FORCEPOINT

Forcepoint DLP REST API Guide

Mar 2022

Published 2022

Document last updated March 31, 2022

Contents

1 Introduction

Forcepoint DLP REST API Guide | Mar 2022

This guide provides information about the Representational State Transfer (REST) Application Programming Interface (APIs) available for Forcepoint DLP.

The Forcepoint DLP REST APIs provide a set of rules that lets your application communicate with Forcepoint DLP using the HTTP protocol over predefined URLs. These URLs represent Forcepoint DLP content that can be returned as JavaScript Object Notation (JSON) files.

Starting in Forcepoint DLP v8.9, you can use these REST APIs to get a list of DLP and Discovery incidents from Forcepoint DLP or update and remediate DLP and Discovery incidents.

In this guide:

2 Authentication

Forcepoint DLP REST API Guide | Mar 2022

Before you can connect to the Forcepoint DLP REST APIs, you need to create a new Application administrator in the Forcepoint Security Manager to create the username and password for authentication to be used in order to get a JSON Web Token (JWT) that allows you to send API requests.

In this chapter:

Registering an Application in the Forcepoint Security Manager

Forcepoint DLP REST API Guide | Mar 2022

To connect an application to Forcepoint DLP through a REST API connection, you need to create an Application administrator in the Forcepoint Security Manager.

1. On the **Global Settings > General > Administrators** settings page, select **Add Local Account**.

2. On the **Add Local Account** page, add the information for the administrator account, then select the **Application** option for the **Administrator type**.

3. Click **OK** to save the new account.

For more information, see the Enabling access to the Security Manager topic in the Forcepoint Security Manager Help.

Note
The Application administrator type is only supported for Local accounts. Network accounts cannot be configured as an Application type.

Authentication process

Forcepoint DLP REST API Guide | Mar 2022

The login and authentication workflow accesses the **refresh token** and **access token** APIs to create and send a JWT.

1. The client application accesses the refresh token API with Forcepoint DLP `username` and `password` header parameters.

Important
Only the Application administrator type can request a refresh token through the REST API. If a User administrator type requests the refresh token, the system returns a 403 error code.

2. The refresh token is issued as the JWT with the expiration time of 1 day. A refresh token is used to retrieve the access token and to prevent the continual passing of the `username` and `password` header parameters over the network. The access token is provided on this call as a return parameter to save a first get access token call.

Important
Communication is through HTTPS. The username and password are not encrypted.

Note
If the account password is changed, the refresh token is still valid until the expiration time.

3. After the first access token expires, the client application accesses the get access token API with a received refresh token to receive a new access token. Access tokens expire after 15 minutes.

Note
The refresh token cannot be used for the client application API call. The system returns a 403 error code if the refresh token is used to get or update the Incidents API.

Refresh Token API

The **refresh token** API retrieves the refresh token (JWT) that is used for the application REST API authentication.

POST https://<DLP Manager IP>:<DLP Manager port>/dlp/rest/v1/auth/refresh-token

In this section:

Header parameters for the refresh token API

The following two header parameters are required: username and password.

Parameter	username
Valid Values	The administrator account user name in Forcepoint DLP.
Example	`username: <your Forcepoint DLP Admin application username>`

Parameter	password
Valid Values	The administrator account password in Forcepoint DLP. The password is passed through HTTPS, so it does not need to be encrypted or encoded.
Example	`password: <your Forcepoint DLP Admin application password>`

Request example for the refresh token API

A full request is not sent for this API. Only the header parameters are sent in the request.

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/auth/refresh-token' \
--header 'username: <username> '\
--header 'password: <password> '
```

Response for the refresh token API example

```
{
    "refresh_token": "<refresh token>",
    "refresh_expires_in": <expiration value>
    "access_token": "<access token>",
    "access_expires_in": <expiration_value>,
    "token_type": "JWT"
}
```

Status and error codes for the refresh token API

Code	Message
200	Success
403	Forbidden (authentication failed)

Access Token API

Forcepoint DLP REST API Guide | Mar 2022

The **access token** API retrieves the access token that is used for the application REST API authentication.

POST https://<DLP Manager IP>:<DLP Manager port>/dlp/rest/v1/auth/access-token

In this section:

Header parameters for the access token API

The following header parameter is required: refresh-token.

Parameter	refresh-token
Valid Values	The JWT received on /v1/auth/refresh-token call.
Example	`refresh-token: Bearer <refresh token>`

Request example for the access token API

A full request is not sent for this API. Only the header parameter is sent in the request.

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/auth/access-token' \
--header 'refresh-token: Bearer <refresh token> '
```

Response example for the access token API

```
{
    "access_token": "<JWT access token>",
    "access_expires_in": <expiration_value>,
    "token_type": "JWT"
}
```

Status and error codes for the access token API

Code	Message
200	Success
403	Forbidden (authentication failed)

3 REST APIs

Forcepoint DLP REST API Guide | Mar 2022

This chapter provides detailed information about the specific REST APIs available in Forcepoint DLP:

- *Get Incidents API*, page 9 - retrieves a list of the incidents by provided filters.
- *Update Incidents API*, page 20 - updates the incidents retrieved during the **get incidents** API call.

