

Product Platform: **All platforms**
Product Version: **All versions**
Document Date: **4 April 2025**
Document Version: **2.0**

API Overview

Public API

We offer a free REST API as part of your licensing. You can use the API to get your data into to your SIEM (security information and event management) system, such as Splunk or Power BI. Alternatively, you can offload the data into your own backend or integrate seamlessly into your ticketing system. You can read all data you can see in the portal and approve or deny requests through the API.

Pull (API) versus Push (WebHooks)

There are two ways you can get a copy of your data from us. The most typical way to get data is for you to call our public API in interval to download data (pull from your side). You can also subscribe to WebHooks (push from our side).

WebHooks

The advantage of WebHooks is that you get your data in real-time instead of pulling data in intervals. The drawback is that you have to have a public webserver to retrieve the data and you would still need the API to approve requests.

There are also many SaaS systems that allow you to integrate with WebHooks – such as getting notified in Teams or Slack by pointing a WebHook from Admin By Request to such a system.

WebHooks and public API have the same data structure. The difference is only in the delivery model – Push vs Pull. You configure WebHooks in your settings under **Settings > Tenant Settings > Webhooks > WEBHOOKS**.

API

If you use the API instead of WebHooks, you must first enable the API and set up an API key. This is done by logging into your portal account and navigating to **Settings > Tenant Settings > API Keys > API KEYS**, then clicking **Add New**:

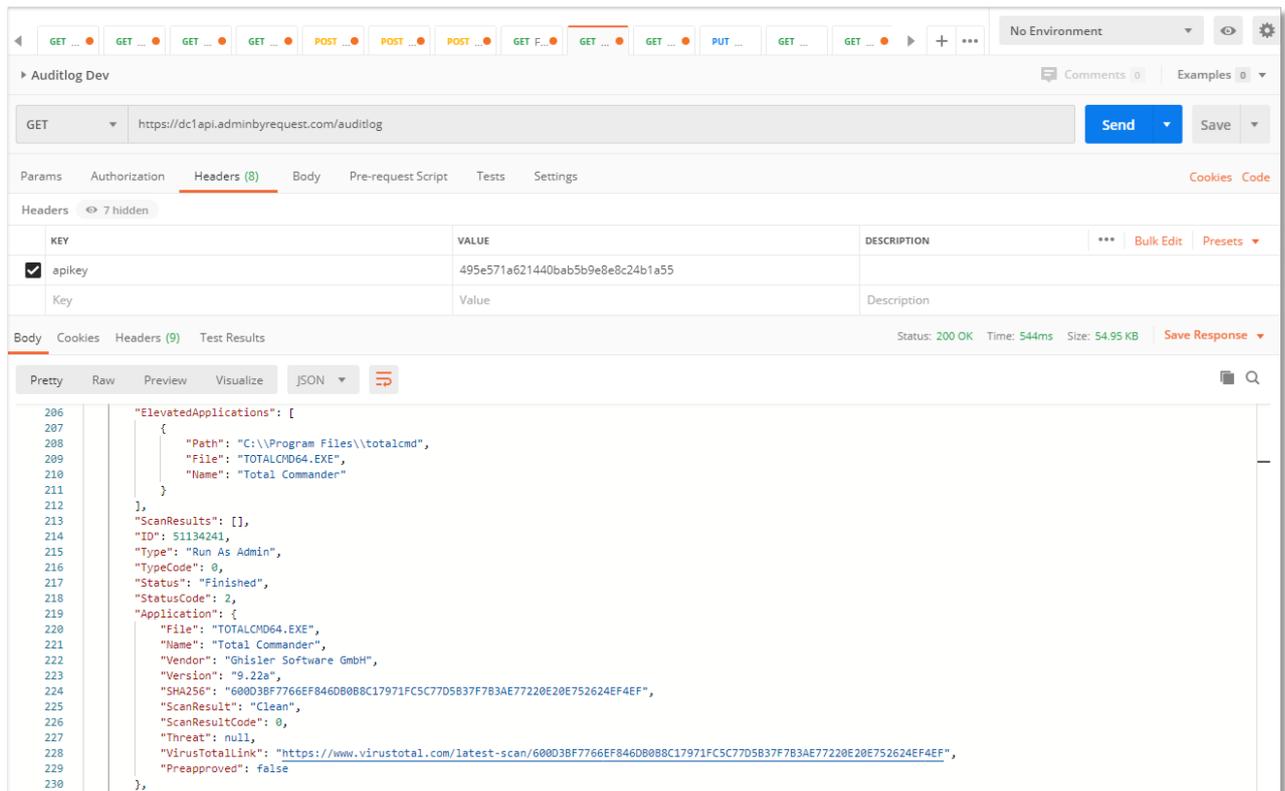
The screenshot shows the Admin By Request interface. At the top, there's a navigation bar with 'Settings' highlighted. Below it, a 'Tenant Settings' section is visible, with 'API KEYS' selected in the left sidebar. The main content area displays three API keys, each with a status of 'Active' and a 'Last use' date. A red 'Add New' button is located at the bottom of the API Keys section. Below the API Keys section, there's an 'About API Keys' section with a brief description and a URL: <https://dctapi.adminbyrequest.com>.

