



API Documentation

V. 0.2.1

1.	Introduction	3
2.	Getting Started	4
2.1.	The process	5
2.2.	Environnement	6
3.	Authentication	7
3.1.	Bearer Authentication	7
3.2.	API Key authentication	7
4.	Code Action	8
4.1.	Requirement	8
4.2.	API Paths	8
4.3.	Swagger Documentation	8
5.	Code Status	9
5.1.	Code status	9
5.2.	Code Expiration	9
5.3.	Code Metadata	10
6.	API Return & Error	11
6.1.	API Return	11
6.2.	API Errors	12
7.	Happy Path Example	13
7.1.	Get status request	13
7.2.	Get status response	13
7.3.	Lock Request	14
7.4.	Lock Response	14
7.5.	Redeem Request	15
7.6.	Redeem Response	15

1. Introduction

The Crossout API provide complete freedom in the integration of our access control ticketing system. In this Developer Documentation we provide code and response examples of the functions you need to verify and use a code.

The API is organized around REST. Our API has predictable resource-oriented URLs, accept from-encoded request bodies, return JSON-encoded responses and uses standard HTTP response code, authentication, and verbs.

2. Getting Started

This documentation covers the following topics :

- The check in process
- Authenticating a request
- Requesting the status of a code
- Locking a code
- Releasing a code
- Redeeming a code
- Cancelling a code just used
- Possible code statuses

Before you use the Crossout API you need to establish the following two things:

- The integration methode you want to use :
 - Machine 2 Machine integration of the REST API (API key authentication + m2m REST API)
 - Client integration (Bearer authentication with login + client API)
- Request credentials to use the Crossout API (API key or client AGENT user)
 - Contact Seaters

2.1. The process

The check in process consists of several steps that are shown in the image.

1. Get code status
2. Lock code
3. Release code
4. Redeem code
5. Cancel redeem action

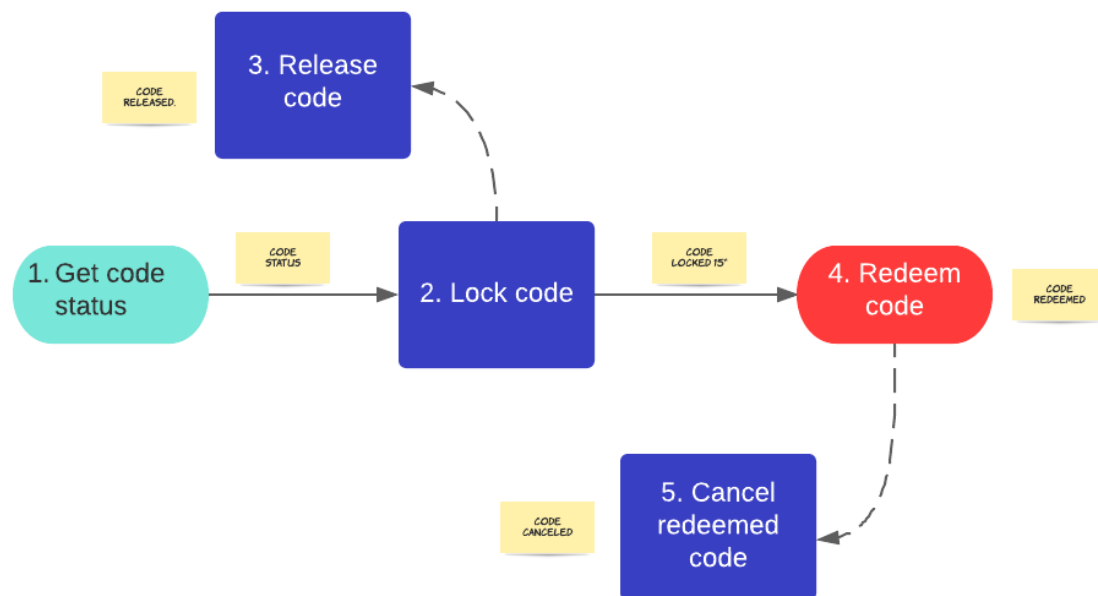


FIG 1. CHECK IN PROCESS FLOW

2.2. Environnement

Within Crossout, 2 environnements are available.

The following table describes the base URLs of the endpoints in each environment. The URLs described in later sections are relative to these base URLs.

Environment	Endpoint base Url
Acceptance	https://qa-crossout.app/api/
Production	https://crossout.app/api/

3. Authentication

3.1. Bearer Authentication

In order to authenticate an API request as a user, *Bearer (Token) Authentication* is used. This entails attaching a token prefixed by *Bearer* to the header off the request wick called the Authorization Header.