Troubleshooting

If you run into issues with the REST APIs, check the following:

- Application errors are available by default and are printed in Websense/Data Security/tomcat/logs/dlp/dlp-all.log
- Debug logs are available under Websense/Data Security/tomcat/lib/log4j-dlp.properties in the #REST API section. These entries are printed into dlp-all.log after being opened.

Get Incidents API

Forcepoint DLP REST API Guide | Mar 2022

The **Get Incidents** API retrieves a list of the incidents by provided filters. This API receives either an ID list or filters request (not both).

POST https://<DLP Manager IP>:<DLP Manager port>/dlp/rest/v1/incidents

This API returns a maximum of 10,000 incidents per response. The list of returned incidents can contain one type of the two options: DLP incidents (INCIDENTS) or Discovery incidents (DISCOVERY).

In this section:

- *Header parameters for the Get Incidents API*, page 10
- *Input parameters for the Get Incidents API*, page 10

Header parameters for the Get Incidents API

The following two header parameters are required: Authorization and Content-Type.

Parameter Authorization
Valid Values The JWT used to authenticate requests.
Example `Authorization: Bearer <access token>`

Parameter Content-Type
Valid Values application/json
Example `Content-Type: application/json`

Input parameters for the Get Incidents API

Root parameters

The following parameters are shown in the request. For examples of requests, see *Request examples for the Get Incidents API*, page 12.

Name	Required/ Optional	Supported	Valid values
type	Required	INCIDENTS, DISCOVERY	INCIDENTS DISCOVERY
ids	Required (for by IDs filter)	INCIDENTS, DISCOVERY	Comma separated array of incident IDs. Example: `[123, 345]` The number of provided IDs is limited to 1,000. Error code 400 is returned if this limit is violated. Note: If this parameter is provided, it is used where any provided filters are ignored.
sort_by	Optional	INCIDENTS, DISCOVERY	INSERT_DATE

Name	Required/ Optional	Supported	Valid values
from_date	Required (for not by IDs filter)	INCIDENTS, DISCOVERY	Date in format “dd/MM/yyyy HH:mm:ss” Example: `12/08/2021 16:00:00`
to_date	Required (for not by IDs filter)	INCIDENTS, DISCOVERY	Date in format “dd/MM/yyyy HH:mm:ss” Example: `13/08/2021 18:55:00`
detected_by	Optional	INCIDENTS, DISCOVERY	Agent detected the violation. Example: `Endpoint Agent`, `Crawler 100190120a`
analyzed_by	Optional	INCIDENTS, DISCOVERY	Policy Engine ID. Example: `Policy Engine 100190120a`
event_id	Optional	INCIDENTS, DISCOVERY	Event ID number. Example: `5121411628328991975`
destination	Optional	INCIDENTS	Destination. Example: `Windows Portable Device (WPD)`
policies	Optional	INCIDENTS	Policy that triggered the incident. Example: `PCI`
action	Optional	INCIDENTS	AUDITED QUARANTINED BLOCKED ENCRYPTED RELEASED ESG_ACTION QUARANTINE_WITH_NOTE UNSHARE_EXTERNAL UNSHARE_ALL UNSHARE_INTERNAL
source	Optional	INCIDENTS	Source. Example: `DESKTOP-3NG4NN6\\Lenovo`
status	Optional	INCIDENTS, DISCOVERY	NEW IN_PROCESS CLOSE FALSE_POSITIVE ESCALATED Note: Also supports a custom status.
severity	Optional	INCIDENTS, DISCOVERY	HIGH MEDIUM LOW

Name	Required/ Optional	Supported	Valid values
endpoint_type	Optional	INCIDENTS	LAPTOP DESKTOP NA
channel	Optional	INCIDENTS	EMAIL ENDPOINT_EMAIL FTP HTTP HTTPS ENDPOINT_HTTP ENDPOINT_HTTPS ENDPOINT_PRINTING ENDPOINT_APPLICATION ENDPOINT_REMOVABLE_MEDIA ENDPOINT_LAN ENDPOINT_DISCOVERY CASB_REAL_TIME CASB_NEAR_REAL_TIME CASB_DISCOVERY
assigned_to	Optional	INCIDENTS, DISCOVERY	The administrator name assigned to a ticket Example: `admin`
tag	Optional	INCIDENTS, DISCOVERY	The Incident tag. Example: `my tag`
remove_ignored _incidents	Optional (default is false)	INCIDENTS, DISCOVERY	Filter out ignored incidents from the results. TRUE FALSE

Request examples for the Get Incidents API

This section shows examples of requests to the API. For more information about each parameter shown in the request, see *Input parameters for the Get Incidents API*, page 10.