Once a key is generated, you can use Click to show, followed by Copy to clipboard in the api key window to copy and paste it.

Authentication

You can authenticate in two ways when consuming the API:

1. Send the api key as a header named "apikey":



2. Use standard basic authentication and send api key as the password (username is ignored).

Please refer to [Using PowerShell to Query ABR](#) for further screenshots.

Quota

These quota rules are enforced to avoid flooding the API. If quotas are passed, access will automatically be blocked for your tenant until next business day.

Daily quota: **100,000** API calls

Data

Data are returned in JSON format and standard http status codes apply (e.g. 200 = OK, 404 = Not Found, 500 = Internal Server Error).

The goal is to keep the structures as similar as possible to how they appear in the portal. If you need to do initial investigation or data mining, you can easily extract the data using Postman, Insomnia or other API tool of your choice. All you have to do is add the basic authentication password or add the apikey under "headers" and call a url, as shown above.

For more information on the available APIs from Admin By Request:

- [Auditlog API](#)
- [Requests API](#)

- [Inventory API](#)
- [Events API](#)
- [PIN Code API](#)

Emptying requests from an external system

This section demonstrates how requests can be retrieved and approved from an external system. The goal is to empty the request queue at an interval and approve or deny requests that are already stored in the external system. The scenario could be to create a recurring job in the external system (e.g. ServiceNow) and then add buttons on tickets to approve or deny requests.

NOTE

Integration connectors are added periodically between Admin By Request and other systems. At the time of writing, we have several integrations, including Jira, Teams and ServiceNow. If you use any of the integrations for which there is a predefined application, you do not need to use the API. Refer to [Integrations](#) for more information.

API tasks

The following tasks are covered in this section:

- Generate API Key
- Get requests
- Approve a request
- Track request state

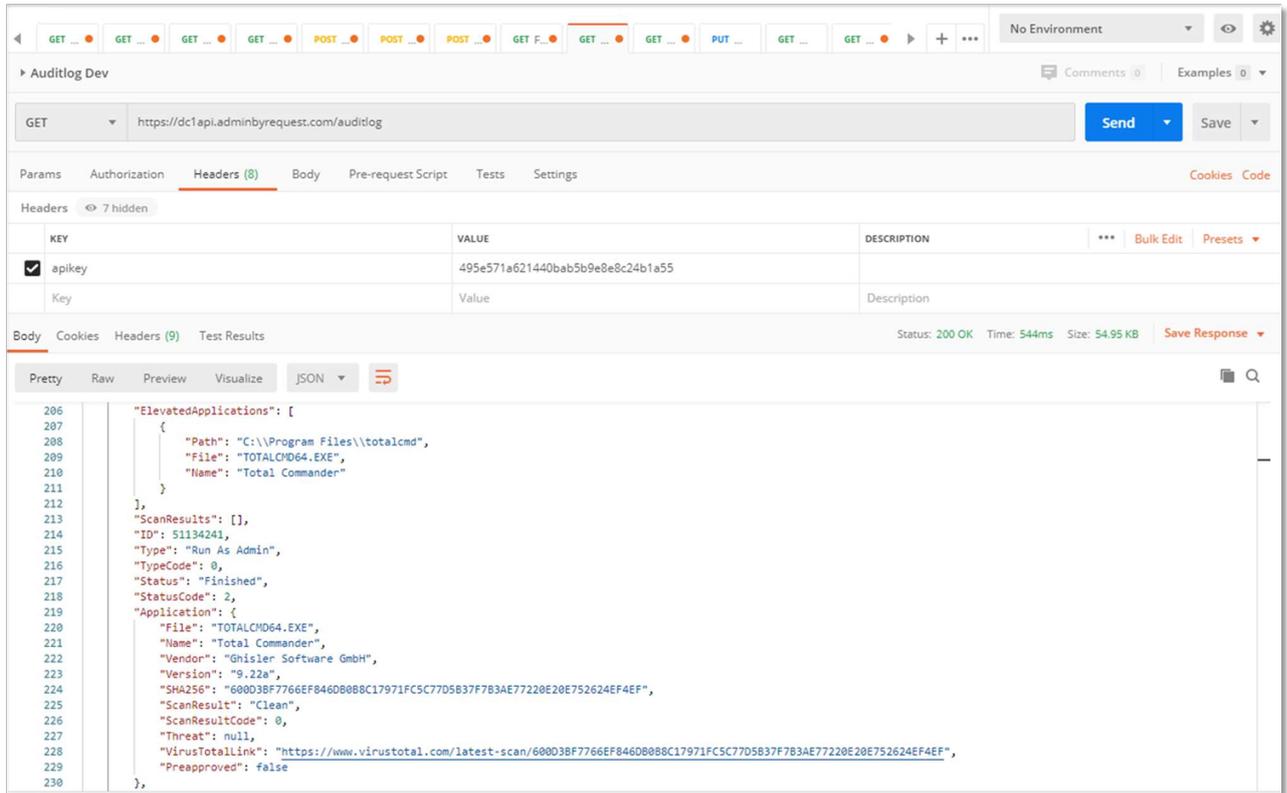
Generate API Key

To interact with the API, you must first generate an API key. The API key acts as a unique identifier that authenticates your requests, ensuring that only authorized users can access the API.

The API Key has a number of security features:

- **Default Security Settings** - By default, your tenant's data is not publicly accessible. This means that without an API key, external systems or users cannot retrieve data.
- **Queries and Updates** - Once generated, the API key must be included in API requests (usually in headers or query parameters). This allows the API to verify your identity and access permissions.
- **Brute Force Protection** - The API has security mechanisms in place to prevent excessive or unauthorized requests. Brute force protection helps prevent attackers from trying multiple API keys or requests rapidly. This protects your data from abuse and ensures the API remains available for legitimate users.
- **Safe to Open** - Since the API has built-in security measures (such as rate limiting and authentication), it is considered safe to expose it to external use. You should still follow best practices like:
 - Keeping your API key confidential.
 - Using secure storage for credentials.
 - Enforcing least privilege access (restricting API access to only necessary data).

