeCode>/customer/<customer>/avatar` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/customer/<customer>/avatar
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer ID
avatar[file]	request	Avatar file

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/1cb6d205-8b77-40e1-a801-052185ed52d9/  
→avatar \  
  -X "POST" \  
  -H "Accept: application/json" \  
  -H "Content-type: application/x-www-form-urlencoded" \  
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..." \  
  -d "avatar[file]=C:\\fakepath\\avatar.jpg"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

**Note:** The `photo[file]=C:fakepathavatar.png` is an example value. Your value may be different.

---

### Example Response

```
STATUS: 204 No Content
```



## Assign a customer to a level

To assign a customer to a level using an admin token, you need to call the `/api/<storeCode>/customer/<customer>/level` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/customer/<customer>/level
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer ID
levelId	request	Level ID

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/1cb6d205-8b77-40e1-a801-052185ed52d9/
↪level \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
  -d "levelId=e82c96cf-32a3-43bd-9034-4df343e52222"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```
[ ]
```

## Assign a POS to a customer

To assign a POS to a customer using an admin token, you need to call the `/api/<storeCode>/customer/<customer>/pos` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/customer/<customer>/pos
```





## Open Loyalty

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 204 No Content
```

### Remove a Pushy token

To remove a pushy token, you need to call the `/api/<storeCode>/customer/<customer>/pushy-token/<tokenToRemove>` endpoint with the DELETE method.

### Definition

```
DELETE /api/<storeCode>/customer/<customer>/pushy-token/<tokenToRemove>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer ID
<tokenToRemove>	query	Pushy token to remove

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/1cb6d205-8b77-40e1-a801-052185ed52d9/  
→pushy-token/pushy_token \  
-X "DELETE" \  
-H "Accept: application/json" \  
-H "Content-type: application/x-www-form-urlencoded" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 204 No Content
```

### Remove a customer from a manually assigned level

To unassign a customer from a manually assigned level using a token, you need to call the `/api/<storeCode>/customer/<customer>/remove-manually-level` endpoint with the POST method.

## Definition

```
POST /api/<storeCode>/customer/<customer>/remove-manually-level
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer ID

## Example

```
curl http://localhost:8181/api/DEFAULT/customer/1cb6d205-8b77-40e1-a801-052185ed52d9/
→remove-manually-level \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The *eyJhbGciOiJSUzI1NiIsInR5cCI6I..* authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 204 No Content
```

## Customer status (customer)

To retrieve the status of a customer, you need to call the `/api/<storeCode>/customer/customer/<customer>/status` endpoint with the GET method.

## Definition

```
GET /api/<storeCode>/customer/customer/<customer>/status
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer UUID

## Example



## Definition

```
GET /api/<storeCode>/seller/customer/<customer>/status
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer UUID

## Example

```
curl http://localhost:8181/api/DEFAULT/seller/customer/00000000-0000-474c-b092-
↪b0dd880c07e1/status \
-X "GET" \
-H "Accept:\ application/json" \
-H "Content-type:\ application/x-www-form-urlencoded" \
-H "Authorization:\ Bearer\ eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** When using endpoints starting with `/api/<storeCode>/seller`, you need to authorize using seller account credentials.

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "firstName": "John",
  "lastName": "Doe",
  "customerId": "00000000-0000-474c-b092-b0dd880c07e1",
  "points": 161.9,
  "p2pPoints": 0,
  "totalEarnedPoints": 274.9,
  "usedPoints": 25,
  "expiredPoints": 88,
  "lockedPoints": 0,
  "level": "0.00%",
  "levelName": "level0",
  "levelConditionValue": 0,
  "nextLevel": "5.00%",
  "nextLevelName": "level1",
  "nextLevelConditionValue": 20,
  "transactionsAmountWithoutDeliveryCosts": 3,
  "transactionsAmountToNextLevel": 17,
  "averageTransactionsAmount": "1.50",
```

(continues on next page)





## Send an activation token as a customer

To send/resend an sms activation code, you need to call the `/api/<storeCode>/customer/customer-phone/send-sms-code` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/customer/customer-phone/send-sms-code
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/customer-phone/send-sms-code \
  -X "POST" \
  -H "Accept:\ application/json" \
  -H "Content-type:\ application/x-www-form-urlencoded" \
  -H "Authorization:\ Bearer\ eyJhbGciOiJSUzI1NiIsInR5cCI6IiI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6IiI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

## Send SMS activation code as a seller

To send an SMS activation code to a specific customer, you need to call the `/api/<storeCode>/seller/customer/<customer>/send-sms-code` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/seller/customer/<customer>/send-sms-code
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer UUID





## Open Loyalty

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

**Note:** The `customerId = 00000000-0000-474c-b092-b0dd880c07e1` id is an example value. Your value may be different.

---

### Example Response

```
STATUS: 200 OK
```

### Deactivate a customer as a seller

To deactivate a specific customer, you need to call the `/api/<storeCode>/seller/customer/<customer>/deactivate` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/seller/customer/<customer>/deactivate
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer UUID

### Example

```
curl http://localhost:8181/api/DEFAULT/seller/customer/00000000-0000-474c-b092-
↪b0dd880c07e1/deactivate \
  -X "POST" \
  -H "Accept:\ application/json" \
  -H "Content-type:\ application/x-www-form-urlencoded" \
  -H "Authorization:\ Bearer\ eyJhbGciOiJSUzI1NiIsInR5cCI6Ii.."
```

**Note:** When using endpoints starting with `/api/<storeCode>/seller`, you need to authorize using seller account credentials.

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

---

**Note:** The *customerId* = *00000000-0000-474c-b092-b0dd880c07e1* id is an example value. Your value may be different.

---

### Example Response

```
STATUS: 200 OK
```

### Register new customer as a seller

To register a customer, you need to call the `/api/<storeCode>/seller/customer/register` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/seller/customer/register
```

---

**Note:** This endpoint allows you to set more customer parameters than `/api/<storeCode>/customer/self_register` and is used when creating a new customer in the admin cockpit or POS cockpit. The self register endpoint is used in the client cockpit for registration and has some limitations.

---

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer will belong to.
customer[firstName]	request	First name
customer[lastName]	request	Last name
customer[gender]	request	(optional) Gender. Possible values male, female, not_disclosed
customer[email]	request	(unique) E-mail address
customer[phone]	request	(optional) A phone number (unique)
customer[birthDate]	request	(optional) Birth date in format YYYY-MM-DD HH:mm, for example 2017-10-05
customer[createdAt]	request	(optional) Created at in format YYYY-MM-DD HH:mm:ss, for example 2017-01-01 14:15:16.
customer[address][street]	request	(optional) Street name
customer[address][address1]	request	(optional) Building number
customer[address][address2]	request	(optional) Flat/Unit name
customer[address][postal]	request	(optional) Post code
customer[address][city]	request	(optional) City name
customer[address][province]	request	(optional) Province name
customer[address][country]	request	(optional) Country name
customer[company][name]	request	(optional) Company name
customer[company][nip]	request	(optional) Tax ID
customer[loyaltyCardNumber]	request	(optional) Loyalty card number (unique)
customer[labels]	request	(optional) String of labels in form of key1:val1;key2:val2.
customer[agreement1]	request	First agreement. Set 1 if true, otherwise 0
customer[agreement2]	request	(optional) Second agreement. Set 1 if true, otherwise 0
customer[agreement3]	request	(optional) Third agreement. Set 1 if true, otherwise 0

### Example

```
curl http://localhost:8181/api/DEFAULT/seller/customer/register \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "customer[firstName]=Lady" \
-d "customer[lastName]=Mini" \
-d "customer[email]=test@openloyalty.com" \
```

(continues on next page)

(continued from previous page)

```
-d "customer[gender]=female" \
-d "customer[agreement1]=1"
```

**Note:** When using endpoints starting with `/api/<storeCode>/seller`, you need to authorize using seller account credentials.

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "customerId": "53c16b8e-dble-42f9-af71-3bb76f5c3aca",
  "email": "test@openloyalty.com"
}
```

## Search for customers as a seller

To search for a customer in POS, you need to call the `/api/<storeCode>/pos/search/customer` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/pos/search/customer
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customers belong to.
search[loyaltyCardNumber]	query	<i>(optional)</i> Loyalty card number
search[phone]	request	<i>(optional)</i> A phone number
search[email]	request	<i>(optional)</i> Email address
search[firstName]	request	<i>(optional)</i> First name
search[lastName]	request	<i>(optional)</i> Last name
search[city]	request	<i>(optional)</i> City name
search[postcode]	request	<i>(optional)</i> Post code

## Example

```
curl http://localhost:8181/api/DEFAULT/pos/search/customer \
-X "POST" \
-H "Accept: application/json" \
```

(continues on next page)

(continued from previous page)

```
-H "Content-type: application/x-www-form-urlencoded" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IiI6Ij0iLCJ0eXciOiJ1cm9udGVudC54aW50IiwiaWF0IjoiMj019-03-15T13:00:05+0100" \  
-d "search[firstName]=John" \  
-d "search[lastName]=Doe"
```

**Note:** When using this endpoint, you need to authorize using seller account credentials.

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6IiI6Ij0iLCJ0eXciOiJ1cm9udGVudC54aW50IiwiaWF0IjoiMj019-03-15T13:00:05+0100` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

STATUS: 200 OK

```
{  
  "customers": [  
    {  
      "customerId": "00000000-0000-474c-b092-b0dd880c07e1",  
      "active": false,  
      "posId": "00000000-0000-474c-1111-b0dd880c07e3",  
      "firstName": "John",  
      "lastName": "Doe",  
      "gender": "male",  
      "email": "user@example.com",  
      "phone": "+48234234000",  
      "birthDate": "1990-09-11T02:00:00+0200",  
      "lastLevelRecalculation": "2019-03-15T13:00:05+0100",  
      "loyaltyCardNumber": "47834433524",  
      "createdAt": "2016-08-08T10:53:14+0200",  
      "levelId": "e82c96cf-32a3-43bd-9034-4df343e50000",  
      "agreement1": true,  
      "agreement2": false,  
      "agreement3": false,  
      "status": {  
        "availableTypes": [  
          "new",  
          "active",  
          "blocked",  
          "deleted"  
        ],  
        "availableStates": [  
          "no-card",  
          "card-sent",  
          "with-card"  
        ],  
        "type": "blocked"  
      },  
      "updatedAt": "2019-03-18T14:44:49+0100",  
      "campaignPurchases": [  
        {
```

(continues on next page)



(continued from previous page)

```

    "canBeUsed": false,
    "purchaseAt": "2019-03-18T13:45:39+0100",
    "costInPoints": 1,
    "campaignId": "f1eddc46-e985-43e8-bc2a-8007dca3df95",
    "used": true,
    "coupon": {
      "id": "83d6a65e-d237-4049-84aa-bb107cd6f9a4",
      "code": "test1"
    },
    "status": "active",
    "activeTo": "2019-06-16T13:45:39+0200",
    "deliveryStatus": "ordered",
    "usageDate": "2019-03-18T13:51:10+0100"
  },
  {
    "canBeUsed": false,
    "purchaseAt": "2019-03-18T13:45:39+0100",
    "costInPoints": 1,
    "campaignId": "f1eddc46-e985-43e8-bc2a-8007dca3df95",
    "used": true,
    "coupon": {
      "id": "6a2456ec-49b3-4970-9ac4-75ca01eab0ee",
      "code": "test2"
    },
    "status": "active",
    "activeTo": "2019-06-16T13:45:39+0200",
    "deliveryStatus": "ordered",
    "usageDate": "2019-03-18T13:51:10+0100"
  }
],
"transactionsCount": 2,
"transactionsAmount": 3,
"transactionsAmountWithoutDeliveryCosts": 3,
"amountExcludedForLevel": 0,
"averageTransactionAmount": 1.5,
"lastTransactionDate": "2019-03-16T12:53:23+0100",
"labels": [],
"level": {
  "levelId": {
    "id": "e82c96cf-32a3-43bd-9034-4df343e50000"
  },
  "name": "level0",
  "translations": {
    "en": {
      "name": "level0"
    }
  },
  "pl": {
    "name": "poziom0"
  }
}
},
"currency": "eur",
"levelPercent": "0.00%",
"posIdentifier": "pos2"
},
{
  "customerId": "11111111-0000-474c-b092-b0dd880c07e1",

```

(continues on next page)

(continued from previous page)

```
"active": true,
"firstName": "John1",
"lastName": "Doel",
"gender": "male",
"email": "user-1@example.com",
"phone": "+48456456000",
"birthDate": "1990-09-11T02:00:00+0200",
"lastLevelRecalculation": "2019-03-15T13:00:05+0100",
"createdAt": "2016-08-08T10:53:14+0200",
"levelId": "e82c96cf-32a3-43bd-9034-4df343e50000",
"agreement1": false,
"agreement2": false,
"agreement3": false,
"status": {
  "availableTypes": [
    "new",
    "active",
    "blocked",
    "deleted"
  ],
  "availableStates": [
    "no-card",
    "card-sent",
    "with-card"
  ],
  "type": "active",
  "state": "no-card"
},
"updatedAt": "2019-03-15T12:53:18+0100",
"campaignPurchases": [],
"transactionsCount": 0,
"transactionsAmount": 0,
"transactionsAmountWithoutDeliveryCosts": 0,
"amountExcludedForLevel": 0,
"averageTransactionAmount": 0,
"labels": [
  {
    "key": "test",
    "value": "test"
  }
],
"level": {
  "levelId": {
    "id": "e82c96cf-32a3-43bd-9034-4df343e50000"
  },
  "name": "level0",
  "translations": {
    "en": {
      "name": "level0"
    },
    "pl": {
      "name": "poziom0"
    }
  }
},
"currency": "eur",
"levelPercent": "0.00%"
```

(continues on next page)

(continued from previous page)

```

    }
  ]
}

```

## Import customers

To import customers, you need to call the `/api/<storeCode>/admin/customer/import` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/admin/customer/import
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customers will belong to.
file[file]	query	XML file

### Example

```

curl http://localhost:8181/api/DEFAULT/admin/customer/import \
-X "POST" \
-H "Accept:\ application/json" \
-H "Content-type:\ application/x-www-form-urlencoded" \
-H "Authorization:\ Bearer\ eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "file[file]=C:\\fakepath\\customers.xml"

```

**Note:** When using endpoints starting with `/api/<storeCode>/admin`, you need to authorize using admin account credentials.