Request to get incidents by IDs (list one incident)

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/' \
--header 'Authorization: Bearer <access token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "ids" : [262458],
    "type" : "INCIDENTS"
```

```
}'
```

Request to get incidents by date range

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/' \
--header 'Authorization: Bearer <access token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "type" : "INCIDENTS",
    "from_date" : "31/10/2021 09:56:00",
    "to_date" : "08/11/2021 09:57:00"
}'
```

Request to get incidents by action

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/' \
--header 'Authorization: Bearer <access token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "type" : "INCIDENTS",
    "from_date" : "01/08/2021 16:00:00",
    "to_date" : "12/08/2021 20:00:00",
    "action" : "BLOCKED"
}'
```

Request to get incidents by severity

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/' \
--header 'Authorization: Bearer <access token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "type" : "INCIDENTS",
    "from_date" : "01/08/2021 16:00:00",
    "to_date" : "12/08/2021 20:00:00",
    "severity" : "MEDIUM"
}'
```

Request to get incidents by status

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/' \
--header 'Authorization: Bearer <access token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "type" : "INCIDENTS",
    "from_date" : "01/08/2021 16:00:00",
    "to_date" : "12/08/2021 20:00:00",
    "status" : "NEW"
}'
```

Request to get incidents by policy name

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/' \
--header 'Authorization: Bearer <access token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "type" : "INCIDENTS",
    "from_date" : "01/08/2021 16:00:00",
    "to_date" : "12/08/2021 20:00:00",
    "policies" : "PCI"
}'
```

Request with date filters and sorting

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/' \
--header 'Authorization: Bearer <access token> ' \
--header 'Content-Type: application/json' \
--data-raw '{
    "sort_by": "INSERT_DATE",
    "type" : "INCIDENTS",
    "from_date" : "01/08/2021 16:00:00",
    "to_date" : "12/08/2021 20:00:00",
}'
```

Request with a filter

Default hidden filter that excludes false positive incidents (same as in UI):

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/' \
--header 'Authorization: Bearer <access token> ' \
--header 'Content-Type: application/json' \
--data-raw '{
    "sort_by": "INSERT_DATE",
    "type" : "INCIDENTS",
    "from_date" : "01/08/2021 16:00:00",
    "to_date" : "12/08/2021 20:00:00",
    "detected_by" : "Endpoint Agent",
    "analyzed_by": "Policy Engine  100190120a",
    "event_id" : 5121411628328991975,
    "destination" : "Windows Portable Device (WPD)",
    "policies" : "PCI",
    "action" : "BLOCKED",
    "source" : "DESKTOP-3NG4NN6\\Lenovo",
    "status" : "NEW",
    "severity" : "MEDIUM",
    "endpoint_type" : "LAPTOP",
    "channel" : "ENDPOINT_REMOVABLE_MEDIA",
    "assigned_to" : "admin",
    "tag" : "Vadim tag"
}'
```

Response output for the Get Incidents API

Note

The output parameters listed in this section cannot be used as input parameters. To send a request using specific filters (parameters), use the input parameters above (*Input parameters for the Get Incidents API*, page 10).

Root output

The following parameters are shown in the response. For examples of responses, see *Response example for the Get Incidents API*, page 18.

Name	Supported	Comments
total_count	INCIDENTS DISCOVERY	Total number of incidents that found by the provided filters.
total_returned	INCIDENTS DISCOVERY	Number of incidents returned on response.
not_found_ids	INCIDENTS DISCOVERY	List of IDs that were provided on API input but were not found in the database. This list is returned along with the custom error code 420.
incidents	INCIDENTS DISCOVERY	Array of incident objects.

Object properties for incidents

The following objects are included in the response for the `incidents` parameter.