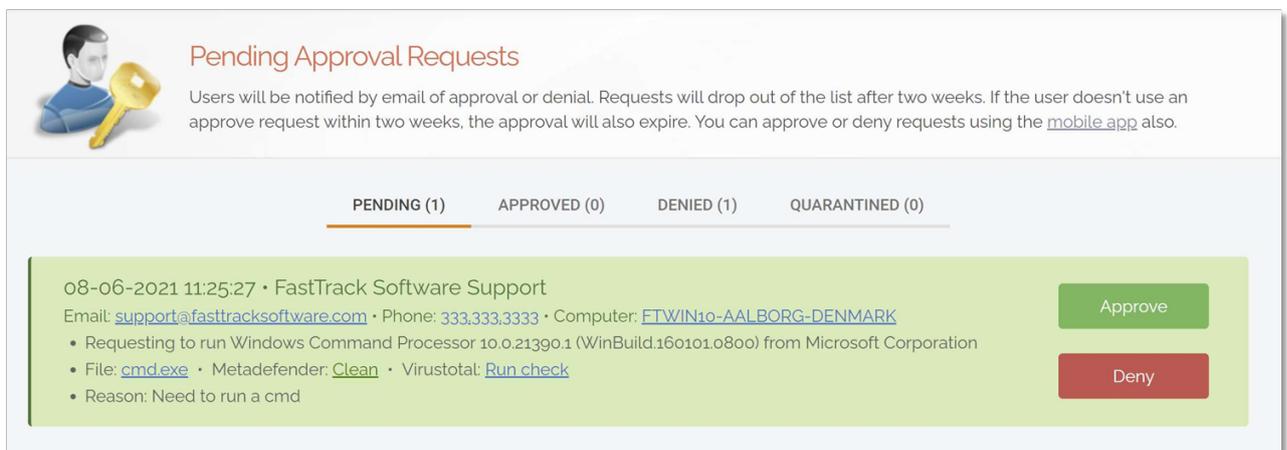
1. Generate an API Key by logging into your portal account and navigating to **Settings > Tenant Settings > API Keys > API KEYS**, as described above ("API" on page 2).
2. Once we have this key, we can make a call to the API using the API key as the "apikey" header. This example uses the free **Postman tool** to demonstrate. Here we have called the **Auditlog API** to get entries:



The next task is to use the API to get outstanding "Pending" requests.

Get requests

1. Requests will appear in the "Requests" top menu in the portal as shown here:



2. To get a list of pending requests for approval, we need to call the [Requests API](#), indicating that we are only interested in "Pending" states. For this, you will need to know the data center to which you are connected.

To determine your data center, go to page [Tenant Settings > API Keys](#) in the portal and check which API prefix is shown under **About API Keys**. The data center (which is also the API prefix) will be one of the following:

- <https://dc1api.adminbyrequest.com> (Europe)
- <https://dc2api.adminbyrequest.com> (USA)

Make a note of your prefix - among other things, this is the domain used when an API Key is created.

