AdsExplorer API

AdsExplorer API document outlines integration for the two main system tasks: scanning tasks (incoming API requests) and client notifications (outgoing API requests).

The format used to transfer data in all types of integrations — JSON, protocol — HTTPS.

All types of integrations must be authorized via authorization key (API token) which can be found in your personal settings.

Task Request (incoming API request)

Authorization and error codes

The client authentication with incoming API request is used with HTTP header **X-AdsExplorer-Api-Token** which transfers the client's API token. The API does not use sessions, cookies or CSRF tokens.

The responsive HTTP header **X-AdsExplorer-Status** is used to identify the status of the request which contains the status line. HTTP-codes are not used. Any processed request will always contain HTTP-code 200 (success).

The status of the request is always displayed in the **status** field of the response body.

The list of possible incoming request statuses:

- **OK**: Request completed successfully.
- **ERROR**: Request failed due to internal error.
- **NOT_FOUND**: Requested API-method or resource was not found.
- **USER_ERROR**: Request consist errors in parameters or the limit for incoming requests has been exceeded.
- **MAINTENANCE**: AdsExplorer is not accepting requests due to maintenance.
- **FORBIDDEN**: Request lacks header **X-AdsExplorer-Api-Token** or header contains an invalid API_TOKEN.
- **NOT_MODIFIED**: Nothing has changed from the last request. For example, when user passes HTTP header If-None-Match containing value, equal current ETag.

Additional information about the error reasons can be found in the **data** or **description** fields of the response body.

Directory Update

The data from directories is used when creating scanning tasks; for example, a list of active countries (Proxy) and a list of supported devices. The client could occasionally make requests for current versions of the directories via API, so that when creating tasks, the check parameters are up to date. The frequency of directory updates depends on the clients needs. Since the main and most used countries and devices are always available, there is no need for frequent directory update unless there is a request for scans with rare parameters. The directory should be updated once per hour or occasionally in order to create most of the tasks.

The following API-request should be used in order to get latest directories for task creation:

GET https://adsexplorer.com/task/create/?projectName=PROJECT

projectName property must contain name of an existing project. If project in **projectName** doesn't exist, first available project would be used.

To cache directories data and check their variability in time, use the HTTP header **ETag** containing the MD5 hash of the response body.

HTTP request example:

```
GET /task/create/?projectName=PROJECT HTTP/1.1

Host: adsexplorer.com
X-AdsExplorer-Api-Token: CLIENT_API_TOKEN
Connection: close
```

PHP request example:

```php
<?php
$ch = curl_init('https://adsexplorer.com/task/create?projectName=Project');

curl_setopt_array($ch, [
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_HTTPHEADER => ['X-AdsExplorer-Api-Token: CLIENT_API_TOKEN']
]);

$result = curl_exec($ch);

if (curl_error($ch)) {
    // Error handling.
}

curl_close($ch);
```

Responses

On successful request “OK” status and directory data will be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: OK
Etag: RESPONSE_HASH

{
  "status":"OK",
  "data":{
    "devices":[
      {
        "type":"DESKTOP",
        "os":"Windows",
        "browser":"Edge"
      },
      {
        "type":"MOBILE",
        "os":"Android",
        "browser":"UC Browser"
      },
    ],
    "countries":{
      "HOSTED":[
        "AF",
        "ZW"
      ],
      "RESIDENTIAL":[
        "AF",
        "ZM",
      ],
      "CARRIER":[
        "AL",
        "YE",
      ]
    },
    "isps":[
      {
        "id":1,
        "text":"AE Emirates ISP"
      },
```

```
      {
        "id":2,
        "text":"AO Acme"
      },
    ],
    "checkTypes":[
      "TDS",
      "POPUNDER",
      "ADTAG",
      "CUSTOM",
      "VAST"
    ],
    "networkTypes":[
      "RESIDENTIAL",
      "CARRIER",
      "ISP"
    ],
    "projects":[
      "Main",
      "Marketing"
    ],
    "notifications":{
      "EMAIL":{
        "compliance@example.com":[

