

# 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:

<b>OK</b>	Request completed successfully
<b>ERROR</b>	Request failed due to internal error
<b>FORBIDDEN</b>	Request failed due to limited access or authorization error
<b>NOT_FOUND</b>	Requested API-method or resource was not found
<b>USER_ERROR</b>	Request consist errors in parameters that need to be corrected by the client or the limit for incoming requests has been exceeded
<b>MAINTENANCE</b>	AdsExplorer is not accepting requests due to maintenance

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
```

The **projectName** field contains a name of a current project. HTTP request example:

```
GET /task/create/?projectName=PROJECT HTTP/1.1
Host: adsexplorer.com
X-AdsExplorer-Api-Token: CLIENT_API_TOKEN
Connection: close
```

The FORBIDDEN status will return in case of authorization error:

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

{
  "status": "FORBIDDEN",
  "description": "NOT_AUTHORIZED",
  "data": {}
}
```

The OK status and directory data will return in case of successful authorization:

```
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": "DESKTOP",
      "os": "macOS",
      "browser": "Safari"
    },
    {
      "type": "MOBILE",
      "os": "Android",
      "browser": "UC Browser"
    },
    {
      "type": "MOBILE",
      "os": "iOS",
      "browser": "Safari"
    },
    {
      "type": "TABLET",
      "os": "iOS",
      "browser": "Safari"
    },
    {
      "type": "TABLET",
      "os": "Android",
      "browser": "Chrome"
    },
    {
      "type": "TV",
      "os": "Windows",
      "browser": "Edge"
    }
  ],
  "checkTypes": [
    "TDS",
    "POPUNDER",
    "CUSTOM"
  ],
  "networkTypes": [
    "HOSTED",
    "RESIDENTIAL",
    "CARRIER"
  ],
  "countries": {
    "HOSTED": [
```

```

        "AF",
        "AX",
        "AL",
        // ...
        "YE",
        "ZM",
        "ZW"
    ],
    "RESIDENTIAL": [
        "AL",
        "DZ",
        "AO",
        // ...
        "YE",
        "ZM",
        "ZW"
    ],
    "CARRIER": [
        "AL",
        "DZ",
        "AR",
        // ...
        "UY",
        "VE",
        "VN"
    ]
},
"projects": [
    "Main",
    "Marketing"
],
"notifications": {
    "EMAIL": {
        "compliance@example.com": [],
    },
    "SLACK": {
        "marketing_chat": []
    },
    "API": {
        "My_API_notifications": []
    }
}
}
}

```

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

## 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:

<b>checkType</b>	String	Possible options: "TDS", "POPUNDER", "CUSTOM".
<b>click</b>	String	It's optional and can be used only for CUSTOM checks. Any CSS-selector, for example "#id" or ".class".
<b>networkType</b>	String	Possible options: "HOSTED", "RESIDENTIAL", "CARRIER".
<b>countryId</b>	String	Country code for ISO 3166-1 alpha-2.
<b>projectName</b>	String	The name of the Project where the task is created.
<b>deviceType</b>	String	Possible options: "DESKTOP", "MOBILE", "TABLET", "TV".
<b>deviceOs</b>	String	Possible options: "Windows", "macOS", "Android", "iOS", "*nix", "Other".
<b>deviceBrowser</b>	String	Possible options: "Chrome", "Firefox", "IE", "Edge", "Opera", "Safari", "UC Browser", "Other".
<b>checkLoops</b>	Integer	Number of checks on repeat within one task. From 1 to 5.
<b>isCleanCache</b>	Boolean	It's optional and can be used only with two or more recurrences.
<b>isCleanCookie</b>	Boolean	It's optional and can be used only with two or more recurrences.
<b>isFullpageScreenshot</b>	Boolean	Make a full page screenshot and detect banners.
<b>url</b>	String	Full URL of the checked page.
<b>referrer</b>	String	Full URL of the previous page (HTTP Referer). It could be displayed as an empty line which can mean the same URL from the checked page.
<b>labels</b>	Object	It's optional and may be left empty. Random collection of user labels: check name, responsible manager name, internal client ID. The length and

		amount of labels are not limited.
<b>notifications</b>	Object	It's optional. Used for personal task notification settings. If the object is empty, all notifications are disabled.

Example of HTTP request:

```
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": "RESIDENTIAL",
  "countryId": "US",
  "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
      }
    }
  }
}
```

```
}  
}  
}
```

The status OK, the code "TASK\_CREATED" in the **description** field, as well as the data of the new task and new amount of user balance will be returned if the task is completed correctly:

```
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"  
  }  
}
```

## 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](mailto: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](mailto: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, due to some triggers are analyzed in several separate threads.

```
{
  "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"
        }
      }
    ]
  }
}

```

## 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.