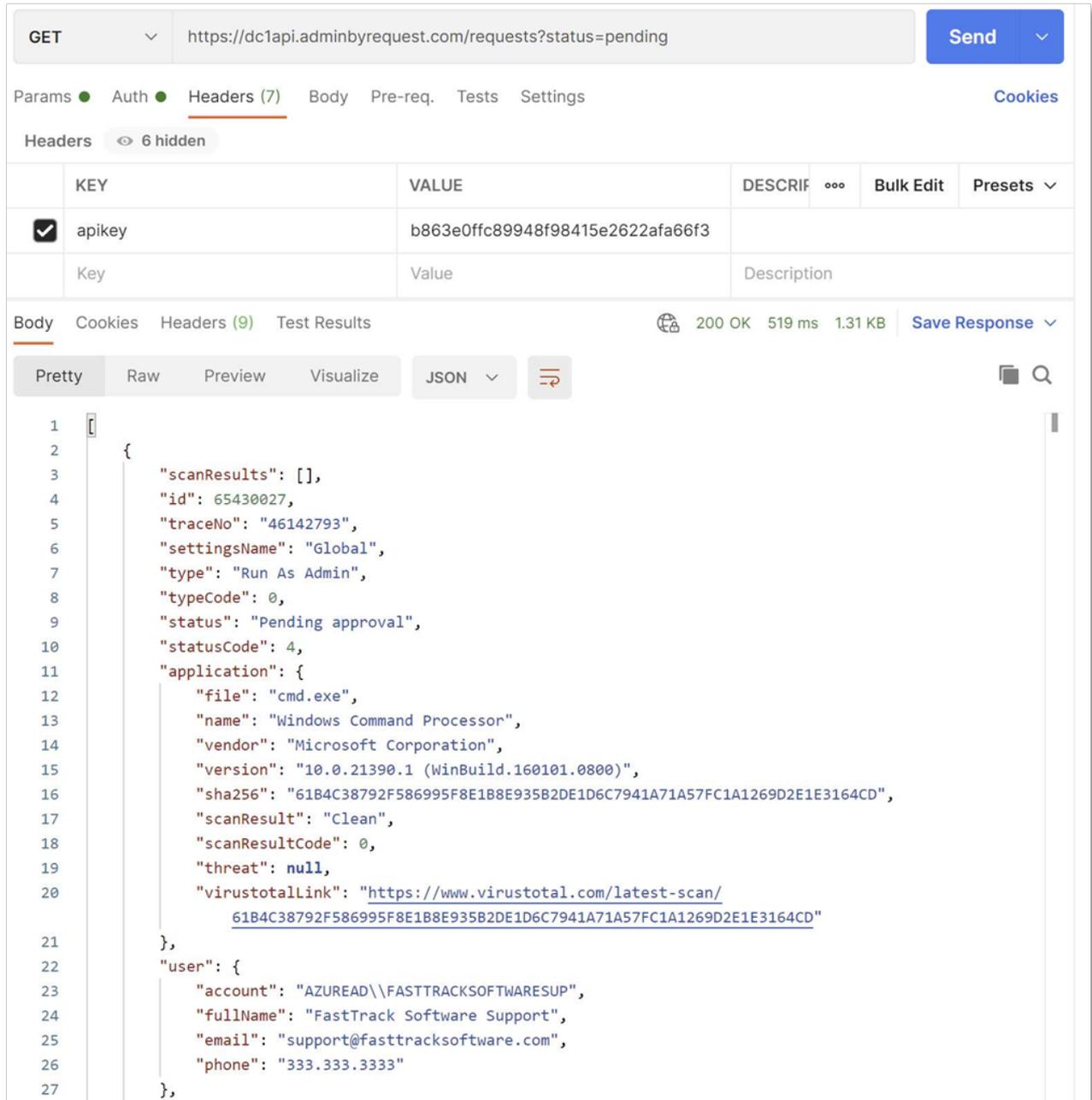
You can also see your API prefix on the API web pages (e.g. [Public API > Auditlog API](#)). However, a small script runs in the background that determines to which data center you are attached, so JavaScript must be enabled in your browser for this to work.

Continued next page ...

The URL for our example is then:

`https://dc1api.adminbyrequest.com/requests?status=pending`

Submitting this via **Send** returns a list in json format with the same information that is shown in the portal:



The screenshot shows a REST client interface with the following details:

- Method: GET
- URL: `https://dc1api.adminbyrequest.com/requests?status=pending`
- Headers (7): One header is visible, `apikey: b863e0ffc89948f98415e2622afa66f3`.
- Body: The response is a JSON array with one object. The JSON is displayed in the 'Pretty' view.

```
1 [
2   {
3     "scanResults": [],
4     "id": 65430027,
5     "traceNo": "46142793",
6     "settingsName": "Global",
7     "type": "Run As Admin",
8     "typeCode": 0,
9     "status": "Pending approval",
10    "statusCode": 4,
11    "application": {
12      "file": "cmd.exe",
13      "name": "Windows Command Processor",
14      "vendor": "Microsoft Corporation",
15      "version": "10.0.21390.1 (WinBuild.160101.0800)",
16      "sha256": "61B4C38792F586995F8E1B8E935B2DE1D6C7941A71A57FC1A1269D2E1E3164CD",
17      "scanResult": "Clean",
18      "scanResultCode": 0,
19      "threat": null,
20      "virustotalLink": "https://www.virustotal.com/latest-scan/61B4C38792F586995F8E1B8E935B2DE1D6C7941A71A57FC1A1269D2E1E3164CD"
21    },
22    "user": {
23      "account": "AZUREAD\\FASTTRACKSOFTWARESUP",
24      "fullName": "FastTrack Software Support",
25      "email": "support@fasttracksoftware.com",
26      "phone": "333.333.3333"
27    }
28  },
29 ]
```

The next step is to approve a request.