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```

{
  "items": [
    {
      "status": "success",
      "processImportResult": {
        "object": "4e2a75c2-f194-40e7-b54e-f208b2fd1732"
      }
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

    },
    "identifier": "aXX2222X1@tXXXXXXst.pl"
  },
  {
    "status": "success",
    "processImportResult": {
      "object": "db081ad2-d035-4edd-8bda-21da198592db"
    },
    "identifier": "222b1222@test.pl"
  },
  {
    "status": "error",
    "message": "Convert exception: birthDate has invalid date format (Y-m-d_
↵required)",
    "identifier": "b22221a@st.pl"
  },
  {
    "status": "success",
    "processImportResult": {
      "object": "c4c169b0-265b-4ead-94c0-1972f181e100"
    },
    "identifier": "aa2222c@dgf.pl"
  },
  {
    "status": "error",
    "message": "Convert exception: gender is required node",
    "identifier": "bz22221z@test.pl"
  },
  {
    "status": "success",
    "processImportResult": {
      "object": "479129a4-283b-414d-b48b-4c3541f9f8d9"
    },
    "identifier": "bxx2222x@teist.pl"
  },
  {
    "status": "success",
    "processImportResult": {
      "object": "00c2f4ff-a8d0-4b31-a119-2bb3f0ec7b6e"
    },
    "identifier": "bx2222x@tetst.pl"
  },
  {
    "status": "error",
    "message": "Convert exception: gender is required node",
    "identifier": "cccc2222cc@test.pl"
  },
  {
    "status": "success",
    "processImportResult": {
      "object": "7f4d0ebd-69e5-488b-b7e2-42985792d63c"
    },
    "identifier": "vvvv111v7@test.pl"
  },
  {
    "status": "success",
    "processImportResult": {

```

(continues on next page)

(continued from previous page)

```
    "object": "36c0e0e7-1231-4817-9fc9-3fd26280026f"
  },
  "identifier": "bb111bbbb@tesyzt.pl"
},
{
  "status": "error",
  "message": "Convert exception: gender is required node",
  "identifier": "nnn111n@test.pl"
}
],
"totalProcessed": 11,
"totalSuccess": 7,
"totalFailed": 4
}
```

### Register a new customer as an admin

To create a new customer, you need to call the `/api/<storeCode>/admin/customer/register` endpoint with the POST method.

**Note:** This endpoint allows you to set more customer parameters than `/api/<storeCode>/customer/self_register` and is used when creating a new customer in the admin cockpit or POS cockpit. The self register endpoint is used in the client cockpit for registration and has some limitations.

### Definition

```
POST /api/<storeCode>/admin/customer/register
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer will belong to.
customer[firstName]	request	First name
customer[lastName]	request	Last name
customer[gender]	request	<i>(optional)</i> Gender. Possible values male, female, not_disclosed
customer[email]	request	<i>(unique)</i> E-mail address
customer[phone]	request	<i>(optional)</i> A phone number <i>(unique)</i>
customer[birthDate]	request	<i>(optional)</i> Birth date in format YYYY-MM-DD HH:mm, for example 2017-10-05
customer[createdAt]	request	<i>(optional)</i> Created at in format YYYY-MM-DD HH:mm:ss, for example 2017-01-01 14:15:16.
customer[address][street]	request	<i>(optional)</i> Street name
customer[address][address1]	request	<i>(optional)</i> Building number
customer[address][address2]	request	<i>(optional)</i> Flat/Unit name
customer[address][postal]	request	<i>(optional)</i> Post code
customer[address][city]	request	<i>(optional)</i> City name
customer[address][province]	request	<i>(optional)</i> Province name
customer[address][country]	request	<i>(optional)</i> Country name
customer[company][name]	request	<i>(optional)</i> Company name
customer[company][nip]	request	<i>(optional)</i> Tax ID
customer[loyaltyCardNumber]	request	<i>(optional)</i> Loyalty card number <i>(unique)</i>
customer[labels]	request	<i>(optional)</i> String of labels in form of key1:val1;key2:val2.
customer[agreement1]	request	First agreement. Set 1 if true, otherwise 0
customer[agreement2]	request	<i>(optional)</i> Second agreement. Set 1 if true, otherwise 0
customer[agreement3]	request	<i>(optional)</i> Third agreement. Set 1 if true, otherwise 0

### Example

```
curl http://localhost:8181/api/DEFAULT/admin/customer/register \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "customer[firstName]=John" \
-d "customer[lastName]=Kowalski" \
-d "customer[email]=john@example.com" \
```

(continues on next page)

(continued from previous page)

```
-d "customer[phone]=0665998332" \
-d "customer[agreement1]=1"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "customerId": "e0eb0355-8aaa-4fb1-8159-f58e81b7a25c",
  "email": "john@example.com"
}
```

## Remove customer

To remove the customer using admin token you need to call the `/api/<storeCode>/customer/<customer>` endpoint with the DELETE method.

## Definition

```
DELETE /api/<storeCode>/customer/<customer>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer ID

## Example

```
curl http://localhost:8181/api/DEFAULT/customer/1cb6d205-8b77-40e1-a801-052185ed52d9 \
-X "DELETE" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value can be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 204 No Content
```

## 2.1.9 Customer Campaign API

These endpoints will allow you to see and use Reward Campaigns for a customer.

### Get all campaigns bought by a customer

To retrieve a list of rewards bought by a specific customer use the `/api/<storeCode>/admin/customer/{customer}/campaign/bought` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/admin/customer/<customer>/campaign/bought
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	request	Customer UUID
includeDetails	query	<i>(optional)</i> Include details about bought campaign For example 1
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name, by default = firstName
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC

### Example

```
curl http://localhost:8181/api/DEFAULT/admin/customer/00000000-0000-474c-b092-
↪b0dd880c07e1/campaign/bought \
-X "GET" \
-H "Accept:application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization:\ Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

**Note:** The `customer = 00000000-0000-474c-b092-b0dd880c07e1` id is an example value. Your value may be different. Check the list of all customers if you are not sure which id should be used.

---



## Example Response

STATUS: 200 OK

```
{
  "campaigns": [],
  "total": 0
}
```

## Example Response

STATUS: 200 OK

```
{
  "campaigns": [
    {
      "purchaseAt": "2018-01-30T18:23:24+0100",
      "costInPoints": 20,
      "campaignId": {
        "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93"
      },
      "used": false,
      "coupon": {
        "code": "123"
      }
    }
  ],
  "total": 1
}
```

## Example

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `customer = 00000000-0000-474c-b092-b0dd880c07e1` id is an example value. Your value may be different. Check the list of all customers if you are not sure which id should be used.

## Example Response

STATUS: 200 OK

```
{
  "campaigns": [
    {
      "purchaseAt": "2018-01-30T18:23:24+0100",
```

(continues on next page)

```
"costInPoints": 20,
"campaignId": {
  "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93"
},
"campaign": {
  "levels": [
    "000096cf-32a3-43bd-9034-4df343e5fd93",
    "e82c96cf-32a3-43bd-9034-4df343e5fd94",
    "000096cf-32a3-43bd-9034-4df343e5fd94",
    "0f0d346e-9fd0-492a-84aa-2a2b61419c97"
  ],
  "segments": [],
  "coupons": [
    "123"
  ],
  "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93",
  "reward": "discount_code",
  "name": "tests",
  "active": true,
  "costInPoints": 20,
  "singleCoupon": false,
  "unlimited": false,
  "limit": 10,
  "limitPerUser": 2,
  "campaignActivity": {
    "allTimeActive": true
  },
  "campaignVisibility": {
    "allTimeVisible": true
  },
  "segmentNames": [],
  "levelNames": {
    "000096cf-32a3-43bd-9034-4df343e5fd93": "level0",
    "e82c96cf-32a3-43bd-9034-4df343e5fd94": "level1",
    "000096cf-32a3-43bd-9034-4df343e5fd94": "level2",
    "0f0d346e-9fd0-492a-84aa-2a2b61419c97": "level3"
  },
  "usageLeft": 0,
  "visibleForCustomersCount": 6,
  "usersWhoUsedThisCampaignCount": 1
},
"used": false,
"coupon": {
  "code": "123"
}
}
],
"total": 1
}
```

### Get all campaigns available for a logged-in customer

To get all campaigns available for a logged-in customer, use the `/api/<storeCode>/customer/campaign/available` endpoint with the GET method.



(continued from previous page)

```

    "reward": "discount_code",
    "name": "for test",
    "active": true,
    "costInPoints": 10,
    "singleCoupon": false,
    "unlimited": false,
    "limit": 10,
    "limitPerUser": 2,
    "campaignActivity": {
      "allTimeActive": true
    },
    "campaignVisibility": {
      "allTimeVisible": true
    },
    "segmentNames": [],
    "levelNames": {
      "000096cf-32a3-43bd-9034-4df343e5fd93": "level0",
      "e82c96cf-32a3-43bd-9034-4df343e5fd94": "level1",
      "000096cf-32a3-43bd-9034-4df343e5fd94": "level2",
      "0f0d346e-9fd0-492a-84aa-2a2b61419c97": "level3"
    },
    "usageLeft": 1,
    "usageLeftForCustomer": 1,
    "canBeBoughtByCustomer": true,
    "visibleForCustomersCount": 6,
    "usersWhoUsedThisCampaignCount": 0
  }
],
"total": 1
}

```

### Get all campaigns bought by a logged-in customer

To get all campaigns bought by a logged-in customer, use the `/api/<storeCode>/customer/campaign/bought` endpoint with the POST method.

#### Definition

```
GET /api/<storeCode>/customer/campaign/bought
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
includeDetails	query	<i>(optional)</i> Include details about bought campaign For example 1
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name, by default = firstName
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC

## Example

```
curl http://localhost:8181/api/DEFAULT/customer/campaign/bought \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IiJ0eS...
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6IiJ0eS...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Warning:** Calling this endpoint is meaningful only when you call it with an authorization token that belongs to the logged-in customer. Otherwise, it will return a 403 Forbidden error response.

## Example Response

```
STATUS: 200 OK
```

```
{
  "campaigns": [
    {
      "purchaseAt": "2018-01-30T18:23:24+0100",
      "costInPoints": 20,
      "campaignId": {
        "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93"
      },
      "used": false,
      "coupon": {
        "code": "123"
      }
    }
  ],
  "total": 1
}
```

## Example

```
curl http://localhost:8181/api/DEFAULT/customer/campaign/bought \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IiJ0eS..." \
  -d "includeDetails=1"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6IiJ0eS...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Warning:** Calling this endpoint is meaningful only when you call it with an authorization token that belongs to the logged-in customer. Otherwise, it will return a 403 Forbidden error response.

### Example Response

STATUS: 200 OK

```
{
  "campaigns": [
    {
      "purchaseAt": "2018-01-30T18:23:24+0100",
      "costInPoints": 20,
      "campaignId": {
        "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93"
      },
      "campaign": {
        "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93",
        "reward": "discount_code",
        "name": "tests",
        "active": true,
        "costInPoints": 20,
        "singleCoupon": false,
        "unlimited": false,
        "limit": 10,
        "limitPerUser": 2,
        "campaignActivity": {
          "allTimeActive": true
        },
        "campaignVisibility": {
          "allTimeVisible": true
        },
        "segmentNames": [],
        "levelNames": {
          "000096cf-32a3-43bd-9034-4df343e5fd93": "level0",
          "e82c96cf-32a3-43bd-9034-4df343e5fd94": "level1",
          "000096cf-32a3-43bd-9034-4df343e5fd94": "level2",
          "0f0d346e-9fd0-492a-84aa-2a2b61419c97": "level3"
        },
        "usageLeft": 0,
        "visibleForCustomersCount": 6,
        "usersWhoUsedThisCampaignCount": 1
      },
      "used": false,
      "coupon": {
        "code": "123"
      }
    }
  ],
  "total": 1
}
```

## Mark multiple coupons as used/unused by a customer.

Mark customer coupons as used/unused using the `/api/<storeCode>/admin/campaign/coupons/mark_as_used` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/admin/campaign/coupons/mark_as_used
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
coupons[][campaignId]	request	Campaign UUID
coupons[][couponId]	request	Coupon UUID
coupons[][code]	request	Coupon code
coupons[][used]	request	Is coupon used, 1 if true, 0 if not used
coupons[][transactionId]	request	( <i>optional</i> ) Transaction ID for which coupon has been used
coupons[][customerId]	request	Customer UUID

### Example

```
curl http://localhost:8181/api/DEFAULT/admin/campaign/coupons/mark_as_used \
-X "POST" -H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "coupons[0][campaignId]=f1eddc46-e985-43e8-bc2a-8007dca3df95" \
-d "coupons[0][couponId]=83d6a65e-d237-4049-84aa-bb107cd6f9a4" \
-d "coupons[0][code]=test1" \
-d "coupons[0][used]=1" \
-d "coupons[0][customerId]=00000000-0000-474c-b092-b0dd880c07e1" \
-d "coupons[0][campaignId]=f1eddc46-e985-43e8-bc2a-8007dca3df95" \
-d "coupons[0][couponId]=6a2456ec-49b3-4970-9ac4-75ca01eab0ee" \
-d "coupons[0][code]=test2" \
-d "coupons[0][used]=1" \
-d "coupons[0][customerId]=00000000-0000-474c-b092-b0dd880c07e1"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `campaignId = f1eddc46-e985-43e8-bc2a-8007dca3df95` id is an example value. Your value may be different.

