Group-IB TI&A Integration Guide

APIv2 is a program interface for obtaining data designed to integrate the Group-IB Threat Intelligence & Attribution with the client's internal security and antifraud systems. APIv2 uses the RESTful protocol, and all responses are returned in JSON format.

Content

How to start

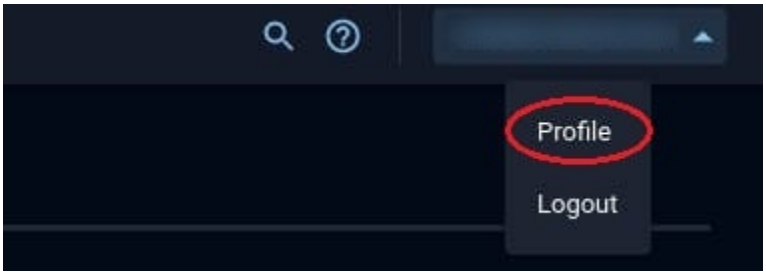
To access the API you need to:

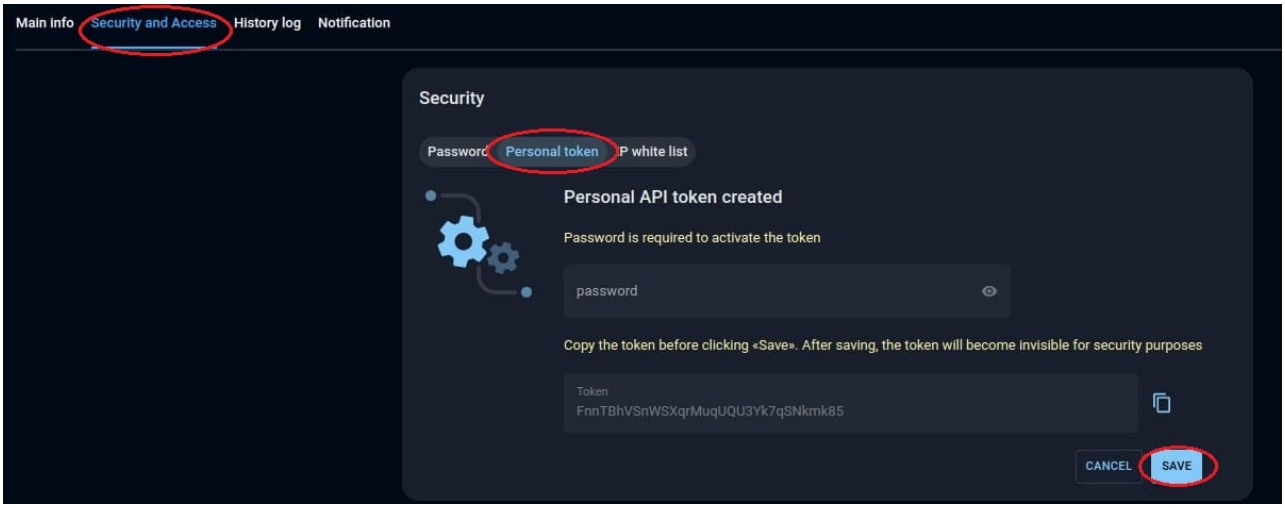
1. Provide your public IP addresses so we can whitelist them for API access.
2. Log in to your account, **generate** and **save** the API_KEY in your **Profile**.
3. Use something that will help you to automate work with the REST API.

API key generation

To generate the API_KEY go to the Group IB TI&A:

- In the new interface (https://tap.group-ib.com/): click on your name in the right upper corner -> choose **Profile** option -> switch to **Security and Access** tab -> click **Personal token** -> follow instructions to generate API token.
- In old interface (https://bt.group-ib.com/): click on your name in right upper corner -> choose **Profile** option -> click on **Go to my setting** button under your name -> under **Change password** button you will see **API KEY generator** -> enter your password, click **Generate**, then click **Save**.
- **Do not forget to save the API key**.





Authentication

For authentication, we use basic authentication.

Curl example

```
curl 'https://tap.group-ib.com/api/v2/sequence_list' -u 'LOGIN:API_KEY' -H
'Accept: */*'
```

Note: **LOGIN** is your username to log on to the Group IB portal. **API_KEY** key you have generated on the previous step.

Rest API Scope

There are several ways to get feeds:

1. Global search
2. Receive only new data
3. Receive up-to-date data
4. Get a feed by ID

1. Global search

It is possible to perform a global search across all collections.

Request format

```
curl 'https://tap.group-ib.com/api/v2/search?q=8.8.8.8' -u 'LOGIN:API_KEY'
-H 'Accept:*/*'
```

Params:

Name	Type	Required	Description
q	string	true	Search query

Response example

```
[
    {
        "apiPath": "attacks/ddos",
        "count": 389,
        "detailedLinks": [],
        "label": "Attack :: DDoS",
        "link": "https://bt.group-ib.com/attacks/ddos?searchValue=%
228.8.8.8%22&q=%228.8.8.8%22"
    },
    {
        "apiPath": "attacks/deface",
        "count": 1,
        "detailedLinks": [],
        "label": "Attack :: Deface",
        "link": "https://bt.group-ib.com/attacks/deface?searchValue=%
228.8.8.8%22&q=%228.8.8.8%22"
    },
    ...
]
```

Global search simplifies the process of searching through all collections and indicates which collections generally have matching entries. However, to get the found items, you need to perform the same search query in the collection specified in the **apiPath** parameter using one of the methods described below.

