

API Help

Table of Contents

- [Usage](#)
- [Types](#)
- [DynamicRecordWithKey Fields](#)
- [Reports \(AdvertiserReports\)](#)
- [CPM Advertiser Reports \(CpmAdvertiserReports\)](#)
- [Clicks Reports \(XmlClickReports\)](#)
- [Campaign \(Campaign\)](#)
- [Payment Transactions \(Transactions\)](#)
- [Account \(Account\)](#)
- [IP List \(IpList\)](#)
- [System IP Lists \(IpListsNetwork\)](#)
- [Domain List \(DomainList\)](#)
- [System Domain Lists \(DomainListsNetwork\)](#)
- [App List \(AppList\)](#)
- [System App Lists \(AppListsNetwork\)](#)
- [Offer Ads \(OfferAd\)](#)
- [Offer \(OfferNew\)](#)
- [Offer Field Details](#)
- [Browser Versions \(BrowserVersions\)](#)
- [Carriers \(Carriers\)](#)
- [Geo Countries \(GeoCountries\)](#)
- [Geo Regions \(GeoRegions\)](#)
- [Geo Cities \(GeoCities\)](#)
- [Geo Postal Codes \(GeoPostalCodes\)](#)
- [System Tasks \(SystemTasks\)](#)
- [Referrals By Date \(ReferralsByDate\)](#)
- [Notifications \(Notifications\)](#)
- [Files Service \(files\)](#)

CRUD API is created to automate the process of creating, modifying, deleting, and viewing entries available in the platform's user interface. Also, you can write scripts for automated processing of publisher or advertiser data.

In general, you can create your own tool for a more comfortable workflow specific for your network's needs.

Usage

Authorization

API requires authorization with a generated token. You need to send a GET request to the endpoint

<https://login.myadcampaigns.com/advertiser/auth> with login and password parameters, which you must specify according to the username and password of your account.

Example

<https://login.myadcampaigns.com/advertiser/auth?login=foo&password=bar>

The server responds with a token, which is used for subsequent requests to the API.

<https://login.myadcampaigns.com/advertiser/api/Campaign/?version=4&token=u22L2chc0kD36qCF2rkvUe0ktiBNw2yh.u2>

User Tokens



You can create a user token [here](#) and use it instead of the `token` parameter for additional security and ease of use.

Example

<https://login.myadcampaigns.com/advertiser/api/Campaign/?userToken=userToken>

Login Service

To use the Login Service, you need to send a GET request to the following endpoint

https://login.myadcampaigns.com/advertiser/login_service The **action** parameter is responsible for the service operation mode. The default **action** is **login**. In case of successful authorization, the login service responds with **authToken** which can be used for subsequent API or Login Service requests.

• Login (action=login)

Supported parameters:

- **authToken** - user authorization token. Can be used for subsequent API or Login Service requests.
- **deviceToken** - device authorization token. If the device token is already associated with the user, an authorization attempt from a different device will not require two-factor authorization.
- **login, password** - user login and password.
- **deviceCode, captchaCode** - codes for two-factor authorization and captcha accordingly.
- **newPassword, newPasswordRepeat** - new user password. Can be used to change the password upon authorization.

Examples

https://login.myadcampaigns.com/advertiser/login_service?action=login&login=login&password=password

In case of successful login, the following JSON is returned:

```
{
  "status": "OK",
  "authToken": "authToken",
  "deviceToken": "deviceToken"
}
```

authToken and **deviceToken** parameters can be further used for re-authorization without specifying username and password: https://login.myadcampaigns.com/advertiser/login_service?authToken=authToken&deviceToken=deviceToken

It is recommended to store the **authToken** and **deviceToken** for subsequent requests and resend **login** and **password** only if authorization by **authToken** failed - this means that the validity period of this **authToken** has expired.

In case of an error, Login Service returns JSON response with the following fields:

- **status** - with "Error" as a value
- **errors** - an object that contains error messages for individual fields **fieldErrors** and global errors **globalErrors**
- **captchaSiteKey** - contains a **siteKey** for [ReCaptcha](#) if captcha is required
- **devicePrompt** - contain details for obtaining confirmation code when two-factor authorization is required
- **changePassword - true**, if you need to change the password for the account. In this case, in addition to **login** and **password** it is also necessary to provide **newPassword** and **newPasswordRepeat**