An example of an Authorization header with Bearer token :

```
'Authorization: Bearer
eyJ0eXA0IjK1VjIjCjhbGciOiJSUzI1NiJ9.eyJpc3MiOiJqdHRwOi8vc3RyaWt1b3V0LmNvZGVzIiwidXBUiJoic3VwZXJhZG1pbjJAc2VhdG9ycy5jb20iLCJncm91c2hiOiJsIiwiaWF0IjVQRVJfQURNSU4iLCJBR0V0VCI6IjE7bTRFRHVrIiwic29iLCJBRRE1TjIidCJpYXQ0IjE2Mjg5bW9DES3N0SImV4C16MTYyODI0NTM0OwCiw3ViJoic3VwZXJhZG1pbjJAc2VhdG9ycy5jb20iLCJqdGkiOiI1MzZkNTI1S0S0N1NRhRlTRhM2M0OWEzNSI1ZTBiZDMwOGU0ZDMiQ0.0a8wxzpdqwx80k3nG8aUY_c01_CNXmRBFY0Ek9MdaGAPvps8S9fCwCkLkvLwPQvSMjeyIOh_ch2fyQ0ZQIEJCLd1Eg6zp0Iu6p4WkNRX08gFcpsmLCFQ1UiPco7Pndm0EZSPdAYBv4R1nJdZHuwniG51DvPqYa1Xs9UyZBg3_ODRFKtUicQ_Q61yLrKRNpfHJ0ue5yqrUYEYKEURY1NrAi4DkNRdL6gSyZJvchznK_EJwm4Wrrqas1k0oNvXSGkLfaJgwmuujmhpIBS3BCUAWY9I117JAuwk5LSLQJ34WZFa2i5BHzcThlysvsxjwgB7r28936Z6L9jsw'
```

To obtain a token, with a validity period of 60 minutes, a prior authentication request is necessary.

An example of Login request :

```
POST /api/auth/login HTTP/1.1
Host: crossout.app
Content-Type: application/json
Content-Length: 64

{
  "username": "agent@crossout.app",
  "password": "Ag3ntSecret!"
}
```

3.2. API Key authentication

To facilitate the integration of the Crossout API, a second authentication method is available. Authentication by API key is dedicated to applications for machine-to-machine communication.

APIs keys must not be shared and must remain secret, never accessible by the client and only used for server to server requests.

API key-based authentication is only considered secure if used together with other security mechanisms such as HTTPS/SSL.

An example of an Authorization header with Bearer token :

```
'Authorization: crossoutApiKey v2BdfEbGJnrIr6e5cL0GsZETJMsSufTG'
```

4. Code Action

4.1. Requirement

Any action on a code requires an authorization header (bearer or api key)

When you perform an action on a code, you must provide in addition to the code a venue Id, which corresponds to the place concerned by the request. When you are asked for credentials to use the Crossout API, you will be given a list specific to your use of venue Id

In most of the requests relative to codes Crossout return JSON-encoded response including the actual status of the code and a check to determine if the action has been completed (eventInserted : true/false)

4.2. API Paths

The url of the requests depends on the environment on which you are (acceptance or production) and the type of authentication you use (Bearer token or API key).

The following table describes the path of the endpoints relative to Crossout code in each authentication type.

Authentication type	Code endpoint path
Bearer token	/api/codes
API key	/api/m2m/service/codes

4.3. Swagger Documentation

In order to simplify Crossout API development for users, teams, and enterprises, we provide a Swagger documentation which is kept up to date with each new version of the Crossout API.

The following table describes the swagger documentation url in each authentication type.

Authentication type	Swagger documentation URL
Bearer token	https://qa-crossout.app/q/swagger-ui/#/Code

API key	https://qa-crossout.app/q/swagger-ui/#/REST%20Code%20API
---------	----------------------------------------------------------

5. Code Status

5.1. Code status

When you request the status of a code, you may receive a number of different results. The following table contains a complete overview.

Status	Type	Description
CREATED	<i>lockable</i>	The code is created and ready to be locked.
LOCK_EXPIRED	<i>lockable</i>	The locking has expired, the code could be locked again.
RELEASED	<i>lockable</i>	The code has been released, it could be locked again.
LOCKED	<i>redeemable</i>	The code has been locked and ready to be redeemed.
REDEEMED	<i>cancelable</i>	The code is redeemed. The action can be cancelled within 5'.
CANCELED	<i>lockable</i>	Redeem has been cancelled, the code can be locked again.
SUSPENDED	<i>n.a</i>	The code has been suspended. No action is possible.