Name	Supported	Comments
id	INCIDENTS DISCOVERY	
severity	INCIDENTS DISCOVERY	
action	INCIDENTS DISCOVERY	
tag	INCIDENTS	
status	INCIDENTS	
destination	INCIDENTS	
details	INCIDENTS	
released_incident	INCIDENTS	
event_id	INCIDENTS DISCOVERY	
maximum_matches	INCIDENTS DISCOVERY	
transaction_size	INCIDENTS DISCOVERY (by ID only)	

Name	Supported	Comments
assigned_to	INCIDENTS DISCOVERY (by ID only)	
analyzed_by	INCIDENTS DISCOVERY (by ID only)	
ignored_incidents	INCIDENTS	
event_time	INCIDENTS	
incident_time	INCIDENTS DISCOVERY	
channel	INCIDENTS DISCOVERY (by filter only)	
policies	INCIDENTS DISCOVERY	
partition_index	INCIDENTS	
detected_by	INCIDENTS	
endpoint_type	INCIDENTS	
violation_triggers	INCIDENTS DISCOVERY (by ID only)	Number of violation triggers.
file_name	INCIDENTS	
file_path	DISCOVERY	

Object properties for source

The following objects are included in the response for the `source` parameter.

Name	Supported	Comments
manager	INCIDENTS	
department	INCIDENTS	
ip_address	INCIDENTS	
login_name	INCIDENTS	
host_name	INCIDENTS DISCOVERY (by ID only)	
email_address	INCIDENTS	
dn	INCIDENTS	
nt_domain	INCIDENTS	

Name	Supported	Comments
risk_level	INCIDENTS	Set only if a value > 0
business_unit	INCIDENTS	

Response example for the Get Incidents API

This section shows an example of a response from this API. For more information about each parameter shown in the response, see *Response output for the Get Incidents API*, page 15.


```
{
    "incidents": [
        {
            "id": 373623,
            "severity": "HIGH",
            "action": "RELEASED",
            "tag": "Tag",
            "status": "Closed",
            "source": {
                "email_address": test2@aaa.com
            },
            "event_id": "7728775614896485765",
            "maximum_matches": 13,
            "transaction_size": 2632,
            "analyzed_by": "Policy Engine  1272021",
            "ignored_incidents": false,
            "event_time": "19/10/2021 10:12:02",
            "incident_time": "19/10/2021 10:12:02",
            "channel": "EMAIL",
            "policies": "Credit Cards; PCI",
            "partition_index": 20211019,
            "destination": aaa@aaa.net,
            "detected_by": "Protector on 1272021",
            "details": "Automatic Email Subject with
<keyword>",
            "released_incident": true,
            "violation_triggers": 2,
            "file_name": "visa.txt - 1.09 KB"
        }
    ],
    "total_count": 1,
```

```
    "total_returned": 1
}
```

If no incidents were found, the response returns a 420 error code along with a response similar to the following:

```
{
    "incidents": [],
    "total_count": 0,
    "total_returned": 0,
    "not_found_ids": [
        373623111
    ]
}
```

Status and error codes for Get Incidents API

Code	Message
200	Success
400	Bad request (no valid input)
403	Forbidden (authentication failed)
420	No incidents were found

Update Incidents API

Forcepoint DLP REST API Guide | Mar 2022

The **update** API updates the incidents retrieved during the **incidents** API call.

POST https://<DLP Manager IP>:<DLP Manager port>/dlp/rest/v1/incidents/update

In this section

Header parameters for the Update Incidents API

The following two header parameters are required: Authorization and Content-Type.

Parameter	Authorization
Valid Values	The JWT used to authenticate requests.
Example	`Authorization: Bearer <access token>`

Parameter	Content-Type
Valid Values	application/json
Example	`Content-Type: application/json`

Input parameters for the Update Incidents API

Root parameters

The following parameters are shown in the request. For examples of requests, see *Request examples for the Update Incidents API*, page 23.

Name	Required/ Optional	Supported	Valid values
type	Required	INCIDENTS, DISCOVERY	INCIDENTS DISCOVERY
action_type	Required	INCIDENTS, DISCOVERY	STATUS SEVERITY ASSIGN_TO ADD_COMMENT TAG RELEASE (not supported for DISCOVERY) FALSE_POSITIVE
value	Required Optional (ADD_COMMENT, RELEASE)	INCIDENTS, DISCOVERY	STATUS: NEW, IN_PROCESS, CLOSE, FALSE_POSITIVE, ESCALATED, custom status SEVERITY: HIGH, MEDIUM, LOW ASSIGN_TO: admin name to be assigned to TAG: tag name (maximum 100 chars) FALSE_POSITIVE: 1 (ignore), 0 (include)
comment	Optional Required for ADD_COMMENT	INCIDENTS, DISCOVERY	Supported for the following actions: ADD_COMMENT ASSIGN_TO TAG RELEASE (not supported for DISCOVERY) FALSE_POSITIVE
scan_partitions	Optional (default value is NONE)	INCIDENTS (relevant only if incident_keys is populated)	Parameter to identify if partition_index was provided on each event key. ALL: Scans all partitions to get incidents and fetches their partition_index that is required for an update. NONE: Assumes partition_index is provided. If partition_index is missing on any key, then an exception is thrown. LAST_ACTIVE: Sets last 2 partitions and sends update with them. If the incident is not located on those 2 partitions, then update does not execute.