Examples

```
{
  "status": "Error",
  "errors": {
    "globalErrors": [
      "Login or password is incorrect. Please try again\nFor security reasons you have 9 login attempts left"
    ]
  }
}
```

```
{
  "status": "Error",
  "errors": {
    "fieldErrors": {
      "login": [
        "May not be empty"
      ]
    }
  }
}
```



```
    ]
  }
}
```

```
{
  "status": "Error",
  "captchaSiteKey": "siteKey",
  "errors": {
    "fieldErrors": {
      "captcha": [
        "Please verify that you are not a robot"
      ]
    }
  }
}
```

```
{
  "status": "Error",
  "devicePrompt": "Please enter verification code we sent to your email"
}
```

• Password reset (action=reset)

In case the user has lost the password for his account, you can use the password reset option. In this mode, the following parameters are used:

- **email** - e-mail of the user. A new password will be sent to this e-mail

Example

https://login.myadcampaigns.com/advertiser/login_service?action=reset&email=foo@bar.com

If the password is successfully reset, the following JSON will be returned:

```
{
  "status": "OK"
}
```

In case of an error, Login Service returns JSON with the following fields.

- **status** - with "Error" as a value
- **errors** - **globalErrors** object with errors description

Example

```
{
  "status": "Error",
  "errors": {
    "globalErrors": [
      "Invalid e-mail. Check the spelling and try again or contact your network support for further assistance"
    ]
  }
}
```

• Termination of the token (action=logout)

It is possible to terminate the **authToken** that was received by the Login Service. In this mode, the following parameter is used:

- **authToken** - user token to be terminated



If an active **authToken** is terminated, all active UI and API sessions that use it will be closed.

If the action is successful, the following JSON will be returned:

```
{
  "status": "OK"
}
```

• Using the Login Service to Access the UI

The Login Service can be used on the browser side to log in and navigate a user into UI. To do so, additional parameters must be used:

- **redirect** - the path specified relative to <https://login.myadcampaigns.com/advertiser/> where the user will be redirected in case of successful authorization
- **errorRedirect** - URL where the user will be redirected in case of an error. By default, the user will be redirected to <https://login.myadcampaigns.com/advertiser/login>

For example, this method can be used in HTML to build an authorization link:

```
<a href="https://login.myadcampaigns.com/advertiser/login_service?
authToken=authToken&redirect=%23report%2FAdvertiserReports%2Fdate">AdvertiserReports</a>
```

Request format

For querying the API endpoint <https://login.myadcampaigns.com/advertiser/api> is used. When sending queries with a body (POST, PUT), the HTTP header with **Content-Type: application/json** must be used.

Each request is addressed to a particular controller as follows:

https://login.myadcampaigns.com/advertiser/api/controller_name where **controller_name** must match one of the controllers given below. Each request to the API must include the token, which can be obtained through the [authorization service](#).

For GET, PUT, and DELETE requests one or more identifiers of appropriate resources can be specified in order to identify which resource is executed by this query:

<https://login.myadcampaigns.com/advertiser/api/Campaign/1?version=4&token=u22L2chc0kD36qCF2rkVUe0ktiBNw2yh.u2> or
<https://login.myadcampaigns.com/advertiser/api/Campaign/1,2,3,4?version=4&token=u22L2chc0kD36qCF2rkVUe0ktiBNw2yh.u2>

Response Format

The server response is a JSON document that must contain a **status** field. There are two possible statuses:

- **OK** request is processed successfully and the document contains **result** field with the answer
- **Error** request failed. The **message** field contains a message explaining an error.

Request types

• GET

Is used to read resource information. Supported parameters:

- **range** is given in **start-length** format (for example, 0-10, 55-20) and specifies the initial identifier and maximum number of entries in the response. The default value is 0-100. The max length is 200 entries (100000 entries for report controllers). If the specified parameter requests for more entries, only first 200 entries will be returned.
- **ord** is given in **[-]colname** format, where **colname** means ascending order, and **-colname** means descending order by **colname** column.
- **columns** are given in **col1,col2,...** format and indicate a set of columns that will be present in each record which comes in response.
- **filters** are given in **filterName1:filterValue1;filterName2:filterValue2;...** format and specify a set of filters with corresponding values that will be applied to the request.

Response is a structure in which the key is the key of the record, and value is a structure of fields with appropriate values.