**Note:** The `couponId = 6a2456ec-49b3-4970-9ac4-75ca01eab0ee` id is an example value. Your value may be different.

**Note:** The `customerId = 00000000-0000-474c-b092-b0dd880c07e1` id is an example value. Your value may be different.

---

### Example Response

STATUS: 200 OK

```
{
  "coupons": [
    {
      "name": "test1",
      "used": true,
      "campaignId": "f1eddc46-e985-43e8-bc2a-8007dca3df95",
      "customerId": "00000000-0000-474c-b092-b0dd880c07e1"
    },
    {
      "name": "test2",
      "used": true,
      "campaignId": "f1eddc46-e985-43e8-bc2a-8007dca3df95",
      "customerId": "00000000-0000-474c-b092-b0dd880c07e1"
    }
  ]
}
```

### Mark a logged-in customer's coupons as used

Mark coupons bought by a logged-in customer as used using the `/api/<storeCode>/customer/campaign/coupons/mark_as_used` endpoint with the POST method.

#### Definition

POST `/api/<storeCode>/customer/campaign/coupons/mark_as_used`

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
coupons[][campaignId]	request	Campaign UUID
coupons[][couponId]	request	Coupon UUID
coupons[][code]	request	Coupon code
coupons[][used]	request	Is coupon used, 1 if true, 0 if not used
coupons[][transactionId]	request	(optional) Transaction ID for which coupon has been used

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/campaign/coupons/mark_as_used \
-X "POST" \
```

(continues on next page)





### Buy a campaign by the logged-in customer

To buy a campaign bought by the logged-in customer, use `/api/<storeCode>/customer/campaign/{campaign}/buy` endpoint with the `POST` method.

#### Definition

```
POST /api/<storeCode>/customer/campaign/<campaign>/buy
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<store-Code>	query	Code of the store the customer belongs to.
campaign	request	Campaign UUID
quantity	query	( <i>optional</i> ) default 1 - number of coupons to buy (not valid for cashback and percentage_discount_code)

#### Example

```
curl http://localhost:8181/api/DEFAULT/customer/campaign/000096cf-32a3-43bd-9034-4df343e5fd92/buy
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Warning:** Calling this endpoint is meaningful only when you call it with an authorization token that belongs to the logged-in customer. Otherwise, it will return a 403 Forbidden error response.

#### Example Response

```
STATUS: 200 OK
```

```
{
  "coupons": [{
    "code": "123",
    "id": "ceb169c7-4fe2-4b49-9f2a-5a18634d7236"
  }]
}
```

## Example Error Response

If there are no more coupons left, you will receive the following responses.

```
STATUS: 400 Bad Request
```

```
{
  "error": "No coupons left"
}
```

## Example Error Response

If you don't have enough points to buy a reward, you will receive following responses.

```
STATUS: 400 Bad Request
```

```
{
  "error": "Not enough points"
}
```

## Get all campaigns bought by a customer (seller)

To retrieve a list of rewards bought by a specific customer, use the `/api/<storeCode>/seller/customer/{customer}/campaign/bought` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/seller/customer/<customer>/campaign/bought
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	request	Customer UUID
includeDetails	query	(optional) Include details about bought campaign For example 1
page	query	(optional) Start from page, by default 1
perPage	query	(optional) Number of items to display per page, by default = 10
sort	query	(optional) Sort by column name, by default = firstName
direction	query	(optional) Direction of sorting [ASC, DESC], by default = ASC

### Example

```
curl http://localhost:8181/api/DEFAULT/seller/customer/00000000-0000-474c-b092-
↪b0dd880c07e1/campaign/bought \
-X "GET" \
-H "Accept:application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization:\ Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

## Open Loyalty

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

**Note:** The `customer = 00000000-0000-474c-b092-b0dd880c07e1` id is an example value. Your value may be different. Check the list of all customers if you are not sure which id should be used.

---

**Note:** When using endpoints starting with `/api/<storeCode>/seller`, you need to authorize using seller account credentials.

---

**Note:** As a seller, you will receive less information about campaigns than an administrator.

---

### Example Response

```
STATUS: 200 OK
```

```
{
  "campaigns": [],
  "total": 0
}
```

### Example Response

```
STATUS: 200 OK
```

```
{
  "campaigns": [
    {
      "purchaseAt": "2018-01-30T18:23:24+0100",
      "costInPoints": 20,
      "campaignId": {
        "campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93"
      },
      "used": false,
      "coupon": {
        "code": "123"
      }
    }
  ],
  "total": 1
}
```



```
"campaignId": "000096cf-32a3-43bd-9034-4df343e5fd93",
"reward": "discount_code",
"name": "tests",
"active": true,
"costInPoints": 20,
"singleCoupon": false,
"unlimited": false,
"limit": 10,
"limitPerUser": 2,
"campaignActivity": {
  "allTimeActive": true
},
"campaignVisibility": {
  "allTimeVisible": true
},
"segmentNames": [],
"levelNames": {
  "000096cf-32a3-43bd-9034-4df343e5fd93": "level0",
  "e82c96cf-32a3-43bd-9034-4df343e5fd94": "level1",
  "000096cf-32a3-43bd-9034-4df343e5fd94": "level2",
  "0f0d346e-9fd0-492a-84aa-2a2b61419c97": "level3"
},
"usageLeft": 0,
"visibleForCustomersCount": 6,
"usersWhoUsedThisCampaignCount": 1
},
"used": false,
"coupon": {
  "code": "123"
}
},
"total": 1
}
```

## 2.1.10 Customer Earning API

These endpoints will allow you to easily viewing active earning rules.

---

**Note:** Each role in the Open Loyalty has individual endpoints to viewing active earning rules.

---

### Return all active earning rules

To view active earning rules, you need to call the `/api/<storeCode>/customer/earningRule` endpoint with the GET method.

#### Definition

```
GET /api/<storeCode>/customer/earningRule
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the earning rules from.

## Example

```
curl http://localhost:8181/api/DEFAULT/customer/earningRule \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXUyJ9..."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXUyJ9...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "earningRules": [
    {
      "levels": [
        "000096cf-32a3-43bd-9034-4df343e5fd93"
      ],
      "segments": [],
      "earningRuleId": "00000000-0000-474c-b092-b0dd880c0121",
      "name": "Facebook like test rule",
      "description": "sth",
      "active": true,
      "startAt": "2018-01-19T09:45:00+0100",
      "endAt": "2018-03-19T09:45:00+0100",
      "allTimeActive": false,
      "usages": [],
      "eventName": "facebook_like",
      "pointsAmount": 100,
      "limit": {},
      "type": "custom_event",
      "usageUrl": "http://backend.openloyalty.test.openloyalty.io/api/v1/earnRule/
↔facebook_like/customer/:customerId",
      "segmentNames": [],
      "levelNames": {
        "000096cf-32a3-43bd-9034-4df343e5fd93": "level0"
      }
    },
    {
      "levels": [
        "000096cf-32a3-43bd-9034-4df343e5fd93"
      ],
      "segments": [],
    }
  ]
}
```

(continues on next page)

```

    "earningRuleId": "7664138c-b5a4-4dcd-80ba-0049a92166db",
    "name": "name",
    "description": "description",
    "active": true,
    "allTimeActive": true,
    "usages": [],
    "eventName": "custom_event_name",
    "pointsAmount": 1,
    "limit": {
      "active": false
    },
    "type": "custom_event",
    "usageUrl": "http://backend.openloyalty.test.openloyalty.io/api/v1/earnRule/
↔custom_event_name/customer/:customerId",
    "segmentNames": [],
    "levelNames": {
      "000096cf-32a3-43bd-9034-4df343e5fd93": "level0"
    }
  },
  {
    "levels": [
      "000096cf-32a3-43bd-9034-4df343e5fd93"
    ],
    "segments": [],
    "earningRuleId": "7d482776-318a-48dd-90cd-6b3f06a3f4e8",
    "name": "sdgsdgsdg",
    "description": "description",
    "active": true,
    "allTimeActive": true,
    "usages": [],
    "eventName": "custom_event_name_1",
    "pointsAmount": 1,
    "limit": {
      "active": false
    },
    "type": "custom_event",
    "usageUrl": "http://backend.openloyalty.test.openloyalty.io/api/v1/earnRule/
↔custom_event_name_1/customer/:customerId",
    "segmentNames": [],
    "levelNames": {
      "000096cf-32a3-43bd-9034-4df343e5fd93": "level0"
    }
  }
],
"currency": "eur"
}

```

### Use a custom event earning rule

To trigger custom event earning rules, you need to call the `/api/<storeCode>/customer/earningRule/<eventName>` endpoint with the POST method.



## Definition

```
POST /api/<storeCode>/customer/earningRule/<eventName>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the earning rule belongs to.
eventName	string	Name of custom event

## Example

```
curl http://localhost:8181/api/DEFAULT/customer/earningRule/customerAttendedEvent \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXUyJ9..."
```

**Note:** The *eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXUyJ9...* authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "points": 12
}
```

## 2.1.11 Customer Level API

These endpoints will allow you to see Levels for a customer.

### Get a complete list of levels

To retrieve a complete list of levels, you need to call the `/api/<storeCode>/customer/level` endpoint with the GET method.

## Definition

```
GET /api/<storeCode>/customer/level
```



## List of all logged-in customer points transfer

To retrieve a list of points transfer by a specific customer, use `/api/<storeCode>/customer/points/transfer` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/customer/points/transfer
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
state	query	Set 1 if always active, otherwise 0
type	query	Current points status: adding or spending
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name, by default = firstName
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/points/transfer \
-X "GET" \
-H "Accept:application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization:\ Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```
{
  "transfers": [
    {
      "pointsTransferId": "e82c96cf-32a3-43bd-9034-4df343e5f211",
      "accountId": "adbdb586-317b-4bed-8cc0-346199064d45",
      "customerId": "00000000-0000-474c-b092-b0dd880c07e1",
      "customerFirstName": "John",
      "customerLastName": "Doe",
      "customerEmail": "user@example.com",
      "customerPhone": "11111",
      "createdAt": "2018-01-21T09:45:05+0100",
      "value": 100,
      "state": "active",
    }
  ]
}
```

(continues on next page)



(continued from previous page)

```
-d "transfer[receiver]=00000000-0000-474c-b092-b0dd880c07f5" \
-d "transfer[points]=11"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "pointsTransferId": "5db67ae4-ddc8-4590-ac2d-0b3e0b8f4c7e"
}
```

## 2.1.13 Earning Rule

These endpoints will allow you to easily manage Earning Rules.

### Get a complete list of earning rules

To retrieve a paginated list of earning rules, you need to call the `/api/<storeCode>/earningRule` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/earningRule
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the earning rules of.
active	query	<i>(optional)</i> Possible values: active, inactive
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC

To see the first page of all earning rules, use the method below:

### Example

```
curl http://localhost:8181/api/DEFAULT/earningRule \
-X "GET" \
-H "Accept: application/json" \
```

(continues on next page)





### Example Response

STATUS: 200 OK

```
{
  "levels": [
    "000096cf-32a3-43bd-9034-4df343e5fd93"
  ],
  "segments": [],
  "earningRuleId": "00000000-0000-474c-b092-b0dd880c0121",
  "name": "Facebook like test rule",
  "description": "sth",
  "active": true,
  "startAt": "2018-01-19T09:45:00+0100",
  "endAt": "2018-03-19T09:45:00+0100",
  "allTimeActive": false,
  "usages": [],
  "eventName": "facebook_like",
  "pointsAmount": 100,
  "limit": {},
  "type": "custom_event",
  "hasPhoto": false,
  "usageUrl": "http://backend.openloyalty.test.openloyalty.io/api/v1/earnRule/
↪facebook_like/customer/:customerId",
  "segmentNames": [],
  "levelNames": {
    "000096cf-32a3-43bd-9034-4df343e5fd93": "level0"
  }
}
```

### Get earning rule details (seller)

To retrieve a page of earning rule details, you need to call the `/api/<storeCode>/seller/earningRule/<earningRule>` endpoint with the GET method.