        ]
      },
      "SLACK":{
        "marketing_chat":[

        ]
      },
      "API":{
        "My_API_notifications":[

        ]
      }
    }
  }
}
```

If authorization header is omitted or **CLIENT_API_TOKEN** is wrong following response would be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: NOT_AUTHORIZED
{
  "status": "FORBIDDEN",
  "description": "NOT_AUTHORIZED",
  "data": {}
}
```

In case of a general unrecoverable error, following response would be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: ERROR
{
    "status": "ERROR",
    "data": {}
}
```

Creating a task request

The following request is used to create a task via API:

POST https://adsexplorer.com/task/create/

The body of the request contains the following JSON parameters:

- **projectName**
  - Type: String
  - Always required
  - Name of the project where task would be created. If this project doesn't exist, first available project would be used.
- **url**
  - Type: String
  - Always required.
  - Full URL of the target page.
- **referrer**
  - Type: String
  - Always required.
  - Full URL of the previous page (HTTP Referer). If it is empty, **url** would be used instead.
- **checkType**
  - Type: String
  - Always required
  - Possible options can be acquired from Directory update.
- **networkType**
  - Type: String
  - Always required
  - Possible options can be acquired from Directory update.
- **countryId**
  - Type: String
  - Always required
  - Country code from ISO 3166-1 alpha-2.
- **ispId**
  - Type: Integer
  - Required only when networkType is "ISP".
  - Can be acquired from Directory update.
- **click**
  - Type: String
  - Required only when checkType is "CUSTOM".
  - Any CSS-selector, for example "#id" or ".class".
- **adTag**

  - Type: Boolean
  - Required only when checkType is “ADTAG”.
  - AdTag HTML code.
- **deviceType**
  - Type: String
  - Always required
  - Can be acquired from Directory update.
- **deviceOs**
  - Type: String
  - Always required
  - Can be acquired from Directory update.
- **deviceBrowser**
  - Type: String
  - Always required
  - Can be acquired from Directory update.
- **checkLoops**
  - Type: Integer
  - Always required
  - Number of check repeats within one task. From 1 to 15.
- **isCleanCache**
  - Type: Boolean
  - Optional
  - Should cache be cleaned between check loops. Can be used only when checkLoops is 2 or more.
- **isCleanCookie**
  - Type: Boolean
  - Optional
  - Should cookies be cleaned between check loops. Can be used only when checkLoops is 2 or more.
- **isFullpageScreenshot**
  - Type: Boolean
  - Always required
  - Make a full page screenshot and detect banners.
- **isNewWindowRequired**
  - Type: Boolean
  - Optional. Can be used only when networkType is “CUSTOM”.
  - Default is true. If set to true, task would fail if new window was not opened.
- **isReclick**
  - Type: Boolean
  - Optional.
  - Should click be retried if new window does not open.
- **isAdTagClick**
  - Type: Boolean
  - Optional.
  - Should Ad Tag be clicked.
- **isCheckWithAdxMetaEngine**
  - Type: Boolean
  - Optional.
  - If true, content of the target page with be checked through ADx Meta Engine. Beware, such tasks are covered with an extra fee.
- **labels**
  - Type: JSON object
  - Optional.
  - Collection of user labels which would be attached to task. Must be a valid JSON object.
- **notifications**
  - Type: JSON object
  - Optional.
  - Used for personal task notification settings. If the object is empty, all notifications are disabled.

HTTP request example:

```
POST /task/create/ HTTP/1.1
Host: adsexplorer.com
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Api-Token: CLIENT_API_TOKEN
Connection: close