Examples



<https://login.myadcampaigns.com/advertiser/api/Campaign/1?version=4&token=u22L2chcOkD36qCF2rkvUeOktiBNw2yh.u2>
Shows information for Campaign with id=1

```
{
  "status": "OK",
  "response": {
    "total": 1,
    "rows": {
      "1": {
        "id": 1,
        "remotefeed_id": 100,
        "name": "Name",
        "pricing_model": "CPC",
        "type": "CPC",
        "attribute": null,
        "description": "description",
        "start_date": null,
        "end_date": null,
        "is_active": true,
        "budget_total": null,
        "budget_limiter_type": "ASAP",
        "cost_today": 8
      }
    }
  }
}
```

https://login.myadcampaigns.com/advertiser/api/Campaign?range=0-10&columns=id,name&filters=pricing_model_filter:CPM&version=4&token=u22L2chcOkD36qCF2rkvUeOktiBNw2yh.u2

Shows first 10 records with `pricing_model_filter` = CPM filter. Response will contain `id`, `name` columns only.

https://login.myadcampaigns.com/advertiser/api/Campaign?range=0-10&columns=id,name&filters=pricing_model_filter:CPM&ord=-bid&version=4&token=u22L2chcOkD36qCF2rkvUeOktiBNw2yh.u2

Shows first 10 records with `pricing_model_filter` = CPM filter ordered by `bid` column. Response will contain `id`, `name` columns only.

• POST

Is used to create a resource. Values for required fields must be present in request body in JSON format.

The result is a structure with a **created** field containing the identifier of the created resource.

Examples

POST request to <https://login.myadcampaigns.com/advertiser/api/Campaign?version=4&token=u22L2chcOkD36qCF2rkvUeOktiBNw2yh.u2> with the following body:

```
{
  "name": "name",
  "type": "DISPLAY"
}
```

creates a new record with specified parameters and returns:

```
{
  "status" : "OK",
  "response" : {
    "created" : 11
  }
}
```

• PUT

Is used to modify a resource. The values for required fields are sent in request body in JSON format. In the case of multiple identifiers all specified resources will be modified. Notice, that only fields with controller field value `mode=BULK_EDITABLE` will be modified.

The result is a structure with the field **updated** which is successfully updated and resources identifiers which were modified.



Examples

PUT request to <https://login.myadcampaigns.com/advertiser/api/Campaign/1,2,3?version=4&token=u22L2chcOkD36qCF2rkVUeOkTiBNw2yh.u2> with request body:

```
{
  "description": "description"
}
```

Changes 'company' field for publishers with id=1,2,3 and returns such a response:

```
{
  "status" : "OK",
  "response" : {
    "updated" : [1,2,3]
  }
}
```

• DELETE

Is used to delete a resource. It is necessary to specify the resource identifiers that will be removed. The result is a structure with **deleted** field which indicates deleted resources identifiers.

Examples

DELETE request to <https://login.myadcampaigns.com/advertiser/api/Campaign/1,2,3?version=4&token=u22L2chcOkD36qCF2rkVUeOkTiBNw2yh.u2> deletes records with id=1,2,3 and returns:

```
{
  "status" : "OK",
  "response" : {
    "deleted" : [1,2,3]
  }
}
```

Reports requests

Reports support GET requests. All parameters for the GET request retain their functionality. In addition, **columns** parameter uses records for the corresponding controller from the **Metrics** table, instead of records from table **Fields**.

Also reporting uses a special syntax for specifying resource:

[https://login.myadcampaigns.com/advertiser/api/REPORT_CONTROLLER\[/DIMENSION=DRILLDOWN_VALUE\]*/DIMENSION\[,DIMENSION\]*](https://login.myadcampaigns.com/advertiser/api/REPORT_CONTROLLER[/DIMENSION=DRILLDOWN_VALUE]*/DIMENSION[,DIMENSION]*), where:

- **REPORT_CONTROLLER** report controller's name;
- **DIMENSION** dimension from **Dimensions** table for the following controller;
- **DRILLDOWN_VALUE** drill-down value for the following report dimension.

Examples

https://login.myadcampaigns.com/advertiser/api/AdvertiserReports/campaign=1/offer=10/date?filters=date:2025-05-01_2025-05-31&version=4&token=u22L2chcOkD36qCF2rkVUeOkTiBNw2yh.u2

Displays report with drill-downs on `campaign` = 1, `offer` = 10 for the current month by date.

https://login.myadcampaigns.com/advertiser/api/AdvertiserReports/campaign,offer?filters=date:2025-05-01_2025-05-31&version=4&token=u22L2chcOkD36qCF2rkVUeOkTiBNw2yh.u2

Displays report by `campaign`, `offer` for the current month.

https://login.myadcampaigns.com/advertiser/api/AdvertiserReports/campaign=1/offer=10/date?filters=date:2025-05-01_2025-05-31&columns=adv_clicks,adv_cost&version=4&token=u22L2chcOkD36qCF2rkVUeOkTiBNw2yh.u2

Displays report for `campaign` = 1, `offer` = 10 by date for the current month. Only `adv_clicks`, `adv_cost` columns will be shown.