### Definition

GET `/api/<storeCode>/seller/earningRule/<earningRule>`

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get the earning rule from.
<earningRule>	query	earningRule ID

### Example

To see the earning rule with id `earningRule = 00000000-0000-474c-b092-b0dd880c0725`, use the method below:





## Open Loyalty

---

### Definition

```
POST /api/<storeCode>/earningRule
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to create the earning rule in.
earningRule[type]	request	The type of earning points. Possible types: Custom event rule, Customer Referral, Event Rule, General spending rule, Multiple earned points, Product Purchase, Multiple by product labels
earningRule[rewardType]	request	Who will be rewarded. Possible types: referred,referrer, both
earningRule[name]	request	EarningRule name
earningRule[description]	request	A short description
earningRule[endAt]	request	earningRule visible to YYYY-MM-DD HH:mm, e.g.: 2019-10-05 10:59. <i>(required only if "allTimeActive=0")</i>
earningRule[startAt]	request	earningRule visible from YYYY-MM-DD HH:mm, e.g.: 2017-10-05 10:59. <i>(required only if "allTimeActive=0")</i>
earningRule[active]	request	Set 1 if active, otherwise 0
earningRule[pointsAmount]	request	How many points customer can earn
earningRule[target]	request	Set level to choose target from defined levels. Set segment to choose target from defined segments
earningRule[levels]	request	Array of level IDs. <i>(required only if "target=level")</i>
earningRule[segments]	request	Array of segment IDs. <i>(required only if "target=segment")</i>
earningRule[limit][active]	request	Set 1 if usage limit active, otherwise 0
earningRule[limit][period]	request	Period usage limit. <i>(required only if "[limit][active]=1")</i> Possible parameters: day,week,month,3months,6months,year,forever
earningRule[limit][limit]	request	Usage limit. <i>(required only if "[limit][active]=1")</i>
earningRule[eventName]	request	Custom Event name
earningRule[allTimeActive]	request	Set 1 if always visible, otherwise 0
earningRule[excludeDeliveryCost]	request	Points will not be calculated for delivery cost. Set 1 to active, otherwise 0
earningRule[excludedSKUs]	request	Excluding products with the given SKU
earningRule[minOrderValue]	request	Points will not be calculated for whole purchase if its value will be below value
earningRule[inclusionType]	request	Accepted values: [include_labels, exclude_labels]. If include_labels is set - includedLabels will be used. if exclude_labels is set - excludedLabels will be used.
earningRule[excludedLabels]	request	Points will not be calculated for the purchase of products with defined labels
earningRule[includedLabels]	request	Points will be calculated only for the purchase of products with defined labels
earningRule[multiplier]	request	Points gained for product purchase will be multiplied by this factor
earningRule[labelMultipliers]	request	Points gained for product purchase by labels will be multiplied
earningRule[skuIds][0]	request	Refers to products with the given SKU

### Example

```
curl http://localhost:8181/api/DEFAULT/earningRule \  
-X "POST" \  
-H "Accept: application/json" \  
-H "Content-type: application/x-www-form-urlencoded" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..." \  
-d "earningRule[active]=1" \  
-d "earningRule[type]=event" \  
-d "earningRule[description]=test" \  
-d "earningRule[endAt]=2018-03-19+09:45" \  
-d "earningRule[eventName]=oloy.customer.logged_in" \  
-d "earningRule[levels][0]=000096cf-32a3-43bd-9034-4df343e5fd93" \  
-d "earningRule[name]=nowy+rule" \  
-d "earningRule[pointsAmount]=5" \  
-d "earningRule[segments]=[+]" \  
-d "earningRule[startAt]=2019-03-19+09:45" \  
-d "earningRule[target]=level" \  
-d "earningRule[limit][active]=1" \  
-d "earningRule[limit][period]=month" \  
-d "earningRule[limit][limit]=5" \  
-d "earningRule[rewardType]=both" \  
-d "earningRule[allTimeActive]=0" \  
-d "earningRule[excludeDeliveryCost]=true" \  
-d "earningRule[excludedSKUs]=123" \  
-d "earningRule[minOrderValue]=2" \  
-d "earningRule[inclusionType]=exclude_labels" \  
-d "earningRule[excludedLabels]=1:1" \  
-d "earningRule[multiplier]=2" \  
-d "earningRule[skuIds][0]=SKU123"
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

---

**Note:** The `000096cf-32a3-43bd-9034-4df343e5fd93` or `00000000-0000-474c-b092-b0dd880c0121` id are example values. Your value may be different. Check the list of all levels if you are not sure which id should be used.

---

### Example Response

```
STATUS: 200 OK
```

### Edit an existing earning rule

To edit an existing earning rule, you need to call the `/api/<storeCode>/earningRule/<earningRule>` endpoint with the PUT method.

## Definition

```
PUT /api/<storeCode>/earningRule/<earningRule>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the updated earning rule belongs to.
<earningRule>	query	EarningRule ID
earningRule[type]	request	The type of earning points. Possible types: Custom event rule, Customer Referral, Event Rule, General spending rule, Multiple earned points, Product Purchase, Multiple by product labels
earningRule[rewardType]	request	Who will be rewarded. Possible types: referred,referrer, both
earningRule[name]	request	Earning Rule name
earningRule[description]	request	A short description
earningRule[endAt]	request	earningRule visible to YYYY-MM-DD HH:mm, e.g.: 2019-10-05 10:59. (required only if "allTimeActive=0")
earningRule[startAt]	request	earningRule visible from YYYY-MM-DD HH:mm, e.g.: 2017-10-05 10:59. (required only if "allTimeActive=0")
earningRule[active]	request	Set 1 if active, otherwise 0
earningRule[pointsAmount]	request	How many points customer can earn
earningRule[target]	request	Set level to choose target from defined levels. Set segment to choose target from defined segments
earningRule[levels]	request	Array of level IDs. (required only if "target=level")
earningRule[segments]	request	Array of segment IDs. (required only if "target=segment")
earningRule[limit][active]	request	Set 1 if usage limit active, otherwise 0
earningRule[limit][period]	request	Period usage limit. (required only if "[limit][active]=1")
earningRule[limit][limit]	request	Usage limit. (required only if "[limit][active]=1")
earningRule[eventName]	request	Custom Event name
earningRule[allTimeActive]	request	Set 1 if always visible, otherwise 0
earningRule[excludeDeliveryCost]	request	Points will not be calculated for delivery cost. Set 1 to active, otherwise 0
earningRule[excludedSKUs]	request	Excluding products with the given SKU
earningRule[minOrderValue]	request	Points will not be calculated for whole purchase if its value will be below value
earningRule[inclusionType]	request	Accepted values: [include_labels, exclude_labels]. If include_labels is set - includedLabels will be used. if exclude_labels is set - excludedLabels will be used.
earningRule[excludedLabels]	request	Points will not be calculated for the purchase of products with defined labels
earningRule[includedLabels]	request	Points will be calculated only for the purchase of products with defined labels
earningRule[multiplier]	request	Points gained for product purchase will be multiplied by this factor
162 earningRule[labelMultipliers]	request	Points gained for product purchase by label will be multiplied
earningRule[skuIds][0]	request	Refers to products with the given SKU



```
{
  "earningRuleId": "00000000-0000-474c-b092-b0dd880c0121"
}
```

### Change earning rule status

To make an earning rule active or inactive, you need to call the `/api/<storeCode>/earningRule/<earningRule>/activate` endpoint with the POST method.

#### Definition

```
POST /api/<storeCode>/earningRule/<earningRule>/activate
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the updated earning rule belongs to.
<earningRule>	query	earningRule ID
active	request	Possible values: active, inactive

#### Example

To make the earning rule `earningRule = 7d482776-318a-48dd-90cd-6b3f06a3f4e8` active, use the method below:

```
curl http://localhost:8181/api/DEFAULT/earningRule/7d482776-318a-48dd-90cd-
↳6b3f06a3f4e8/active \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
  -d "active=1"
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

---

**Note:** The `earningRule = 7d482776-318a-48dd-90cd-6b3f06a3f4e8` id is an example value. Your value may be different. Check the list of all earningRules if you are not sure which id should be used.

---

#### Example Response

```
STATUS: 204 No Content
```





## Open Loyalty

---

```
STATUS: 404 Not Found
```

```
{
  "error": {
    "code": 404,
    "message": "Not Found"
  }
}
```

### Remove an earning rule's photo

To remove an earning rule's photo, you need to call the `/api/<storeCode>/earningRule/<earningRule>/photo` endpoint with the DELETE method.

#### Definition

```
DELETE /api/<storeCode>/earningRule/<earningRule>/photo
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the earning rule belongs to.
<earningRule>	query	Earning rule ID

#### Example

To remove the photo for earning rule `earningRule = 000096cf-32a3-43bd-9034-4df343e5fd93`, use the method below:

```
curl http://localhost:8181/api/DEFAULT/earningRule/000096cf-32a3-43bd-9034-4df343e5fd93/photo \
  -X "DELETE" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

---

**Note:** The `earningRule = 000096cf-32a3-43bd-9034-4df343e5fd93` id is an example value. Your value may be different. Check the list of all earning rules if you are not sure which id should be used.

---

#### Example Response

```
STATUS: 200 OK
```

## Add a photo to an earning rule

To add a photo to an earning rule, you need to call the `/api/<storeCode>/earningRule/<earningRule>/photo` endpoint with the `POST` method.

### Definition

```
POST /api/<storeCode>/earningRule/<earningRule>/photo
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the earning rule belongs to.
<earningRule>	query	Earning rule ID
photo[file]	request	Absolute path to the photo

### Example

To add a photo to earning rule `earningRule = 000096cf-32a3-43bd-9034-4df343e5fd93`, use the method below:

```
curl http://localhost:8181/api/DEFAULT/earningRule/000096cf-32a3-43bd-9034-4df343e5fd93/photo \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
  -d "photo[file]=C:\fakepath\Photo.png"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `earningRule = 000096cf-32a3-43bd-9034-4df343e5fd93` id is an example value. Your value may be different. Check the list of all earning rules if you are not sure which id should be used.

**Note:** The `photo[file]=C:\fakepath\Photo.png` is an example value. Your value may be different.

### Example Response

```
STATUS: 200 OK
```

### QR code

This method allows calculating points using QR codes. You need to call the `/api/<storeCode>/earningRule/qrcode/customer/<customer>` endpoint with the `POST` method.

## Open Loyalty

---

### Definition

```
POST /api/<storeCode>/earningRule/qrcode/customer/<customer>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the earning rule and customer belong to.
<customer>	query	Customer ID
earningRule[code]	request	QR code
earningRule[earningRuleId]	request	(optional) UUID of the earning rule. If specified, only this one rule will be executed. If omitted, all rules applicable to the customer will be executed

### Example Response

```
curl http://localhost:8181/api/DEFAULT/earningRule/qrcode/customer/00000000-0000-474c-b092-b0dd880c07e1 \
-X "POST" \
-H "Accept: application/json" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "earningRule[code]=abccode" \
-d "earningRule[earningRuleId]=e378c813-2116-448a-b125-564cef15f932"
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

---

**Note:** The `00000000-0000-474c-b092-b0dd880c07e1` customer UUID, `e378c813-2116-448a-b125-564cef15f932` earning rule UUID, `abccode` qr code are example values. Your values can be different.

---

```
STATUS: 200 OK
```

```
{
  "points": 10
}
```

### Geolocation

To calculate points using geolocation, you need to call the `/api/<storeCode>/earningRule/geolocation/customer/<customer>` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/earningRule/geolocation/customer/<customer>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the earning rule and the customer belong to.
<customer>	query	Customer ID
earningRule[latitude]	body	Current customer's latitude. Positive and negative values can be used.
earningRule[longitude]	body	Current customer's longitude. Positive and negative values can be used.
earningRule[earningRuleId]	query	(optional) UUID of the earning rule. If specified, only this one geo rule will be executed. If omitted, all rules applicable to the customer will be executed

### Example Response

```
curl http://localhost:8181/api/DEFAULT/earningRule/geolocation/customer/00000000-0000-474c-b092-b0dd880c07e1 \
-X "POST" \
-H "Accept: application/json" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "earningRule[latitude]=52.052240"
-d "earningRule[longitude]=-21.046587"
-d "earningRule[earningRuleId]=51283523-0760-474b-8c08-4ccd2b3a0f41"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Note:** The `00000000-0000-474c-b092-b0dd880c07e1` customer UUID, `83fe084b-3682-4ddb-bc10-c3c2373dfbcc` earning rule UUID, `52.052240`, `-21.046587` coordinates are example values. Your values can be different.

```
STATUS: 200 OK
```

```
{
  "points": 1
}
```

### 2.1.14 Invitation

These endpoints will allow you to easily manage Invitations.

#### Get a complete list of invitations

To retrieve a paginated list of invitations, you need to call the `/api/<storeCode>/invitations` endpoint with the GET method.



(continued from previous page)

```
"referrerName": "John Doe",
"recipientEmail": "test3@example.com",
"status": "invited",
"token": "575c0f0435d0970853b25b967378c4155c8c0843"
},
{
  "referrerId": "00000000-0000-474c-b092-b0dd880c07e1",
  "recipientId": "",
  "invitationId": "22200000-0000-474c-b092-b0dd880c07e1",
  "referrerEmail": "user@example.com",
  "referrerName": "John Doe",
  "recipientEmail": "test1@example.com",
  "status": "invited",
  "token": "ebea0309e2ca40f45b11537694270df8fc7161b7"
},
{
  "referrerId": "00000000-0000-474c-b092-b0dd880c07e1",
  "recipientId": "",
  "invitationId": "22200000-0000-474c-b092-b0dd880c07e4",
  "referrerEmail": "user@example.com",
  "referrerName": "John Doe",
  "recipientEmail": "test4@example.com",
  "status": "invited",
  "token": "1042654f4acd5099f54286acbb10d668173a95d0"
}
],
"total": 4
}
```

## 2.1.15 Points transfers

These endpoints will allow you to easily manage Points transfers.

### Get a complete list of Points transfers

To retrieve a paginated list of Points transfers, you need to call the `/api/<storeCode>/points/transfer` endpoint with the GET method.

#### Definition

```
GET /api/<storeCode>/points/transfer
```





(continued from previous page)

```

    "comment": "Event - First Purchase - 10",
    "issuer": "system"
  },
  {
    "pointsTransferId": "cd470d77-a08e-4c62-9f47-da180524f683",
    "accountId": "49a218ab-71ba-4f7f-8f4e-407df5f84b11",
    "customerId": "00000000-0000-474c-b092-b0dd880c07e1",
    "customerFirstName": "John",
    "customerLastName": "Doe",
    "customerLoyaltyCardNumber": "47834433524",
    "customerEmail": "user@example.com",
    "customerPhone": "+48234234000",
    "createdAt": "2018-09-13T16:37:33+0200",
    "expiresAt": "2018-10-13T16:37:33+0200",
    "value": 6.9,
    "state": "active",
    "type": "adding",
    "transactionId": {
      "transactionId": "00000000-0000-1111-0000-000000000003"
    },
    "comment": "General spending rule - 2.3",
    "issuer": "system",
    "transactionDocumentNumber": "456",
    "transaction": {
      "grossValue": 3,
      "labels": [
        {
          "key": "first label",
          "value": "first test"
        },
        {
          "key": "second label",
          "value": "second test"
        }
      ]
    },
    "items": [
      {
        "sku": {
          "code": "SKU1"
        },
        "name": "item 1",
        "quantity": 1,
        "grossValue": 1,
        "category": "aaa",
        "labels": [
          {
            "key": "test",
            "value": "label"
          },
          {
            "key": "test",
            "value": "label2"
          }
        ]
      },
      {
        "maker": "sss"
      }
    ]
  }
}

```

(continues on next page)

(continued from previous page)

```

        "sku": {
            "code": "SKU2"
        },
        "name": "item 2",
        "quantity": 2,
        "grossValue": 2,
        "category": "bbb",
        "labels": [],
        "maker": "ccc"
    }
}
]
}
},
"total": 2
}

```

### Get a complete list of points transfers (customer)

To retrieve a paginated list of Points transfers, you need to call the `/api/<storeCode>/customer/points/transfer` endpoint with the GET method.