5.2. Code Expiration

The expiration date of a code is returned in the parameter `expirationDate`

`expired` indicates if a code is expired or not. If a code is expired, it can't be used by the API (lock)

In case the expiration date of a code is null, the code is valid indefinitely and `expired` always returns false

```
"expired": false,
```

5.3. Code Metadata

The metadata is specific to a code. Codes from the same campaign can have different metadata

```
"metadata": {  
  "logoImg": "https://static.qa-seaters.com/brightfish_logo-57b2ba1f-396d-4501-92de-84ae49370e57.png",  
  "voucherValue": 1,  
  "voucherType": "TICKET",  
  "facialPrice": 0,  
  "currency": "EUR",  
  "operatingMode": "CHARGEABLE"  
}
```

- **LogoImg** : represents the logo of a code (absolute link to an image)
- **VoucherValue * Voucher type** : represents the nature of the voucher issued by a code. The following list is non-exhaustive and can be completed later. Examples :
 - *X * M_PLUS_N_TICKET*
 - *X * EUR_DISCOUNT*
 - *X * TICKET*
- **OperatingMode** : specifies if a voucher is rechargeable or not
 - CHARGEABLE | NON_CHARGEABLE
- **FacialPrice currency** : represents the value to be re-invoiced

6. API Return & Error

6.1. API Return

Any action (lock, release, redeem, cancel) on a code returns the current state of the code, including status (code event type, campaign info and expiration), audit data (eventInsertedFrom, eventInsertedAt, eventInsertedBy), the actual code.

Example with the following lock request :

```
curl --location --request PUT '{domaine}/api/m2m/service/codes/lock' \
--header 'Authorization: crossoutApiKey {apiKey}' \
--header 'Content-Type: application/json' \
--data-raw '{
  "code": "0123456789123",
  "venueId": "2970500915523502701"
}'
```

RESPONSE HTTP STATUS :

200 OK

RESPONSE BODY :

```
{
  "status": {
    "event": "LOCKED",
    "campaignId": "2752703686899686957",
    "campaignName": "BrightFish Fest 2021",
    "campaignProperties": {
      "startDate": "2021-05-12",
      "endDate": "2021-05-19",
      "templateHeaderImg": "https://static.qa-seaters.com/cinemavip-7dfc81d5-89cc-435e-8372-6bd87e9403f2.png",
      "templateLogoImg": "https://static.qa-seaters.com/brightfish_logo-57b2ba1f-396d-4501-92de-84ae49370e57.png",
      "templateDescriptionTxt": "Quidam urbs vero plerumque punico perspicabilis quem ad litus domicilium consumpsit amni illius",
      "templateFooterTxt": "Powered by Seaters",
      "templateColor": "#3c6cb3",
      "templateFont": "Lato-Regular"
    },
    "expirationDate": "2021-12-31T22:59:59Z",
    "expired": false,
    "metadata": {
      "logoImg": "https://static.qa-seaters.com/brightfish_logo-57b2ba1f-396d-4501-92de-84ae49370e57.png",
      "voucherValue": 1,
      "voucherType": "TICKET",
      "facialPrice": 0,
      "currency": "EUR",
      "operatingMode": "CHARGEABLE"
    }
  },
  "eventInsertedFrom": "UGC Antwerpen",
  "eventInsertedAt": "2021-12-07T13:17:28.452370Z",
  "eventInsertedBy": "ugc-testing@crossout.app",
  "code": "0000000010bf",
  "eventInserted": true
}
```

6.2. API Errors

Http status	Http code	Body	Details
OK	200	Action result (code status)	The action was successful. The result represents the current state of the code after the action
Expectation Failed	417	Action result (code status)	The action failed. The result represents the current state of the code, it has not been modified by the action since it failed
Not Found	404	N.A	The code was not found. It does not exist in the Crossout database or does not belong to the scope of your account
Unauthorized	401	N.A	The request was not completed because valid authentication information for the target resource is missing.
Service Temporarily Unavailable	503	HTML	The server is temporarily unable to handle the request. Possibly ongoing maintenance

7. Happy Path Example

Happy path (sometimes called happy flow) is a default scenario featuring no exceptional or error conditions.

it is composed of a sequence of 3 actions : Getting the status, locking the code (LOCK), and consuming the code (REDEEM).

The following screenshots (from Postman) show the scenario for the consumption of the code **01234567891011** in venue "Cinema" (id **1114890479980279775**)

7.1. Get status request

```
GET /api/m2m/service/codes/01234567891011 HTTP/1.1
Host: qa-crossout.app
Authorization: crossoutApiKey testingApiKey2021
```