2. Receive only new data

This method allows you to search feeds sorted by detection date in descending order.

Request format

```
GET <group-ib-url>/api/v2/<collection_name>
```

Params:

Name	Type	Required	Description
q	string	false	Search query
df	datetime	false	Date from
dt	datetime	false	Date to
resultId	string	false	Result ID to get the next data chunk
limit	integer	false	Feed portion size

Note: datetime should be in one of those formats: YYYY-MM-DD; YYYY-MM-DDThh:mm:ssZ. For example, 2021-11-11.

Request example

```
curl 'https://tap.group-ib.com/api/v2/compromised/account?limit=10&df=2019-01-01&dt=2020-01-01&q=John' -u 'LOGIN:API_KEY' -H 'Accept:*/*'
```

Response example

```
{
    "resultId": "6ee8e3a568b89a48915f99aa489128febb92d7ee",
    "count": 183,
    "items": [
        {
            "client": {
                "ipv4": {
                    "ip": "10.10.10.10",
                    ...
                }
            },
            "cnc": {
                "domain": "some.online",
                "ipv4": {
                    "ip": "10.10.10.10",
                    ...
                },
                "url": "http://some.online/",
                ...
            },
            "dateCompromised": "2019-12-17T20:32:38+00:00",
            "dateDetected": "2019-12-17T18:35:25+00:00",
            "login" : ...,
            ...
        },
        {
            ...
        }
    ]
}
```

Procedure:

1. Send a search query. In response, you will receive a valid JSON document with the following fields:
   "count" - amount of results found
   "resultId" - id of your search query
   "items" - list of items
2. Use **resultId** as a param to get the next batch of feeds:

Request example

```
GET /api/v2/compromised/account?resultId=<resultId>
```

3. Repeat the request until the items list is not empty.

Notes: "resultId" lives for 10 minutes since the last call. If a call fails, you will not be able to use this "resultId" anymore.

3. Receive up-to-date data

This method allows you to search feeds sorted by their **seqUpdate** value in ascending order. Every time feed is changed, the **seqUpdate** of this feed will increase, and this feed will appear on the top of the list. This parameter is saved on the integration side and is used to request relevant information.

For all feeds in the Group IB TIA continuous numbering is carried out. For example, the **seqUpdate** equal to 1999998 can be in the **compromised/accounts** collection, and a feed with **seqUpdate** equal to 1999999 can be in the **attacks/ddos** collection. This information is needed to understand what this parameter means for the enrichment, it does not play a special role.

Warning: you can't use this method for compromised/breached and compromised/reaper.

Warning: if you use only **df** or **dt** parameters, data will be sorted by detection date (similarly to Receive up-to-date data), but in ascending order. Because of this you should **always** use **sequence_list** request to get a initial proper **seqUpdate** (like described in **procedure** section below) and not the /updated request with **df**.

Request format

```
GET <group-ib-url>/api/v2/<collection>/updated
```

Params:

Name	Type	Required	Description
q	string	false	Search query
df	datetime	false	Date from
dt	datetime	false	Date to
seqUpdate	integer	false	Result ID to get the next data chunk
limit	integer	false	Feed portion size

Note: datetime should be in one of those formats: YYYY-MM-DD, YYYY-MM-DDThh:mm:ssZ. For example, 2021-11-11.

Procedure:

1. Use a **sequence_list** request to get an update for a specific date to start the session. Date should be in this format: YYYY-MM-DD

Request format

```
GET /api/v2/sequence_list?date=<date_param>
```

2. Received **seqUpdate** can be used as a parameter for further requests. When you submit this argument as a parameter, you will get a portion of feeds starting with the next **seqUpdate** for this collection.

Request format

```
GET /api/v2/<collection>/updated?seqUpdate=<seqUpdate_param>
```

3. To get the next portion of the feeds, you need to find the **seqUpdate** of the last item in the received batch.
4. Repeat paragraph 3, while the **count** field is not equal to zero,
5. Save the last received **seqUpdate** for future reference after each iteration.

4. Get a feed by ID

Each item in the system has its ID.
It is accessible by the **id** field.

Request example

```
GET /api/v2/<collection>/<id>
```

Useful stuff

Status codes:

- **400**: Bad request. Check the error message in response JSON for more information.
- **401**: Bad credentials, you should use Basic Auth with your username and API key, check that you are using the correct one.
- **403**: Something is wrong with your account, please, contact GIB.
- **404**: Not found. There is no such data on the server.
- **500**: There are some troubles on the server with your request. Retry later.
- **301-302**: Verify that your public IP is whitelisted by Group IB.
- **429**: Maximum count of requests per second reached. You should lower your request rate, e.g. by increasing the limit parameter.

Queries:

- You can use complex queries from the TI&A Web Interface, e.g.: company_id: "Company" AND status: "active".

Specific features in some collections:

- For compromised/reaper collection requests can return only up to 10.000 entries in total, and you should set such queries and dates so you will get less than 10.000 entries.

Misc:

- You can find additional information, such as JSON response schemas or collections limits (you can't set limit higher than the server limit) in other tabs.