#### Definition

```
GET /api/<storeCode>/customer/points/transfer
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
state	query	(optional) Possible values: active, expired, pending
type	query	(optional) Possible values: adding, spending
page	query	(optional) Start from page, by default 1
perPage	query	(optional) Number of items to display per page, by default = 10
sort	query	(optional) Sort by column name
direction	query	(optional) Direction of sorting [ASC, DESC], by default = ASC

#### Example

```

curl http://localhost:8181/api/DEFAULT/customer/points/transfer \
-X "GET" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. "

```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

STATUS: 200 OK

```

{
  "transfers": [
    {
      "pointsTransferId": "ae1c49b1-02ab-4626-982d-71b544b01136",
      "accountId": "49a218ab-71ba-4f7f-8f4e-407df5f84b11",
      "customerId": "00000000-0000-474c-b092-b0dd880c07e1",
      "customerFirstName": "John",
      "customerLastName": "Doe",
      "customerLoyaltyCardNumber": "47834433524",
      "customerEmail": "user@example.com",
      "customerPhone": "+48234234000",
      "createdAt": "2018-09-13T16:37:33+0200",
      "expiresAt": "2018-10-13T16:37:33+0200",
      "value": 10,
      "state": "active",
      "type": "adding",
      "comment": "Event - First Purchase - 10",
      "issuer": "system"
    },
    {
      "pointsTransferId": "cd470d77-a08e-4c62-9f47-da180524f683",
      "accountId": "49a218ab-71ba-4f7f-8f4e-407df5f84b11",
      "customerId": "00000000-0000-474c-b092-b0dd880c07e1",
      "customerFirstName": "John",
      "customerLastName": "Doe",
      "customerLoyaltyCardNumber": "47834433524",
      "customerEmail": "user@example.com",
      "customerPhone": "+48234234000",
      "createdAt": "2018-09-13T16:37:33+0200",
      "expiresAt": "2018-10-13T16:37:33+0200",
      "value": 6.9,
      "state": "active",
      "type": "adding",
      "transactionId": {
        "transactionId": "00000000-0000-1111-0000-000000000003"
      },
      "comment": "General spending rule - 2.3",
      "issuer": "system",
      "transactionDocumentNumber": "456",
      "transaction": {
        "grossValue": 3,
        "labels": [
          {
            "key": "first label",
            "value": "first test"
          },
          {
            "key": "second label",
            "value": "second test"
          }
        ],
        "items": [
          {

```

(continues on next page)

```

        "sku": {
            "code": "SKU1"
        },
        "name": "item 1",
        "quantity": 1,
        "grossValue": 1,
        "category": "aaa",
        "labels": [
            {
                "key": "test",
                "value": "label"
            },
            {
                "key": "test",
                "value": "label2"
            }
        ],
        "maker": "sss"
    },
    {
        "sku": {
            "code": "SKU2"
        },
        "name": "item 2",
        "quantity": 2,
        "grossValue": 2,
        "category": "bbb",
        "labels": [],
        "maker": "ccc"
    }
]
}
}
},
"total": 2
}

```

### Add points to customer's account

To add points, you need to call the `/api/<storeCode>/points/transfer/add` endpoint with the POST method.

#### Definition

```
POST /api/<storeCode>/points/transfer/add
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
transfer[customer]	query	Customer ID
transfer[points]	query	How many points customer can get
transfer[comment]	query	<i>(optional)</i> Comment

## Example

```
curl http://localhost:8181/api/DEFAULT/points/transfer/add \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..\" \
  -d "transfer[customer]=b9af6a8c-9cc5-4924-989c-e4af614ab2a3" \
  -d "transfer[points]=9"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "pointsTransferId": "32132863-3d1e-4a94-8bb4-6e42e3c96c0b"
}
```

## Spend customer's points

To spend a customer's points, you need to call the `/api/<storeCode>/points/transfer/spend` endpoint with the POST method.

## Definition

```
POST /api/<storeCode>/points/transfer/spend
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
transfer[customer]	query	Customer ID
transfer[points]	query	How many points customer can spend
transfer[comment]	query	<i>(optional)</i> Comment

## Example

```
curl http://localhost:8181/api/DEFAULT/points/transfer/spend \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..\" \
  -d "transfer[customer]=b9af6a8c-9cc5-4924-989c-e4af614ab2a3" \
  -d "transfer[points]=1"
```

## Open Loyalty

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...`  authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```
{
  "pointsTransferId": "b97a31fe-9bc9-4fff-a467-487f2c316371"
}
```

### Transfer points between customers (admin)

To transfer points between customers, you need to call the `/api/<storeCode>/admin/p2p-points-transfer` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/admin/p2p-points-transfer
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store both customers belong to.
transfer[sender]	query	email/phone or uuid of customer from whom points will be transferred
transfer[receiver]	query	email/phone or uuid of customer who will get points
transfer[points]	query	How many points will be transferred

### Example

```
curl http://localhost:8181/api/DEFAULT/admin/p2p-points-transfer \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..." \
-d "transfer[sender]=b9af6a8c-9cc5-4924-989c-e4af614ab2a3" \
-d "transfer[receiver]=b9af6a8c-9cc5-4924-989c-e4af614ab3c5" \
-d "transfer[points]=100"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...`  authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

## Example Response

```
STATUS: 200 OK
```

```
{
  "pointsTransferId": "b97a31fe-9bc9-4fff-a467-487f2c316371"
}
```

**Note:** Returned pointsTransferId is a UUID of the P2P spend points transfer created.

## Transfer points between customers (customer)

To transfer points between a logged-in customer and another customer, you need to call the `/api/<storeCode>/customer/points/p2p-transfer` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/customer/points/p2p-transfer
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store both customers belong to.
transfer[receiver]	query	email/phone or uuid of customer who will get points
transfer[points]	query	How many points will be transferred

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/points/p2p-transfer \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "transfer[receiver]=b9af6a8c-9cc5-4924-989c-e4af614ab3c5" \
-d "transfer[points]=100"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

## Open Loyalty

---

```
{
  "pointsTransferId": "b97a31fe-9bc9-4fff-a467-487f2c316371"
}
```

---

**Note:** Returned pointsTransferId is a UUID of the P2P spend points transfer created.

---

### Cancel specific points transfer

To cancel a specific points transfer, you need to call the `/api/<storeCode>/points/transfer/<transfer>/cancel` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/points/transfer/<transfer>/cancel
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the point transfer was made in.
<transfer>	query	Points transfer ID

### Example

```
curl http://localhost:8181/api/DEFAULT/points/transfer/313cf0c1-5376-4f66-9de3-
↪77943760423a/cancel \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```
(no content)
```

### Activate specific points transfer

To activate a specific points transfer, you need to call the `/api/<storeCode>/points/transfer/<transfer>/activate` endpoint with the POST method.

### Requirements:



- transfer MUST HAVE state PENDING;

**Results:**

- transfer will have state ACTIVE and points will be unlocked;

**Definition**

```
POST /api/<storeCode>/points/transfer/<transfer>/activate
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the point transfer was made in.
<transfer>	query	Points transfer ID

**Example**

```
curl http://localhost:8181/api/DEFAULT/points/transfer/313cf0c1-5376-4f66-9de3-
↪77943760423a/activate \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The *eyJhbGciOiJSUzI1NiIsInR5cCI6I..* authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

**Example Response**

```
STATUS: 204 No Content
```

```
(no content)
```

**Expire specific points transfer**

To expire a specific points transfer, you need to call the `/api/<storeCode>/points/transfer/<transfer>/expire` endpoint with the POST method.

**Requirements:**

- transfer MUST HAVE state ACTIVE;

**Results:**

- transfer will have state EXPIRED and current date in `expiredAt`;





### Example

```
curl http://localhost:8181/api/DEFAULT/points/transfers \  
-X "GET" \  
-H "Accept: application/json" \  
-H "Content-type: application/x-www-form-urlencoded" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IiI6..."
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6IiI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```
{  
  "2018-01-06": 0,  
  "2018-01-07": 0,  
  "2018-01-08": 0,  
  "2018-01-09": 5,  
  "2018-01-10": 0,  
  "2018-01-11": 5,  
  "2018-01-12": 4,  
  "2018-01-13": 3,  
  "2018-01-14": 0,  
  "2018-01-15": 0,  
  "2018-01-16": 3,  
  "2018-01-17": 0,  
  "2018-01-18": 5,  
  "2018-01-19": 0,  
  "2018-01-20": 6,  
  "2018-01-21": 5,  
  "2018-01-22": 0,  
  "2018-01-23": 6,  
  "2018-01-24": 0,  
  "2018-01-25": 0,  
  "2018-01-26": 0,  
  "2018-01-27": 0,  
  "2018-01-28": 5,  
  "2018-01-29": 0,  
  "2018-01-30": 0,  
  "2018-01-31": 0,  
  "2018-02-01": 0,  
  "2018-02-02": 5,  
  "2018-02-03": 0,  
  "2018-02-04": 0  
}
```

### Block points on customer's account

The administrator can block points on a customer's account in order to prevent them from spending them. To block points, you need to call the `/api/<storeCode>/points/transfer/block` endpoint with the POST method.

In order to unblock points, you need to use the `/api/<storeCode>/points/transfer/<transfer>/cancel` endpoint.

## Definition

```
POST /api/<storeCode>/points/transfer/block
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
transfer[customer]	query	Customer ID
transfer[points]	query	How many points to block
transfer[comment]	query	<i>(optional)</i> Comment

## Example

```
curl http://localhost:8181/api/DEFAULT/points/transfer/block \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "transfer[customer]=b9af6a8c-9cc5-4924-989c-e4af614ab2a3" \
-d "transfer[points]=9"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "pointsTransferId": "32132863-3d1e-4a94-8bb4-6e42e3c96c0b"
}
```

## 2.1.16 Security

These endpoints will allow you to easily manage password and token-related matters.

### Password reset request (customer)

Invoking this method will send a message to the user with a password reset URL. You need to call the `/api/<storeCode>/customer/password/reset/request` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/customer/password/reset/request
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
username	string	Customer's e-mail address

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/password/reset/request \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "username=user@example.com"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```
{
  "success": true
}
```

### Set new password after requesting a new password

To reset the password for a customer who requested a new password,, you need to call the `/api/password/reset` endpoint with the POST method.

### Definition

```
POST /api/password/reset
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
token	query	Token received during resetting the password
reset[plainPassword]	query	New password







---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---



---

**Note:** Your password must be at least 8 characters long. Your password must include both upper and lower case letters. Your password must include at least one number. Your password must contain at least one special character.

---

### Example Response

```
STATUS: 200 OK
```

```
{
  "success": true
}
```

### Password reset request (admin)

Invoking this method will send a message to the admin user's email with the password reset URL. You need to call the `/api/password/reset/request` endpoint with the `POST` method.

### Definition

```
POST /api/password/reset/request
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
username	query	User name who recovers the password

### Example

```
curl http://localhost:8181/api/password/reset/request \
  -X "POST" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii.." \
  -d "username=admin"
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

## Open Loyalty

---

### Example Response

```
STATUS: 200 OK
```

```
{  
  "success": true  
}
```

### Log out current user

To log out the current user, you need to call the `/api/token/revoke` endpoint with the GET method.

### Definition

```
GET /api/token/revoke
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication

### Example

```
curl http://localhost:8181/api/token/revoke \  
-X "GET" \  
-H "Accept: application/json" \  
-H "Content-type: application/x-www-form-urlencoded" \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```
(no content)
```

## 2.1.17 Store API

These endpoints will allow you to see the list of stores in Open Loyalty.

### Get store details

To retrieve the details of a store, you need to call the `/api/store/<store>` endpoint with the GET method.





## Example

```
curl http://localhost:8181/api/store \
  -X "GET" \
    -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

## Create a new store

To create a new store, you need to call the `/api/store` endpoint with the `POST` method.

## Definition

```
POST /api/store
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
store[code]	query	Store code
store[name]	query	Store name
store[currency]	query	Store currency
store[active]	query	Store activity, possible values: active/inactive

To create a new store, use the method below:

## Example

```
curl http://localhost:8181/api/store \
  -X "POST" \
    -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I..\" \
    -d "store[active] = 1" \
  -d "store[name] = store_name" \
  -d "store[currency] = EUR" \
  -d "store[code] = store_code"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

## 2.1.18 Transactions

These endpoints will allow you to easily manage transactions.

### Import transactions

To import an XML file with transactions, you need to call the `/api/<storeCode>/admin/transaction/import` endpoint with the `POST` method.

### Definition

```
POST /api/<storeCode>/admin/transaction/import
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to import transactions into.
file[file]	query	XML file with transactions

### Example

```
curl http://localhost:8181/api/DEFAULT/admin/transaction/import \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..." \
-d "file[file]=C:\\fakepath\\transaction.xml"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```

{
  "items": [
    {
      "status": "error",
      "message": "Convert exception: Value \"00000000-0000-474c-1111-b0dd880c07e\" is_
↪not a valid UUID.",
      "identifier": "0001_pos2_zzleID"
    },
    {
      "status": "success",
      "processImportResult": {
        "object": {
          "transactionId": "98b15ef5-94ad-43ef-9984-0d41197d14e6"
        }
      },
      "identifier": "id_bez_tymrazem"
    }
  ],
  "totalProcessed": 2,
  "totalSuccess": 1,
  "totalFailed": 1
}

```

## Match transactions with the customers by importing a XML file

In order to match many transactions to many customers using an XML file, you need to call the `api/<storeCode>/admin/transaction/customer/assign/import` endpoint with the POST method.

### Definition

```
POST api/<storeCode>/admin/transaction/customer/assign/import
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to match transactions in.
file[file]	query	XML file with transactions

### Example

```

curl http://localhost:8181/api/DEFAULT/admin/transaction/customer/assign/import \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "file[file]=C:\\fakepath\\match-customer.xml"

```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Open Loyalty

---

### Example XML

---

**Note:** Only one customer\* field is required (customerId, customerEmail, customerPhoneNumber, customerLoyaltyCardNumber).