Types

Here are the types of values used in the API:

- **Integer, Long, Byte**

(1, 15154, 0)

- **Tagged Item**

Can be Item ID (1, 123, 0), or Tag ID, associated with this item (_1, _123, _0)

- **Double**

(0.2153, 15.55, 1.154E+45)

- **String**

("abc", "", "foo bar baz")

- **Boolean**

Boolean value. In JSON format is indicated as constants (true, false).

- **Enum**

One of a set of values. Specified as a string ("NORMAL", "NEW", null)

- **Date**

Date. Specified as a string in the format YYYY-MM-DD ("2014-11-30", "2012-1-21")

- **DateRange**

Date range. Specified as a string in the format "YYYY-MM-DD_YYYY-MM-DD" ("2025-05-01_2025-05-31")

- **YearMonth**

Year-Month. Specified as a string in the format YYYY-MM ("2014-11")

- **Size**

Size. Specified as a string in the format WIDTHxHEIGHT ("300x250")

- **List**

List. Specified as a JSON list with values in the following format ([1,2,3,4,5], ["ua", "us", "ca"], [])

- **GeoLocationValue**

Geo location value is a structure that contains the following fields: **groups**, **countries**, **states**, **cities**, **zips** which contain lists of identifiers of corresponding entities. To create a structure, you can use [Constructor](#)

- **UploadValue**

UploadValue is a field type that represents a file as a value.

When you are trying to **get** data from **UploadValue** fields, they will contain **File ID**.

You can download files using [Files Service](#) by sending this **File ID** as a GET request parameter.

There are two ways to **set** a file as a value:

- Use the uploaded **File ID** provided by [Files Service](#) as a value.
- Add file data encoded in Base64 (UTF-8 charset) as field value. The value must be set as JSON string.

The received **File ID** will be available for 10 minutes.

- **ImageValue**

ImageValue is a field type that represents an image file as a value.

It is a structure that contains the following fields:

- **width** ([Integer](#)) width of the image.
- **height** ([Integer](#)) height of the image.
- **type** ([Enum](#) [MAIN, ICON]) type of the image.
- **image** ([UploadValue](#)) image file itself.

All fields are required.

- **VideoValue**

VideoValue is a field type that represents a video file as a value.

It is a structure that contains the following fields:

- **width** ([Integer](#)) width of the video.
- **height** ([Integer](#)) height of the video.



- **bitrate** ([Integer](#)) video bitrate.
- **duration** ([Integer](#)) video duration.
- **type** ([Enum](#) [MP4, AVI, WEBM, QUICKTIME, MPEG]) type of the video.
- **filename** ([String](#)) video file name.
- **video** ([UploadValue](#)) video file itself.

All fields are required.

• DynamicRecordWithKey

It is a structure that contains **key** and **value** fields. More info [here](#)

- Field **key** usually contains the ID of the parent record, e.g. Offer ID.
- Field **values** contains the structure of the fields set.

• DayTimeValue

It is a structure that represents targeting by day of week and time period in one value.

Key defines day and time period in format **DAY_NAME:PERIOD**. For example:

- **SUNDAY:1** - Sunday 1:00-2:00
- **MONDAY:16** - Monday 16:00-17:00

Value is a boolean value that defines whether a certain period is enabled or not. If some periods are not presented they enabled by default.

• AdQualitySystemType

One of a set of Ad Quality Systems - TMT, GEO_EDGE, ADJESTY, AD_SECURE. Defined as a string ("GEO_EDGE", "TMT", null)

DynamicRecordWithKey Fields

To **update** values in this field you must use the following format:

```
{
  "mode": [REPLACE, UPDATE],
  "create": [...],
  "edit": [...],
  "delete": [...]
}
```

- **mode** (required) - indicates mode which can be used for data modification.
REPLACE mode removes old entries and saves only new ones. This mode allows only the **create** section to be used.
UPDATE mode stores older data and modifies existing entries according to updated values.
- **create** - This field contains data that will be added as new records.
- **edit** - This field contains data that are to be modified.
- **delete** - This field contains data that are to be removed.

Fields **create**, **edit**, and **delete** are represented as lists of structures. Each structure must contain fields presented in the table above.

For modification or removing records each structure must contain key fields with original values.

Original values in key fields cannot be changed.

Lightweight Format Usage

Instead of using our common data format in the **delete** section:

```
{
  "update": "UPDATE",
  "delete": [{"id":1}, {"id":2}, {"id":3}]
}
```

you can use a lightweight format:

```
{
  "mode": "UPDATE",
  "delete": [1,2,3]
}
```