{
  "checkType":"POPUNDER",
  "networkType":"ISP",
  "countryId":"US",
  "ispId":1,
  "projectName":"Marketing",
  "deviceType":"DESKTOP",
  "deviceOs":"Windows",
  "deviceBrowser":"Chrome",
  "checkLoops":3,
  "isCleanCache":false,
  "isCleanCookie":false,
  "isFullpageScreenshot":true,
  "url":"http://example.com/",
  "referrer":"http://example.com/promo",
  "labels":{
    "Comment":"New landing",
    "ManagerID":"1337"
  },
  "notifications":{
    "EMAIL":{
      "compliance@example.com":{
        "AV_DOMAIN":true,
        "GSB_DOMAIN":true
      }
    },
    "SLACK":{
      "marketing_chat":{
        "AUDIO":true,
        "STORE_REDIRECT":true
      }
    },
    "API":{
      "My_API_notifications":{
        "STATUS_CHANGE":true
      }
    }
  }
}
```

PHP request example:

```php
<?php

$ch = curl_init('https://adsexplorer.com/task/create');

$requestBody = json_encode([
    'url' => 'https://example.com',
    'referrer' => 'https://example.com',
    'checkType' => 'TDS',
    'networkType' => 'RESIDENTIAL',
    'countryId' => 'US',
    'projectName' => 'Project',
    'deviceType' => 'DESKTOP',
    'deviceOs' => 'Windows',
    'deviceBrowser' => 'Chrome',
    'checkLoops' => 1,
    'isCleanCache' => false,
    'isCleanCookie' => false,
    'isFullpageScreenshot' => false
]);

// JSON encoding error handling.

curl_setopt_array($ch, [
    CURLOPT_POST => true,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_POSTFIELDS => $requestBody,
    CURLOPT_HTTPHEADER => [
        'Content-Type: application/json',
        'X-AdsExplorer-Api-Token: CLIENT_API_TOKEN'
    ]
]);

$result = curl_exec($ch);

if (curl_error($ch)) {
    // Error handling.
}

curl_close($ch);
```

Responses On successful request “OK” status, code “TASK_CREATED” in the **description** field, information about newly created task and new amount of user balance will be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: OK

{
  "status":"OK",
  "description":"TASK_CREATED",
  "data":{
    "taskId":1337,
    "balanceAmount":100,
    "createTime":"2017-11-07T18:53:30.339978+00:00"
  }
}
```

If authorization header is omitted or **CLIENT_API_TOKEN** is wrong following response would be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: NOT_AUTHORIZED
{
  "status": "FORBIDDEN",
  "description": "NOT_AUTHORIZED",
  "data": {}
}
```

If parameters have errors in them, following response would be returned, where data.errors contains object with problematic parameters as keys and error reason as values. Beware, “EMPTY” may mean either mean omitted parameter or incorrect value.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: OK
{
    "status": "OK",
    "data": {
        "errors": {
            "networkType": "EMPTY",
            "countryId": "EMPTY",
            "deviceType": "EMPTY",
            "deviceOs": "EMPTY",
            "deviceBrowser": "EMPTY",
            "checkLoops": "EMPTY",
            "url": "EMPTY",
            "referrer": "EMPTY",
            "checkType": "EMPTY"
        }
    }
}
```

If user’s balance is empty, following response would be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: USER_ERROR
{
    "status": "USER_ERROR",
    "data": {},
    "description": "NOT_ENOUGH_FUNDS"
}
```

If chosen device combination (deviceOs, deviceBrowser and deviceType) is no longer available, following response would be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: USER_ERROR
{
    "status": "USER_ERROR",
    "data": {},
    "description": "DEVICE_NOT_FOUND"
}
```

If task creation is limited for some reason, following response would be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: USER_ERROR
{
    "status": "USER_ERROR",
    "data": {},
    "description": "API_LIMIT_EXCEEDED"
}
```

If your tariff does not allow creation of these type of tasks, following response would be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: USER_ERROR
{
    "status": "USER_ERROR",
    "data": {},
    "description": "NOT_ALLOWED_BY_TARIFF"
}
```

If project with these name is no longer available, following response would be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: USER_ERROR
{
    "status": "USER_ERROR",
    "data": {},
    "description": "PROJECT_NOT_FOUND"
}
```

If task has **checkType** “ISP”, but parameter **ispId** doesn’t contain a valid ISP ID, following response would be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: USER_ERROR
{
    "status": "USER_ERROR",
    "data": {},
    "description": "INVALID_ISP"
}
```