---

### Example Response

STATUS: 200 OK

```
{
  "items": [
    {
      "status": "error",
      "message": "(match_customer-2019-11-08_1005-5dc52fe92bc59.xml) Processing_
↳exception: Customer is already assigned to this transaction",
      "identifier": "123"
    }
  ],
  "totalProcessed": 1,
  "totalSuccess": 0,
  "totalFailed": 1
}
```

### Assign a customer to a specific transaction (admin)

To assign a customer to a specific transaction, you need to call the `/api/<storeCode>/admin/transaction/customer/assign` endpoint with the POST method.

### Definition

POST `/api/<storeCode>/admin/transaction/customer/assign`

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer and the transaction belong to.
as-sign[transactionDocumentNumber]	query	Transaction Document Number
assign[customerId]	query	Customer ID
as-sign[customerLoyaltyCardNumber]	query	Customer Loyalty Number
assign[customerPhoneNumber]	query	Customer Phone Number





### Definition

```
POST /api/<storeCode>/customer/transaction/customer/assign
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer and the transaction belong to.
as-sign[transactionDocumentNumber]	query	Transaction Document Number
assign[customerId]	query	Customer ID
as-sign[customerLoyaltyCardNumber]	query	Customer Loyalty Number
assign[customerPhoneNumber]	query	Customer Phone Number

---

**Note:** If you are using the auto-generated docs, you may see there are other fields in the assign[] object. They are ignored in this endpoint. Do not use them in your application as they will be removed in a future version.

---

### Example

```
curl http://localhost:8181/api/DEFAULT/customer/transaction/customer/assign \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. ." \
-d "assign[transactionDocumentNumber]=888"
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I.. .` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```
{
  "transactionId": "9f805211-9326-4b47-b5a6-8155d6ae9d2c"
}
```

### Assign a customer to specific transaction (seller)

To assign a customer to a specific transaction, you need to call the `/api/<storeCode>/pos/transaction/customer/assign` endpoint with the POST method.

## Definition

```
POST /api/<storeCode>/pos/transaction/customer/assign
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer and the transaction belong to.
as-sign[transactionDocumentNumber]	query	Transaction Document Number
assign[customerId]	query	Customer ID
as-sign[customerLoyaltyCardNumber]	query	Customer Loyalty Number
assign[customerPhoneNumber]	query	Customer Phone Number

## Example

```
curl http://localhost:8181/api/DEFAULT/pos/transaction/customer/assign \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. ." \
-d "assign[transactionDocumentNumber]=123" \
-d "assign[customerId]=57524216-c059-405a-b951-3ab5c49bae14" \
-d "assign[customerLoyaltyCardNumber]=333" \
-d "assign[customerPhoneNumber]=333333"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I.. .` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "transactionId": "00000000-0000-1111-0000-000000000005"
}
```

## Get a list of transactions (customer)

To retrieve a complete or filtered list of all transactions a customer has access to, you need to call the `/api/<storeCode>/customer/transaction` endpoint with the GET method.



(continued from previous page)

```
"email": "user@example.com",
"name": "Jan Nowak",
"nip": "aaa",
"phone": "123",
"loyaltyCardNumber": "sa2222",
"address": {
  "street": "Bagno",
  "address1": "12",
  "province": "Mazowieckie",
  "city": "Warszawa",
  "postal": "00-800",
  "country": "PL"
},
"labels": [
  {
    "key": "scan_id",
    "value": "123"
  }
],
"items": [
  {
    "sku": {
      "code": "SKU1"
    },
    "name": "item 1",
    "quantity": 1,
    "grossValue": 1,
    "category": "aaa",
    "maker": "sss",
    "labels": [
      {
        "key": "test",
        "value": "label"
      },
      {
        "key": "test",
        "value": "label2"
      }
    ]
  },
  {
    "sku": {
      "code": "SKU2"
    },
    "name": "item 2",
    "quantity": 2,
    "grossValue": 2,
    "category": "bbb",
    "maker": "ccc",
    "labels": []
  }
],
"currency": "eur",
"pointsEarned": 6.9
},
{
```

(continues on next page)

(continued from previous page)

```
"grossValue": 3,
"transactionId": "00000000-0000-1111-0000-000000000005",
"documentNumber": "888",
"purchaseDate": "2018-02-20T09:45:04+0100",
"purchasePlace": "wroclaw",
"documentType": "sell",
"customerId": "57524216-c059-405a-b951-3ab5c49bae14",
"assignedToCustomerDate": "1970-01-01T01:00:00+01:00",
"customerData": {
  "email": "o@lo.com",
  "name": "Jan Nowak",
  "nip": "aaa",
  "phone": "123",
  "loyaltyCardNumber": "sa21as222",
  "address": {
    "street": "Bagno",
    "address1": "12",
    "province": "Mazowieckie",
    "city": "Warszawa",
    "postal": "00-800",
    "country": "PL"
  }
},
"labels": [
  {
    "key": "scan_id",
    "value": "343"
  }
],
"items": [
  {
    "sku": {
      "code": "SKU1"
    },
    "name": "item 1",
    "quantity": 1,
    "grossValue": 1,
    "category": "aaa",
    "maker": "sss",
    "labels": [
      {
        "key": "test",
        "value": "label"
      },
      {
        "key": "test",
        "value": "label2"
      }
    ]
  },
  {
    "sku": {
      "code": "SKU2"
    },
    "name": "item 2",
    "quantity": 2,
    "grossValue": 2,
```

(continues on next page)



(continued from previous page)

```
"documentNumber": "456",
"purchaseDate": "2018-02-20T09:45:04+0100",
"purchasePlace": "wroclaw",
"documentType": "sell",
"customerId": "00000000-0000-474c-b092-b0dd880c07e1",
"assignedToCustomerDate": "1970-01-01T01:00:00+01:00",
"customerData": {
  "email": "user@example.com",
  "name": "Jan Nowak",
  "nip": "aaa",
  "phone": "123",
  "loyaltyCardNumber": "sa2222",
  "address": {
    "street": "Bagno",
    "address1": "12",
    "province": "Mazowieckie",
    "city": "Warszawa",
    "postal": "00-800",
    "country": "PL"
  }
},
"labels": [
  {
    "key": "scan_id",
    "value": "123"
  }
],
"items": [
  {
    "sku": {
      "code": "SKU1"
    },
    "name": "item 1",
    "quantity": 1,
    "grossValue": 1,
    "category": "aaa",
    "maker": "sss",
    "labels": [
      {
        "key": "test",
        "value": "label1"
      },
      {
        "key": "test",
        "value": "label2"
      }
    ]
  },
  {
    "sku": {
      "code": "SKU2"
    },
    "name": "item 2",
    "quantity": 2,
    "grossValue": 2,
    "category": "bbb",
    "maker": "ccc",
```

(continues on next page)



(continued from previous page)

```

    "labels": []
  }
],
"currency": "eur",
"pointsEarned": 6.9
}

```

## Get customer's transactions (seller)

To retrieve a list of customer transactions, you need to call the `/api/<storeCode>/seller/transaction/customer/<customer>` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/seller/transaction/customer/<customer>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the customer belongs to.
<customer>	query	Customer ID
documentNumber	query	<i>(optional)</i> Filter by Document Number
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC

### Example

```

curl http://localhost:8181/api/DEFAULT/seller/transaction/customer/4b32a723-9923-46fc-
↪a2bc-d09767e5e59b \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. "

```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```

{
  "transactions": [
    {

```

(continues on next page)

```
"grossValue": 2200,
"transactionId": "c13e4e89-2e9a-482d-8ab0-41a8eb9927ed",
"documentNumber": "214124124130",
"purchaseDate": "2017-08-23T00:00:00+0200",
"documentType": "return",
"customerId": "4b32a723-9923-46fc-a2bc-d09767e5e59b",
"assignedToCustomerDate": "1970-01-01T01:00:00+01:00",
"customerData": {
  "email": "tomasztest8@wp.pl",
  "name": "Firstname+Lastname",
  "nip": "000000000000000",
  "phone": "000000000000000",
  "loyaltyCardNumber": "11111111111",
  "address": {
    "street": "Street+name",
    "address1": "123",
    "province": "Dolnoslaskie",
    "city": "Wroclaw",
    "postal": "00-000",
    "country": "PL"
  }
},
"labels": [
  {
    "key": "scan_id",
    "value": "333"
  }
],
"items": [
  {
    "sku": {
      "code": "test0101"
    },
    "name": "Product+name",
    "quantity": 1,
    "grossValue": 2200,
    "category": "Category+Name",
    "maker": "Marker+name",
    "labels": [
      {
        "key": "Label+key",
        "value": "Label+value"
      }
    ]
  }
],
"excludedLevelCategories": [
  "category_excluded_from_level"
],
"currency": "eur"
}
],
"total": 1
}
```



```
"phone": "000000000000000",
"loyaltyCardNumber": "11111111111",
"address": {
  "street": "Street+name",
  "address1": "123",
  "province": "Dolnoslaskie",
  "city": "Wroclaw",
  "postal": "00-000",
  "country": "PL"
},
"labels": [
  {
    "key": "scan_id",
    "value": "123"
  }
],
"items": [
  {
    "sku": {
      "code": "test0101"
    },
    "name": "Product+name",
    "quantity": 1,
    "grossValue": 1500,
    "category": "Category+Name",
    "maker": "Marker+name",
    "labels": [
      {
        "key": "Label+key",
        "value": "Label+value"
      }
    ]
  }
],
"excludedLevelCategories": [
  "category_excluded_from_level"
],
"currency": "eur"
}
],
"total": 1
}
```

## Get a list of transactions

To retrieve a complete or filtered list of transactions, you need to call the `/api/<storeCode>/transaction` endpoint with the GET method.

### Definition

```
GET /api/<storeCode>/transaction
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to get transactions from.
customer-Data_loyaltyCardNumber	query	<i>(optional)</i> Loyalty Card Number
customer-Data_name	query	<i>(optional)</i> Customer Name
customer-Data_email	query	<i>(optional)</i> Customer Email
customer-Data_phone	query	<i>(optional)</i> Customer Phone
customerId	query	<i>(optional)</i> Customer ID
documentType	query	<i>(optional)</i> Document Type
documentNumber	query	<i>(optional)</i> Document Number
posId	query	<i>(optional)</i> POS ID
purchaseDateFrom	query	<i>(optional)</i> purchase date's lower limit
purchaseDateTo	query	<i>(optional)</i> purchase date's upper limit
grossValueFrom	query	<i>(optional)</i> transaction gross value lower limit
grossValueTo	query	<i>(optional)</i> transaction gross value upper limit
page	query	<i>(optional)</i> Start from page, by default 1
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC
labels	query	<i>(optional)</i> Filter transactions by labels. Format "labels[0][key]=label_key & labels[0][value]=first_value & labels[1][key]=another_key"

### Example

```
curl http://localhost:8181/api/DEFAULT/transaction \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IiE6Ij09"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6IiE6Ij09` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```
{
  "transactions": [
    {
      "grossValue": 3,
      "transactionId": "00000000-0000-1111-0000-000000000003",
    }
  ]
}
```

(continues on next page)

```
"documentNumber": "456",
"purchaseDate": "2018-02-20T09:45:04+0100",
"purchasePlace": "wroclaw",
"documentType": "sell",
"customerId": "00000000-0000-474c-b092-b0dd880c07e1",
"assignedToCustomerDate": "1970-01-01T01:00:00+01:00",
"customerData": {
  "email": "user@example.com",
  "name": "Jan Nowak",
  "nip": "aaa",
  "phone": "123",
  "loyaltyCardNumber": "sa2222",
  "address": {
    "street": "Bagno",
    "address1": "12",
    "province": "Mazowieckie",
    "city": "Warszawa",
    "postal": "00-800",
    "country": "PL"
  }
},
"labels": [
  {
    "key": "scan_id",
    "value": "123"
  }
],
"items": [
  {
    "sku": {
      "code": "SKU1"
    },
    "name": "item 1",
    "quantity": 1,
    "grossValue": 1,
    "category": "aaa",
    "maker": "sss",
    "labels": [
      {
        "key": "test",
        "value": "label"
      },
      {
        "key": "test",
        "value": "label2"
      }
    ]
  }
],
{
  "sku": {
    "code": "SKU2"
  },
  "name": "item 2",
  "quantity": 2,
  "grossValue": 2,
  "category": "bbb",
  "maker": "ccc",
```

(continues on next page)

(continued from previous page)

```
    "labels": []
  }
],
"currency": "eur",
"pointsEarned": 6.9
},
{
  "grossValue": 3,
  "transactionId": "00000000-0000-1111-0000-000000000005",
  "documentNumber": "888",
  "purchaseDate": "2018-02-20T09:45:04+0100",
  "purchasePlace": "wroclaw",
  "documentType": "sell",
  "customerId": "57524216-c059-405a-b951-3ab5c49bae14",
  "customerData": {
    "email": "o@lo.com",
    "name": "Jan Nowak",
    "nip": "aaa",
    "phone": "123",
    "loyaltyCardNumber": "sa2las222",
    "address": {
      "street": "Bagno",
      "address1": "12",
      "province": "Mazowieckie",
      "city": "Warszawa",
      "postal": "00-800",
      "country": "PL"
    }
  }
},
"labels": [
  {
    "key": "scan_id",
    "value": "234"
  }
],
"items": [
  {
    "sku": {
      "code": "SKU1"
    },
    "name": "item 1",
    "quantity": 1,
    "grossValue": 1,
    "category": "aaa",
    "maker": "sss",
    "labels": [
      {
        "key": "test",
        "value": "label"
      },
      {
        "key": "test",
        "value": "label2"
      }
    ]
  }
],
},
{
```