Approve a request

1. Notice that each request has an "id" field. This field is used to approve or deny a request as explained in [Requests API](#):

Resources			
https://dc1api.adminbyrequest.com/requests		Returns an array of requests	GET
https://dc1api.adminbyrequest.com/requests/{id}		Returns one request	GET
https://dc1api.adminbyrequest.com/requests/{id}		Approve request. If request is already approved or started, nothing happens	PUT
https://dc1api.adminbyrequest.com/requests/{id}		Deny request. If request is already approved or started, nothing happens	DELETE

2. So all we have to do is to make a simple **PUT** request with the ID:

PUT

https://dc1api.adminbyrequest.com/requests/65430027

Send

3. ... and the request is now approved and the user is notified accordingly. It's that simple!

Pending Approval Requests

Users will be notified by email of approval or denial. Requests will drop out of the list after two weeks. If the user doesn't use an approve request within two weeks, the approval will also expire. You can approve or deny requests using the [mobile app](#) also.

PENDING (0)
APPROVED (1)
DENIED (1)
QUARANTINED (0)

08-06-2021 11:25:27 • FastTrack Software Support

Email: support@fasttracksoftware.com • Phone: 333,333,3333 • Computer: [FTWIN10-AALBORG-DENMARK](#)

- Requesting to run Windows Command Processor 10.0.21390.1 (WinBuild.160101.0800) from Microsoft Corporation
- File: [cmd.exe](#) • Metadefender: [Clean](#) • Virustotal: [Run check](#)
- Reason: Need to run a cmd

Deny

Track request state

1. So how do you only get new requests? The easiest way is to store the highest id you have received before and use this for the next calls. Here are the filters for the [Requests API](#):

Filters (query string parameters)		
status	Only get request of a certain type (possible values: Pending, Denied, Approved, Quarantined)	int
startid	The starting ID you wish to receive. Can be used for incremental offload of data to your own system	int
take	Maximum number of resources to return. Default is 50 to preserve bandwidth, maximum is 1000. For queries with more than 1000 records, pagination is mandatory	int
last	Entries are retrieve in ascending order. Last returns the latest X number of entries in descending order. Maximum is 1000. Take and startid cannot be combined with last.	int
wantscandetails	Use this filter, if you wish to receive detailed lists of scan results. The default is to give you the overall result only	bit
reason	Can be passed when denying a request	string
deniedby	Can be passed when denying a request - this has to be an email address that matches a portal user, otherwise it will be ignored	string
approvedby	Can be passed when approving a request - this has to be an email address that matches a portal user, otherwise it will be ignored	string

- Example url to get one pending request: <https://dc1api.adminbyrequest.com/requests?status=pending&take=1&wantscandetails=1>
- Pagination works by using the last id in the list and feeding it as startid in the next query
- To copy new data to your own system, we recommend to store the highest id (last entry in list) you have retrieved from a previous call and pass this number plus 1 as "startid"

- Let's say the last request we received was 65430930. Then we add one (since we already received 65430930) and use as startid, meaning that we only want to see any requests with an id higher than 65430930:

```
GET https://dc1api.adminbyrequest.com/requests?startid=65430931 Send
```

Questions?

Feel free to ask any questions you might have. If you do not have a direct contact at Admin By Request, [contact us](#) with any questions you might have, or submit a [support ticket](#).

Document History

Version	Author	Changes
1.0 28 November 2022	Sophie Alice Dodson	Initial document release (as "API Request Approval").
2.0 4 April 2025	Steve Dodson	Updated structure and layout. Updated screenshots and procedures to improve clarity and reflect changes to the Admin By Request portal.