7.2. Get status response

```
{
  "status": {
    "event": "CREATED",
    "campaignId": "2752703686899686957",
    "campaignName": "BrightFish Fest 2021",
    "campaignProperties": {
      "startDate": "2021-05-12",
      "endDate": "2021-05-19",
      "templateHeaderImg": "https://static.qa-seaters.com/cinemavip-7dfc81d5-89cc-435e-8372-6bd87e9403f2.png",
      "templateLogoImg": "https://static.qa-seaters.com/brightfish_logo-57b2ba1f-396d-4501-92de-84ae49370e57.png",
      "templateDescriptionTxt": "Quidam urbs vero plerumque punico perspicabilis quem ad litus domicilium consumpsit amni illius",
      "templateFooterTxt": "Powered by Seaters",
      "templateColor": "#3c6cb3",
      "templateFont": "Lato-Regular"
    },
    "expirationDate": "2021-12-31T22:59:59Z",
    "expired": false,
    "metadata": {
      "logoImg": "https://static.qa-seaters.com/brightfish_logo-57b2ba1f-396d-4501-92de-84ae49370e57.png",
      "voucherValue": 1,
      "voucherType": "TICKET",
      "facialPrice": 0,
      "currency": "EUR",
      "operatingMode": "CHARGEABLE"
    }
  },
  "eventInsertedFrom": null,
  "eventInsertedAt": "2021-12-07T13:15:17.170371Z",
  "eventInsertedBy": "superadmin2@seaters.com"
}
```

7.3. Lock Request

```
PUT /api/m2m/service/codes/lock HTTP/1.1
Host: qa-crossout.app
Authorization: crossoutApiKey testingApiKey2021
Content-Type: application/json
Content-Length: 68

{
  "code": "01234567891011",
  "venueId": "1114890479980279775"
}
```

7.4. Lock Response

```
{
  "status": {
    "event": "LOCKED",
    "campaignId": "2752703686899686957",
    "campaignName": "BrightFish Fest 2021",
    "campaignProperties": {
      "startDate": "2021-05-12",
      "endDate": "2021-05-19",
      "templateHeaderImg": "https://static.qa-seaters.com/cinemavip-7dfc81d5-89cc-435e-8372-6bd87e9403f2.png",
      "templateLogoImg": "https://static.qa-seaters.com/brightfish_logo-57b2balf-396d-4501-92de-84ae49370e57.png",
      "templateDescriptionTxt": "Quidam urbs vero plerumque punico perspicabilis quem ad litus domicilium consumpsit amni illius",
      "templateFooterTxt": "Powered by Seaters",
      "templateColor": "#3c6cb3",
      "templateFont": "Lato-Regular"
    },
    "expirationDate": "2021-12-31T22:59:59Z",
    "expired": false,
    "metadata": {
      "logoImg": "https://static.qa-seaters.com/brightfish_logo-57b2balf-396d-4501-92de-84ae49370e57.png",
      "voucherValue": 1,
      "voucherType": "TICKET",
      "facialPrice": 0,
      "currency": "EUR",
      "operatingMode": "CHARGEABLE"
    }
  },
  "eventInsertedFrom": "UGC Antwerpen",
  "eventInsertedAt": "2021-12-07T13:17:28.452370Z",
  "eventInsertedBy": "ugc-testing@crossout.app",
  "code": "0000000010bf",
  "eventInserted": true
}
```

7.5. Redeem Request

```
PUT /api/m2m/service/codes/redeem HTTP/1.1
Host: qa-crossout.app
Authorization: crossoutApiKey testingApiKey2021
Content-Type: application/json
Content-Length: 68
```

```
{
  "code": "01234567891011",
  "venueId": "1114890479980279775"
}
```

7.6. Redeem Response

```
{
  "status": {
    "event": "REDEEMED",
    "campaignId": "2752703686899686957",
    "campaignName": "BrightFish Fest 2021",
    "campaignProperties": {
      "startDate": "2021-05-12",
      "endDate": "2021-05-19",
      "templateHeaderImg": "https://static.qa-seaters.com/cinemavip-7dfc81d5-89cc-435e-8372-6bd87e9403f2.png",
      "templateLogoImg": "https://static.qa-seaters.com/brightfish_logo-57b2ba1f-396d-4501-92de-84ae49370e57.png",
      "templateDescriptionTxt": "Quidam urbs vero plerumque punico perspicabilis quem ad litus domicilium consumpsit amni illius",
      "templateFooterTxt": "Powered by Seaters",
      "templateColor": "#3c6cb3",
      "templateFont": "Lato-Regular"
    },
    "expirationDate": "2021-12-31T22:59:59Z",
    "expired": false,
    "metadata": {
      "logoImg": "https://static.qa-seaters.com/brightfish_logo-57b2ba1f-396d-4501-92de-84ae49370e57.png",
      "voucherValue": 1,
      "voucherType": "TICKET",
      "facialPrice": 0,
      "currency": "EUR",
      "operatingMode": "CHARGEABLE"
    }
  },
  "eventInsertedFrom": "UGC Antwerpen",
  "eventInsertedAt": "2021-12-07T13:18:58.373557Z",
  "eventInsertedBy": "ugc-testing@crossout.app",
  "code": "00000000010b",
  "eventInserted": true
}
```