(continues on next page)

(continued from previous page)

```
    "sku": {
      "code": "SKU2"
    },
    "name": "item 2",
    "quantity": 2,
    "grossValue": 2,
    "category": "bbb",
    "maker": "ccc",
    "labels": []
  }
],
"currency": "eur",
"pointsEarned": 6
}
],
"total": 2
}
```

### Register a new transaction

To register a new transaction, you need to call the `/api/<storeCode>/transaction` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/transaction
```



Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to add transaction to.
transaction[transactionData][documentType]	query	Document type for Transaction Data, 2 possible values: return, sell
transaction[transactionData][documentNumber]	query	Document number
transaction[revisedDocument]	query	Sales document number
transaction[transactionData][purchaseDate]	query	<i>(optional)</i> Purchase date
transaction[items][][sku][code]	query	SKU Code
transaction[items][][name]	query	Product name
transaction[items][][quantity]	query	Quantity
transaction[items][][grossValue]	query	Gross value
transaction[items][][category]	query	Category Name
transaction[items][][maker]	query	Brand name
transaction[items][][labels][][key]	query	Label key
transaction[items][][labels][][value]	query	Label value
transaction[customerData][name]	query	Customer name
transaction[customerData][email]	query	<i>(optional)</i> Customer email
transaction[customerData][phone]	query	<i>(optional)</i> Customer phone
transaction[customerData][loyaltyCardNumber]	query	<i>(optional)</i> Customer Loyalty card number
transaction[customerData][nip]	query	<i>(optional)</i> Customer NIP
transaction[customerData][address][street]	query	<i>(optional)</i> Street
transaction[customerData][address][address1]	query	<i>(optional)</i> Customer address 1
transaction[customerData][address][postal]	query	<i>(optional)</i> Postal code
transaction[customerData][address][city]	query	<i>(optional)</i> City
transaction[customerData][address][province]	query	<i>(optional)</i> Province
transaction[customerData][address][country]	query	<i>(optional)</i> Country
transaction[labels][0][key]	query	<i>(optional)</i> First label key
transaction[labels][0][value]	query	<i>(optional)</i> First label value
transaction[labels][1][key]	query	<i>(optional)</i> Second label key
transaction[labels][1][value]	query	<i>(optional)</i> Second label value

**Note:** You need to provide one of the following: transaction[customerData][email], transaction[customerData][phone], transaction[customerData][loyaltyCardNumber], to match a customer with a transaction.

### Example

```
curl http://localhost:8181/api/DEFAULT/transaction \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..." \
-d "transaction[items][0][sku][code]=test0101" \
-d "transaction[items][0][name]=Product+name" \
-d "transaction[items][0][quantity]=1" \
-d "transaction[items][0][grossValue]=1500.00" \
-d "transaction[items][0][category]=Category+Name" \
-d "transaction[items][0][maker]=Marker+name" \
-d "transaction[items][0][labels][0][key]=Label+key" \
-d "transaction[items][0][labels][0][value]=Label+value" \
-d "transaction[customerData][name]=Firstname+Lastname" \
-d "transaction[customerData][email]=tomasztest8@wp.pl" \
-d "transaction[customerData][phone]=000000000000000" \
-d "transaction[customerData][loyaltyCardNumber]=11111111111" \
-d "transaction[customerData][nip]=000000000000000" \
-d "transaction[customerData][address][street]=Street+name" \
-d "transaction[customerData][address][address1]=123" \
-d "transaction[customerData][address][postal]=00-000" \
-d "transaction[customerData][address][city]=Wroclaw" \
-d "transaction[customerData][address][province]=Dolnoslaskie" \
-d "transaction[customerData][address][country]=PL" \
-d "transaction[transactionData][documentNumber]=214124124125" \
-d "transaction[transactionData][purchaseDate]=2019-02-20 09:28" \
-d "transaction[transactionData][documentType]=sell"
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6Ii...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

### Example Response

```
STATUS: 200 OK
```

```
{
  "transactionId": "d5b1119a-698b-40b4-9ac4-8ef704fa4433"
}
```

### Update transaction labels

To update transaction labels, you need to log in as admin and call the `/api/<storeCode>/admin/transaction/labels` endpoint with the POST method.

### Definition

```
POST /api/<storeCode>/admin/transaction/labels
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the transaction belongs to.
transaction_labels[transactionId]	query	Transaction ID
transaction_labels[labels][0][key]	query	<i>(optional)</i> First label key
transaction_labels[labels][0][value]	query	<i>(optional)</i> First label value
transaction_labels[labels][1][key]	query	<i>(optional)</i> Second label key
transaction_labels[labels][1][value]	query	<i>(optional)</i> Second label value

## Example

```
curl http://localhost:8181/api/DEFAULT/admin/transaction/labels \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "transaction_labels[transactionId]=00000000-0000-1111-0000-000000000000" \
-d "transaction_labels[label][0][key]=some label" \
-d "transaction_labels[label][0][value]=some value"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "transactionId": "d5b1119a-698b-40b4-9ac4-8ef704fa4433"
}
```

## Add new transaction labels as customer

To update transaction labels, you need to log in as a customer and call the `/api/<storeCode>/customer/transaction/labels/append` endpoint with the PUT method. A customer can only add new labels to transactions which are assigned to them.

## Definition

```
PUT /api/<storeCode>/customer/transaction/labels/append
```





Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store to simulate transaction in.
transaction[items][ ][sku][code]	query	SKU code
transaction[items][ ][name]	query	Product name
transaction[items][ ][quantity]	query	Quantity
transaction[items][ ][grossValue]	query	Gross value
transaction[items][ ][category]	query	Category name
transaction[items][ ][maker]	query	Brand name
transaction[items][ ][labels][ ][key]	query	Label key
transaction[items][ ][labels][ ][value]	query	Label value
transaction[purchaseDate]	query	Purchase date
transaction[customerData][name]	query	Customer name
transaction[customerData][email]	query	<i>(optional, see below)</i> Customer email
transaction[customerData][phone]	query	<i>(optional, see below)</i> Customer phone
transaction[customerData][loyaltyCardNumber]	query	<i>(optional, see below)</i> Loyalty card number
transaction[customerData][nip]	query	<i>(optional)</i> Customer NIP
transaction[customerData][address][street]	query	<i>(optional)</i> Street
transaction[customerData][address][address1]	query	<i>(optional)</i> Customer address1
transaction[customerData][address][postal]	query	<i>(optional)</i> Postal code
transaction[customerData][address][city]	query	<i>(optional)</i> City
transaction[customerData][address][province]	query	<i>(optional)</i> Province
transaction[customerData][address][country]	query	<i>(optional)</i> Country

**Heads up!** One of the following: email, phone, loyaltyCardNumber is required, along with the name, in order to find the user for the simulation to be performed.

### Example

```
curl http://localhost:8181/api/DEFAULT/transaction/simulate \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "transaction[items][0][sku][code]=SKU1" \
-d "transaction[items][0][name]=item+8" \
-d "transaction[items][0][quantity]=1" \
-d "transaction[items][0][grossValue]=1" \
-d "transaction[items][0][category]=aaa" \
-d "transaction[items][0][maker]=sss" \
-d "transaction[items][0][labels][0]=labels" \
-d "transaction[items][0][labels][0][key]=test" \
-d "transaction[items][0][labels][0][value]=label" \
-d "transaction[purchaseDate]=2022-02-20T09:45:04+0100"
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "points": 2.3
}
```

## Get transaction details (admin)

To get transaction details, you need to call the `/api/<storeCode>/transaction/<transaction>` endpoint with the GET method.

## Definition

```
GET /api/<storeCode>/transaction/<transaction>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<storeCode>	query	Code of the store the transaction belongs to.
<transaction>	query	Transaction ID

## Example

To see details of a transaction with ID 00000000-0000-1111-0000-000000000005, use the below method:

```
curl http://localhost:8181/api/DEFAULT/transaction/00000000-0000-1111-0000-000000000005 \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.."
```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```
{
  "grossValue": 3,
  "transactionId": "00000000-0000-1111-0000-000000000005",
  "documentNumber": "888",
  "purchaseDate": "2018-02-20T09:45:04+0100",
}
```

(continues on next page)

```
"purchasePlace": "wroclaw",
"documentType": "sell",
"customerId": "57524216-c059-405a-b951-3ab5c49bae14",
"assignedToCustomerDate": "1970-01-01T01:00:00+01:00",
"customerData": {
  "email": "o@lo.com",
  "name": "Jan Nowak",
  "nip": "aaa",
  "phone": "123",
  "loyaltyCardNumber": "sa21as222",
  "address": {
    "street": "Bagno",
    "address1": "12",
    "province": "Mazowieckie",
    "city": "Warszawa",
    "postal": "00-800",
    "country": "PL"
  }
},
"labels": [
  {
    "key": "scan_id",
    "value": "123"
  }
],
"items": [
  {
    "sku": {
      "code": "SKU1"
    },
    "name": "item 1",
    "quantity": 1,
    "grossValue": 1,
    "category": "aaa",
    "maker": "sss",
    "labels": [
      {
        "key": "test",
        "value": "label"
      },
      {
        "key": "test",
        "value": "label2"
      }
    ]
  },
  {
    "sku": {
      "code": "SKU2"
    },
    "name": "item 2",
    "quantity": 2,
    "grossValue": 2,
    "category": "bbb",
    "maker": "ccc",
    "labels": []
  }
]
```

(continues on next page)



(continued from previous page)

```

},
"currency": "eur",
"pointsEarned": 6
}

```

## 2.1.19 ACL API

These endpoints will allow you to easily manage ACL for the administrator.

### Creating a role

To create a new role, you need to call the `/api/admin/acl/role` endpoint with the POST method.

### Definition

```
POST /api/admin/acl/role
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
role[name]	request	Name
role[default]   request		Mark as default role
role[permissions][][resource]	request	Permission resource
role[permissions][][access]	request	Permission access type (MODIFY, VIEW)

### Example

To create a new role, use the method below:

```

curl http://localhost:8181/api/admin/acl/role \
-X "POST" \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. " \
-d "role[name]=Super admin" \
-d "role[default]=true" \
-d "role[permissions][0][resource]=LEVEL" \
-d "role[permissions][0][access]=MODIFY" \
-d "role[permissions][1][resource]=EARNING_RULE" \
-d "role[permissions][1][access]=MODIFY"

```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).



(continued from previous page)

```

        "id": 56,
        "resource": "SEGMENT_EXPORT",
        "access": "VIEW"
    },
    {
        "id": 55,
        "resource": "LEVEL",
        "access": "VIEW"
    }
]
}

```

**Note:** The 37 id is an example value. Your value may be different.

## Collection of available roles

To retrieve a list of roles, you need to call the `/api/admin/acl/role` endpoint with the GET method.

### Definition

```
GET /api/admin/acl/role
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication

To see the list of available roles, use the method below:

### Example

```

curl http://localhost:8181/api/admin/acl/role \
-X "GET" -H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I.. "

```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

### Example Response

```
STATUS: 200 OK
```

```

{
  "roles": [
    {

```

(continues on next page)

```

    "id": 37,
    "name": "Super admin",
    "role": "ROLE_ADMIN",
    "master": true,
    "default": true,
    "permissions": []
  },
  {
    "id": 38,
    "name": "Reporter admin",
    "role": "ROLE_ADMIN",
    "master": false,
    "default": false,
    "permissions": [
      {
        "id": 57,
        "resource": "EARNING_RULE",
        "access": "VIEW"
      },
      {
        "id": 56,
        "resource": "SEGMENT_EXPORT",
        "access": "VIEW"
      },
      {
        "id": 55,
        "resource": "LEVEL",
        "access": "VIEW"
      }
    ]
  }
],
"total": 2
}

```

## Updating a role

To update a role, you need to call the `/api/admin/acl/role/<role>` endpoint with the PUT method.

### Definition

```
PUT /api/admin/acl/role/<role>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
role[name]	request	Name
role[default]	request	Mark as default role
role[permissions][][resource]	request	Permission resource
role[permissions][][access]	request	Permission access type (MODIFY, VIEW)

## Example

To update the role with id = 37, use the method below:

```
curl http://localhost:8181/api/admin/acl/role/37 \
-H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..." \
-X "PUT" \
-d "role[name]=Super admin" \
-d "role[default]=true" \
-d "role[permissions][0][resource]=LEVEL" \
-d "role[permissions][0][access]=MODIFY" \
-d "role[permissions][1][resource]=EARNING_RULE" \
-d "role[permissions][1][access]=MODIFY" \
```

## Example Response

```
STATUS: 204 No Content
```

## Collection of available resources

To retrieve a list of available resources, you need to call the `/api/admin/acl/resources` endpoint with the GET method.

## Definition

```
GET /api/admin/acl/resources
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication

To see the list of available resources, use the method below:

## Example

```
curl http://localhost:8181/api/admin/acl/resources \
-X "GET" -H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..."
```

## Example Response

```
STATUS: 200 OK
```

```
{
  "resources": [
    {
      "code": "SEGMENT_EXPORT",
      "name": "Utilities"
    },
    {
      "code": "EARNING_RULE",
      "name": "Earning rules"
    },
    {
      "code": "LEVEL",
      "name": "Levels"
    }
  ],
  "total": 3
}
```

### Collection of available access types

To retrieve a list of available access types, you need to call the `/api/admin/acl/accesses` endpoint with the GET method.

#### Definition

```
GET /api/admin/acl/accesses
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication

To see the list of available access types, use the method below:

#### Example

```
curl http://localhost:8181/api/admin/acl/accesses \
-X "GET" -H "Accept: application/json" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IiI6..."
```

#### Example Response

```
STATUS: 200 OK
```

```
{
  "accesses": [
    {
      "code": "VIEW",
      "name": "View"
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    },
    {
      "code": "MODIFY",
      "name": "Modify"
    }
  ],
  "total": 2
}

```

## Deleting a single role

To delete specific role, you need to call the `/api/admin/acl/role/{role}` endpoint with the `DELETE` method.