In case of a general unrecoverable error, following response would be returned.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: ERROR
{
    "status": "ERROR",
    "data": {}
}
```

Incoming request restrictions

There is a speed restriction when creating tasks via API: no more than 1 task per second. In this case, the short-term surges may occur and last no more than a few minutes, while maintaining the average scanning speed within the indicated limits. If the limits are exceeded, a client will get the USER_ERROR status and "API_LIMIT_EXCEEDED" code.

The limits may change depending on the pricing plan. Please contact us for more information about plans: support@adsexplorer.com

Client notifications (outgoing API requests)

AdsExplorer can notify client's services via outgoing API requests. There are two types of notifications available: suspicious activity detection or task status update.

Creating a channel for API notifications

To connect API notifications, a client must create one or more notification channels in their settings ("API notification channels" section). A client must specify the full URL which will be used by AdsExplorer to send API requests. It is mandatory to use HTTPS scheme for API notifications. The receiving server must have valid certificates (self-signed and non-verified certificates are not allowed). When the notification channel is created, it may be used in the project notification settings or in custom settings for campaign notifications or a specific task (including task generation via API).

Identification of API requests and error processing

To make sure that the request for the client's URL was made by AdsExplorer, the client must check the HTTP-header **X-AdsExplorer-Signature**, which comes with the request. This header will contain a SHA256 hash from the line of the client's API token and the original JSON request body. If the specified header is missing or contains an invalid hash this may indicate a fake request. However, the header may not be delivered to the client due to the wrong settings of the client's server / balancer. If you can not validate AdsExplorer requests using the caption header, please, contact our technical support: support@adsexplorer.com

AdsExplorer makes **only one attempt** to send an API notification and does not check the delivery status in any way. If notification was not sent for any reason, it resets and is never sent. If you stopped receiving API notifications, please, check the availability of your URL address from the outside, as well as the validity of your certificates.

The format of outgoing API notifications

A client will be able to receive the events of the task status change when subscribing to the "Task status updates" (For example: pending, completion of the analysis and other). This type of notifications is only available for API channels.

```
{
  "type":"STATUS_CHANGE",
  "taskId":1337,
  "status":"COMPLETED",
  "timestamp":"2017-11-07T18:53:30.339978+00:00"
}
```

A client may also subscribe to detection events for any selected triggers. Part of the data of the task is sent together with these events, for example: user-defined text labels. This event can appear several times for one task.

```
{
  "type":"TRIGGER_ALERT",
  "task":{
    "taskId":1337,
    "createTime":"2017-11-07T18:53:30.339978+00:00",
    "updateTime":"2017-11-07T18:55:30.834563+00:00",
    "status":"COMPLETED",
    "checkType":"POPUNDER",
    "networkType":"RESIDENTIAL",
    "countryId":"US",
    "projectName":"Marketing",
    "deviceType":"DESKTOP",
    "deviceOs":"Windows",
    "deviceBrowser":"Chrome",
    "deviceName":"Desktop, Windows 10, Chrome 61.0",
    "url":"http://example.com/",
    "referrer":"http://example.com/promo",
    "labels":{
      "Comment":"New landing",
      "ManagerID":"1337"
    },
    "publicLink":"https://adsexplorer.com/report/?id=1337"
  },
  "triggers":{
    "LOCKER":[
      {
        "code":"LOCKER",
        "url":"http://example.com/",
        "data":{
          "type":"JS"
        }
      }
    ]
  }
}
```


Checking reaction of Google Safe Browsing on a domain

This endpoint can be used to check is domain considered malicious by Google Safe Browsing right now.

GET https://adsexplorer.com/domain/is-flagged/gsb?domain=example.com&callbackUrl=https://foo.bar

Query string should contain following parameters:

- **domain**
  - domain you wish to check
- **callbackUrl**
  - URL to which would be sent a message upon check's completion.

HTTP request example:

```
GET /domain/is-flagged/gsb?domain=example.com&callbackUrl=https://foo.bar/callback HTTP/1.1
Host: adsexplorer.com
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Api-Token: CLIENT_API_TOKEN
Connection: close
```

PHP request example:

```php
<?php
$parameters = http_build_query([
    'domain' => 'example.com',
    'callbackUrl' => 'https://foo.bar/callback'
]);

$ch = curl_init("https://adsexplorer.com/domain/is-flagged/gsb?{$parameters}");

curl_setopt_array($ch, [
    CURLOPT_HTTPHEADER => [
        'Content-Type: application/json',
        'X-AdsExplorer-Api-Token: CLIENT_API_TOKEN'
    ]
]);

$result = curl_exec($ch);

if (curl_error($ch)) {
    // Error handling.
}

curl_close($ch);
```

In case of an error response would resemble following.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: USER_ERROR

{
    "status": "USER_ERROR",
    "data": {
        "errors": {
            "callbackUrl": "Callback URL is empty."
        }
    },
    "description": "INVALID_URL"
}
```

Key status contains a general error code.

- **USER_ERROR**
  - There is an error in one of the parameters.
- **FORBIDDEN**
  - This action is not available right now or ever.

Key data.errors would contain a problematic parameter and description of an error.

Key description would contain an error code of a specific error.

- **NOT_AUTHORIZED**
  - Authorization header was not passed.
- **INVALID_DOMAIN**
  - Where was an error in domain.
- **INVALID_URL**
  - Where was an error in callback URL.
- **ALREADY_IN_PROGRESS**
  - Checking of this domain is already in progress.

If everything is OK response would be following.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: OK

{
  "status":"OK",
  "data":{}
}
```

Upon finishing of the domain check POST request would be sent to the passed callbackURL. It would look like this:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8

{
  "domain": "example.com",
  "isFlagged": true
}
```

Where domain is the domain you passed and isFlagged determines is this domain flagged by the GSB.

Checking is domain was ever flagged as malicious by antivirus or Google Safe Browsing

You can use this endpoint to check had we ever recorded a malicious activity on a domain. Domain is considered flagged, if any of antivirus software used by us or Google Safe Browsing ever considered this domain as malicious.

GET https://adsexplorer.com/domain/is-flagged/ever?domain=example.com

Query string should contain domain you wish to check as a parameter.

HTTP request example:

```
GET /domain/is-flagged/ever?domain=example.com HTTP/1.1
Host: adsexplorer.com
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Api-Token: CLIENT_API_TOKEN
Connection: close
```

PHP request example:

```
<?php
$domain = 'example.com';

$ch = curl_init("https://adsexplorer.com/domain/is-flagged/ever?domain={$domain}");

curl_setopt_array($ch, [
    CURLOPT_HTTPHEADER => [
        'Content-Type: application/json',
        'X-AdsExplorer-Api-Token: CLIENT_API_TOKEN'
    ]
]);

$result = curl_exec($ch);

if (curl_error($ch)) {
    // Error handling.
}

curl_close($ch);
```

In case of an error response would resemble the following.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: USER_ERROR

{
    "status": "USER_ERROR",
    "data": {
        "errors": {
            "domain": "Domain is not empty."
        }
    },
    "description": "INVALID_DOMAIN"
}
```

Keys status and description would always be as in example. Key data.errors would contain a detailed descriptino of an error.

If everything is OK response would be following.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-AdsExplorer-Status: OK

{
  "status":"OK",
  "data":{
    "is_triggered": true
  }
}
```

Backward compatibility Guidelines

In order to keep backward compatibility of existing API-integrations and retain flexibility in API development, AdsExplorer has adopted certain rules. The following API changes are considered backward compatible and are not needed to be approved by client:

- to delete the parameters that are no longer in use in the incoming requests;
- to add new parameters in the outgoing requests or responses to incoming requests;
- to add new optional parameters in the incoming requests.

These rules also apply to the nested object parameters, should the rest of the object signature remains untouched.

The following list of changes is considered backward incompatible and will be released via new API methods without interrupting the old ones: existing parameter format change, change of logic of a parameter or group of parameters, adding a new mandatory parameter in incoming requests, and other changes.