Name	Required/ Optional	Supported	Valid values
event_ids	Required when update by event ids	INCIDENTS, DISCOVERY	Array of the Event IDs to be updated. If event_ids is provided, then it is required to perform a lookup for incidents by event ID to get incident_id and partition_index. Currently, there is no API to avoid searching over all partitions. The number of provided IDs is limited to 1,000. Error code 400 is returned if violated.
incident_keys	Required when update by incident keys	INCIDENTS, DISCOVERY	Array of Incident Key objects on which the action should be performed. The number of provided IDs is limited to 1,000. Error code 400 is returned if violated.

Object properties for incident_keys

The following objects are included in the request for the `incident_keys` parameter.

Name	Required/ Optional	Supported	Valid values
incident_id	Required	INCIDENTS, DISCOVERY	Incident ID to be updated.
partition_index	Optional	INCIDENTS	Partition Index from the incidents table that are required to build a query. If this field is not provided on an INCIDENTS request, then the system should look up this parameter internally. In this case, the API will require more resources to complete the update action.

Request examples for the Update Incidents API

This section shows examples of requests to the API. For more information about each parameter shown in the request, see *Input parameters for the Update Incidents API*, page 21.

Request to update incident status by Incident ID and Partition Index

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/update' \
--header 'Authorization: Bearer <access token> ' \
--header 'Content-Type: application/json' \
--data-raw '{
    "incident_keys" : [
        {
            "incident_id" : 2719662,
            "partition_index": 20210831
        },
        {
            "incident_id" : 2719665,
            "partition_index": 20210831
        },
        {
            "incident_id" : 271966800000,
            "partition_index": 20210831
        }
        ],
    "type" : "INCIDENTS",
    "action_type" : "STATUS",
    "value" : "NEW"
}'
```

Request to update incident status with scan_partitions set to ALL

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/update' \
--header 'Authorization: Bearer <access token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "incident_keys" : [
        {
            "incident_id" : 132035
        }
```

```
        ],
    "type" : "INCIDENTS",
    "action_type" : "STATUS",
    "value" : "IN_PROCESS",
    "scan_partitions" : "ALL"
}'
```

Request to update incidents status with scan_partitions set to LAST_ACTIVE

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/update' \
--header 'Authorization: Bearer <access token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "incident_keys" : [
        {
            "incident_id" : 2185301
        },
        {
            "incident_id" : 2719665
        },
        {
            "incident_id" : 2719668
        }
        ],
    "type" : "INCIDENTS",
    "action_type" : "STATUS",
    "value" : "NEW",
    "scan_partitions" : "LAST_ACTIVE"
}'
```

Request to update incidents status by event IDs

```
curl --location --request POST 'https://<DLP Manager
IP>:<DLP Manager port>/dlp/rest/v1/incidents/update' \
--header 'Authorization: Bearer <access token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "event_ids" : [9315711207487646059,
5758754422662242777],
    "type" : "INCIDENTS",
```

```
    "action_type" : "TAG",
    "value" : "custom tag"
}'
```

Response output for the Update Incidents API

> **Note**
> The output parameter listed in this section cannot be used as an input parameter. To send a request using specific filters (parameters), use the input parameters above (*Input parameters for the Update Incidents API*, page 21).

The following parameter is shown in the response. For examples of responses, see *Response output for the Update Incidents API*, page 25.

Name	Supported	Comments
unprocessed_ids	INCIDENTS, DISCOVERY	List of incident_ids or event_ids that were provided on API input but were not updated. API call returns error code 422 when this array is not returned.

Response example for the Update Incidents API

A successful response (code 200) contains an empty response body.

If no incidents are processed, the response returns a 422 error code along with a response similar to the following:

```
{
    "unprocessed_ids": [
        "3624501111",
        "2659051111"
    ]
}
```

Status and error codes for the Update Incidents API

Code	Message
200	Success
400	Bad request (no valid input)
403	Forbidden (authentication failed)
422	Resources not processed (incidents were not updated due to an error)