### Definition

```
DELETE /api/admin/acl/role/{role}
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<role>	query	Id of the role

### Example

```

curl http://localhost:8181/api/admin/acl/role/37
-X "DELETE" -H "Accept: application/json"
-H "Content-type: application/x-www-form-urlencoded"
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IiI6..."

```

### Example Response

```
204 No Content
```

**Note:** The `37` id is an example value. Your value can be different.

## 2.1.20 Event API

These endpoints will allow you to retrieve events from Event Store.

### Get the list of events

To retrieve a paginated list of events, you need to call the `/api/event` endpoint with the `GET` method.

### Definition

```
GET /api/event
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
fromId	query	<i>(optional)</i> Start from event ID, by default from the beginning.
perPage	query	<i>(optional)</i> Number of items to display per page, by default = 10
sort	query	<i>(optional)</i> Sort by column name, by default = name
direction	query	<i>(optional)</i> Direction of sorting [ASC, DESC], by default = ASC

### Example

```
curl http://localhost:8181/api/event \
-X "GET" \
-H "Accept: application/json" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6..."
```

---

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6...` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

---

---

**Note:** Endpoint will return the first 100 items from the given `<eventId>`.

---

### Example Response

```
STATUS: 200 OK
```

```
{
  "events": [
    {
      "id": 1,
      "uuid": "e5f6e6a2-18f6-4569-8c5c-8f87c7717d87",
      "payload": {
        "accountId": {
          "accountId": "e5f6e6a2-18f6-4569-8c5c-8f87c7717d87"
        },
        "pointsTransfer": {
          "id": {
            "pointsTransferId": "e82c96cf-32a3-43bd-9034-4df343e5f333"
          },
          "comment": "Example comment",
          "createdAt": "2020-10-15T12:06:38+02:00",
          "value": 100,
          "canceled": false,
          "issuer": "system"
        }
      }
    }
  ],
}
```

(continues on next page)



(continued from previous page)

```

        "type": "OpenLoyalty.Domain.Account.Event.PointsWereSpent",
        "recorderOn": "2020-10-15T10:06:38+00:00"
      },
    ],
    "total": 1
  }
}

```

## Get event details

To retrieve the details of a event, you need to call the `/api/event/<eventId>` endpoint with the GET method.

## Definition

```
GET /api/event/<eventId>
```

Parameter	Parameter type	Description
Authorization	header	Token received during authentication
<eventId>	query	Event ID

## Example

To see the details of the event with id `eventId = 212333`, use the method below:

```

curl http://localhost:8181/api/event/212333 \
  -X "GET" \
  -H "Accept: application/json" \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6I..

```

**Note:** The `eyJhbGciOiJSUzI1NiIsInR5cCI6I..` authorization token is an example value. Your value may be different. Read more about Authorization [here](#).

## Example Response

```
STATUS: 200 OK
```

```

{
  "id": 73,
  "uuid": "00000000-0000-474c-b092-b0dd880c07e2",
  "payload": {
    "customerId": {
      "customerId": "00000000-0000-474c-b092-b0dd880c07e2"
    },
    "levelId": {
      "levelId": "e82c96cf-32a3-43bd-9034-4df343e51111"
    },
    "oldLevelId": {

```

(continues on next page)

```

        "levelId": "e82c96cf-32a3-43bd-9034-4df343e50000"
    },
    "updateAt": "2020-10-15T12:06:23+02:00",
    "manually": false,
    "removeLevelManually": false
},
"type": "OpenLoyalty.Domain.User.Event.CustomerWasMovedToLevel",
"recorderOn": "2020-10-15T10:06:23+00:00"
}

```

## List of available events

OpenLoyalty.Domain.User.Event.AssignedAccountToCustomer  
 OpenLoyalty.Domain.User.Event.AssignedTransactionToCustomer  
 OpenLoyalty.Domain.User.Event.CampaignCouponWasChanged  
 OpenLoyalty.Domain.User.Event.CampaignStatusWasChanged  
 OpenLoyalty.Domain.User.Event.CampaignUsageWasChanged  
 OpenLoyalty.Domain.User.Event.CampaignWasBoughtByCustomer  
 OpenLoyalty.Domain.User.Event.CampaignWasReturned  
 OpenLoyalty.Domain.User.Event.CustomerAddressWasUpdated  
 OpenLoyalty.Domain.User.Event.CustomerAvatarWasRemoved  
 OpenLoyalty.Domain.User.Event.CustomerAvatarWasSet  
 OpenLoyalty.Domain.User.Event.CustomerCompanyDetailsWereUpdated  
 OpenLoyalty.Domain.User.Event.CustomerDetailsWereUpdated  
 OpenLoyalty.Domain.User.Event.CustomerInvitationTokenWasCreated  
 OpenLoyalty.Domain.User.Event.CustomerLevelWasRecalculated  
 OpenLoyalty.Domain.User.Event.CustomerLoyaltyCardNumberWas  
 OpenLoyalty.Domain.User.Event.CustomerWasActivated  
 OpenLoyalty.Domain.User.Event.CustomerWasAnonymized  
 OpenLoyalty.Domain.User.Event.CustomerWasAssignedToStore  
 OpenLoyalty.Domain.User.Event.CustomerWasAttachedToInvitation  
 OpenLoyalty.Domain.User.Event.CustomerWasDeactivated  
 OpenLoyalty.Domain.User.Event.CustomerWasDeleted  
 OpenLoyalty.Domain.User.Event.CustomerWasMovedToLevel  
 OpenLoyalty.Domain.User.Event.CustomerWasRegistered  
 OpenLoyalty.Domain.User.Event.InvitationWasCreated  
 OpenLoyalty.Domain.User.Event.PosWasAssignedToCustomer  
 OpenLoyalty.Domain.User.Event.SellerWasAssignedToCustomer  
 OpenLoyalty.Domain.Account.Event.AccountWasAssignedToStore  
 OpenLoyalty.Domain.Account.Event.AccountWasCreated  
 OpenLoyalty.Domain.Account.Event.PointsHasBeenReset  
 OpenLoyalty.Domain.Account.Event.PointsTransferHasBeenCanceled  
 OpenLoyalty.Domain.Account.Event.PointsTransferHasBeenEx  
 OpenLoyalty.Domain.Account.Event.PointsTransferHasBeenUnlocked  
 OpenLoyalty.Domain.Account.Event.PointsWereAdded  
 OpenLoyalty.Domain.Account.Event.PointsWereBlocked  
 OpenLoyalty.Domain.Account.Event.PointsWereReturned  
 OpenLoyalty.Domain.Account.Event.PointsWereSpent  
 OpenLoyalty.Domain.Account.Event.PointsWereTransferred  
 OpenLoyalty.Domain.Campaign.Event.CampaignBoughtDeliveryStatusWasChanged  
 OpenLoyalty.Domain.Campaign.Event.CampaignBoughtCashbackStatusWasChanged  
 OpenLoyalty.Domain.Campaign.Event.CampaignUsageWasChanged  
 OpenLoyalty.Domain.Transaction.Event.TransactionWasRegistered  
 OpenLoyalty.Domain.Transaction.Event.CustomerWasAssignedToTransaction

- *Introduction to the Open Loyalty REST API*
- *Authorization*
- *Admin Users API*
- *Analytics API*
- *Audit API*
- *Reward Campaigns API*
- *Customer API*
- *Customer Campaign API*
- *Customer Level API*
- *Customer Earning API*
- *Customer Points transfers*

- *Earning Rule*
- *Invitation*
- */api/level*
- *Points transfers*
- */api/pos*
- *Security*
- */api/seller*
- */api/segment*
- */api/settings*
- *Store API*
- *Transactions*
- */api/utility*
- *Campaigns categories API*
- *ACL API*
- *Event API*
- *Introduction to the Open Loyalty REST API*
- *Authorization*
- *Admin Users API*
- *Analytics API*
- *Audit API*
- *Reward Campaigns API*
- *Customer API*
- *Customer Campaign API*
- *Customer Level API*
- *Customer Earning API*
- *Customer Points transfers*
- *Earning Rule*
- *Invitation*
- */api/level*
- *Points transfers*
- */api/pos*
- *Security*
- */api/seller*
- */api/segment*
- */api/settings*
- *Store API*

- *Transactions*
- */api/utility*
- *Campaigns categories API*
- *ACL API*
- *Event API*

*The Developer's Guide* contains information for developers who want to know more about Open Loyalty architecture and the concepts used within. This guide helps developers to understand how Open Loyalty works.

### 3.1 The Developer's Guide

The Developer's Guide contains information for developers who want to know more about Open Loyalty architecture and used concepts. This guide helps to understand how Open Loyalty works.

#### 3.1.1 Introduction

##### Introduction

- [/developer/introduction/about](#)
- [/developer/introduction/used\\_concepts](#)
- [/developer/introduction/about](#)
- [/developer/introduction/used\\_concepts](#)

#### 3.1.2 Installation

##### Installation

- [/developer/installation/requirements](#)
- [/developer/installation/installation](#)
- [/developer/installation/upgrading](#)
- [/developer/installation/kubernetes](#)

- [/developer/installation/requirements](#)
- [/developer/installation/installation](#)
- [/developer/installation/upgrading](#)
- [/developer/installation/kubernetes](#)

### 3.1.3 Architecture

#### Architecture

- [/developer/architecture/overview](#)
- [/developer/architecture/domain\\_driven\\_design](#)
- [/developer/architecture/cqrs](#)
- [/developer/architecture/event\\_sourcing](#)
- [/developer/architecture/api](#)
- [/developer/architecture/cockpits](#)
- [/developer/architecture/fixtures](#)
- [/developer/architecture/events](#)
- [/developer/architecture/translations](#)
- [/developer/architecture/overview](#)
- [/developer/architecture/domain\\_driven\\_design](#)
- [/developer/architecture/cqrs](#)
- [/developer/architecture/event\\_sourcing](#)
- [/developer/architecture/api](#)
- [/developer/architecture/cockpits](#)
- [/developer/architecture/fixtures](#)
- [/developer/architecture/events](#)
- [/developer/architecture/translations](#)

### 3.1.4 Configuration

#### Configuration

- [/developer/configuration/channels](#)
- [/developer/configuration/locales](#)
- [/developer/configuration/currencies](#)
- [/developer/configuration/channels](#)
- [/developer/configuration/locales](#)
- [/developer/configuration/currencies](#)

### 3.1.5 Local environment

#### Environment setup

- /developer/environment/xdebug
- /developer/environment/tests
- /developer/environment/elasticsearch
- /developer/environment/rabbitmq
- /developer/environment/redis
- /developer/environment/varnish
- /developer/environment/xdebug
- /developer/environment/tests
- /developer/environment/elasticsearch
- /developer/environment/rabbitmq
- /developer/environment/redis
- /developer/environment/varnish
- /developer/introduction
- /developer/installation
- /developer/architecture
- /developer/configuration
- /developer/points





*The Open Loyalty Cookbook* is a collection of solution articles helping you with some specific, narrow problems.

### 4.1 The Cookbook

The Open Loyalty's Cookbook is a collection of solution articles helping you with some specific, narrow problems.

#### 4.1.1 Introduction

- [/cookbook/create\\_a\\_new\\_bundle](#)
- [/cookbook/create\\_a\\_new\\_api\\_endpoint](#)
- [/cookbook/create\\_a\\_new\\_command\\_and\\_handler](#)
- [/cookbook/how\\_to\\_backup\\_elasticsearch](#)
- [/cookbook/how\\_to\\_change\\_domain](#)
- [/cookbook/how\\_to\\_change\\_main\\_language](#)
- [/cookbook/how\\_to\\_schedule\\_transactions\\_import](#)
- [/cookbook/how\\_to\\_work\\_with\\_queues](#)
- [/cookbook/how\\_to\\_add\\_new\\_field\\_to\\_entity](#)
- [/cookbook/how\\_to\\_add\\_new\\_tab\\_in\\_admin\\_panel](#)
- [/cookbook/how\\_to\\_enable\\_ldap\\_authorization](#)
- [/cookbook/translatable\\_field](#)
- [/cookbook/FAQ](#)
- [/cookbook/create\\_a\\_new\\_bundle](#)
- [/cookbook/create\\_a\\_new\\_api\\_endpoint](#)

- [/cookbook/create\\_a\\_new\\_command\\_and\\_handler](#)
- [/cookbook/how\\_to\\_backup\\_elasticsearch](#)
- [/cookbook/how\\_to\\_change\\_domain](#)
- [/cookbook/how\\_to\\_change\\_main\\_language](#)
- [/cookbook/how\\_to\\_schedule\\_transactions\\_import](#)
- [/cookbook/how\\_to\\_work\\_with\\_queues](#)
- [/cookbook/how\\_to\\_add\\_new\\_field\\_to\\_entity](#)
- [/cookbook/how\\_to\\_add\\_new\\_tab\\_in\\_admin\\_panel](#)
- [/cookbook/how\\_to\\_enable\\_ldap\\_authorization](#)
- [/cookbook/translatable\\_field](#)
- [/cookbook/FAQ](#)

*The Changelog* is a detailed list of changes in each version of Open Loyalty.

### 5.1 Changelog

All notable changes to this project will be documented in this file.

The format is based on *Keep a Changelog* and this project adheres to *Semantic Versioning*.

- /changelog/4.x/index
- /changelog/3.x/index
- /changelog/2.x/index
- /changelog/1.x/index
- /changelog/4.x/index
- /changelog/3.x/index
- /changelog/2.x/index
- /changelog/1.x/index



**A**

Authorization, 8

**I**

Introduction, 7