LOGPOINT

Integrations

Universal REST API for Director Console API

*V*3.3.0

CONTENTS

CHAPTER

ONE

UNIVERSAL REST API FETCHER FOR DIRECTOR CONSOLE API

Universal REST API Fetcher provides a generic interface to fetch logs from cloud sources via REST APIs. The cloud sources can have multiple endpoints, and every configured source consumes one device license.

CHAPTER

TWO

INSTALLING UNIVERSAL REST API FETCHER

2.1 Uploading Universal REST API Fetcher to the Fabric Storage

2.1.1 Private Location

You can upload Universal REST API Fetcher to a private location in the Fabric Storage using the **Upload - Upload** API.

Endpoint URL:

https://api-server-host-name/configapi/Uploads/{pool_UUID}/

Method:

POST

Header:

FIELD	DESCRIPTION
file_name	Name of the Universal REST API Fetcher .pak file.
Content-Type	Mandatory value: *(application/octet-stream)*.

Parameters:

FIELD	TYPE	DESCRIPTION	REQUIRED
file	[Object]	The Universal REST API Fetcher .pak file to upload.	Mandatory

Success Response:

```
{
  "status": "Success",
  "message": "UniversalRESTAPIFetcher.pak successfully uploaded in private storage."
 }
```

2.1.2 Public Location

You can upload Universal REST API Fetcher to a public location in the Fabric Storage using the **Upload - UploadPublic files** API.

Endpoint URL:

```
https://api-server-host-name/configapi/Uploads/PublicUpload
```

Method:

```
POST
```

Header:

FIELD	DESCRIPTION
file_name	Name of the Universal REST API Fetcher .pak file.
Content-Type	Mandatory value: *(application/octet-stream)*.

Parameters:

FIELD	TYPE	DESCRIPTION	REQUIRED
file	[Object]	The Universal REST API Fetcher .pak file to upload.	Mandatory

Success Response:

```
{
  "status": "Success",
  "message": "UniversalRESTAPIFetcher.pak successfully uploaded in public storage. "
 }
```

2.2 Installing Universal REST API Fetcher in Logpoint

You can install Universal REST API Fetcher in a Fabric-enabled Logpoint using the **Upload - Install** API. It must be uploaded to the Fabric Storage before the installation.

Endpoint URL:

https://api-server-host-name/configapi/Uploads/{pool_UUID}/{logpoint_identifier}/install

Method:

POST

Parameters:

FIELD	LABEL IN UI	TYPE	DESCRIPTION	REQUIRED
application_type	–	String	Mandatory value: *Application*.	Mandatory
file_name	–	String	Name of the application file. Obtain it from the public location using the **Upload - ListPublic** API and the private location using the **Upload - List** API.	Mandatory
file_location	–	String	Location of the application file. Can be either *private* or *public*.	Mandatory

Request Example:

```
{
 "data": {
    "application_type": "Application",
    "file_name": "UniversalRESTAPIFetcher.pak",
    "file_location": "private"
 }
}
```

Success Response:

```
{
   "status": "Success",
   "message": "monitorapi/{pool_UUID}/{logpoint_identifier}/orders/{request_id}"
}
```

CHAPTER

THREE

UNINSTALLING UNIVERSAL REST API FETCHER

You can uninstall Universal REST API Fetcher in a Fabric-enabled Logpoint using the **Plugins - Uninstall** API. You must first *remove* the Universal REST API Fetcher configuration to uninstall it.

Endpoint URL:

https://api-server-host-name/configapi/{pool_UUID}/{logpoint_identifier}/Plugins/{id}

Method

Delete

Parameters:

FIELD	LABEL IN UI	TYPE	DESCRIPTION	REQUIRED
id	–	String	Universal REST API Fetcher application ID. Obtain it using the **Plugins - List** API.	Mandatory

Success Response:

```
{
  "status": "Success",
  "message": "monitorapi/{pool_UUID}/{logpoint_identifier}/orders/{request_id}"
}
```

CHAPTER

FOUR

CONFIGURING UNIVERSAL REST API FETCHER

You can configure the Universal REST API Fetcher in a Fabric-enabled Logpoint using the **LogSources - Create** API.

Endpoint URL:

https://api-server-host-name/configapi/{pool_UUID}/{logpoint_identifier}/LogSources

Method:

POST

Parameters:

Field	Label in UI	Type	Description
dc_metadata	-	json	Information related to log source template created in the Director Console. Optional Field.
description	Description	String	Additional information about the log source. Optional Field.
documentation_link	Documentation Link	String	URL or hyperlink that points to external documentation or reference materials associated to a specific log source. Optional Field.
logo	Logo	String	Base64 encoded logo image. Optional Field.
name	Name	String	Name of log source. Mandatory Field.
type	-	String	Type or category of the log source. Mandatory Field.
vendor_name	Vendor Name	String	Name of vendor where the log data originates. Optional Field.
config	-	json	Configuration of log source. *Source*, *connector*, *endpoints*, *routing*, *normalization* and *enrichment* must be configured for Universal REST API Fetcher to fetch logs. Mandatory Field.

4.1 Source

In source, you can add details about the *REST APIs*, from where the Universal REST API Fetcher fetches logs for accurate identification, data formatting, and timestamping.

Parameters:

Field	Label in UI	Type	Description
base_url	Base URL	String	Base URL of the RESTful API. Mandatory Field.
request_timeout	Request Timeout (secs)	String	API request timeout. Mandatory Field.
retry_after	Retry After(secs)	Integer	Time to wait after an error or timeout. Mandatory Field.
interval	Fetch Interval (min)	Integer	Fetch Interval in minutes. Mandatory Field.
charset	Charset	String	Existing Logpoint charset. Obtain it using the **Charsets - List** API. Mandatory Field.
timezone	Timezone	String	Timezone of the log source. Mandatory Field.

4.2 Connector

In connector, you can configure how the Universal REST API Fetcher and *REST APIs* communicate with each other.

Field	Label in UI	Type	Description
auth_type	Authentication Type	String	API authentication method. Mandatory Field.
key	Key	String	RESTful API custom headers. Mandatory Field.
value	Value	String	RESTful API custom headers. Mandatory Field.
enforce_https	Enforce HTTPS certificate verification	String	Parameter to enable a secure connection. Mandatory Field.
enable_proxy	Proxy Configuration	json	Proxy configuration of the log source server: **status**: Parameter to enable or disable the proxy server. **IP**: IP of the proxy server. **port**: Port of the proxy server. **protocol**: "HTTP" or "HTTPS" protocol used by the proxy server.

4.3 Endpoints

In endpoints, you can configure details about the REST API endpoints.

Parameters:

Field	Label in UI	Type	Description
endpoint_name	Name	String	Endpoints name. Mandatory Field.
method	Method	String	Request method to call the endpoint. Mandatory Field.
endpoint	Endpoint	String	Endpoint part of the previously added Base URL. Mandatory Field.
endpoints_custom_headers	Key and Value	String	Custom header's Key and its Value. Mandatory Field.
query_params	Query Parameters Key and Value	String	Request parameter's Key and Value. Mandatory Field.
incremental_value_response_field	Increment Value / Check Sum	String	Increment field from the response of the RESTful API. For example, if the increment field is event_date and it is inside Events, then enter Events.event_date. The field is saved in CheckSum, a database that uses the field to record until data is fetched. This ensures there is no log duplication as Universal REST API Fetcher checks the CheckSum every time before fetching any new data. Mandatory Field.
log_filter_params_dataformat	Data format	Date	Date format of the incoming logs. Mandatory Field.
log_filter_params_from_value	Initial Fetch	String	Logs are fetched for the first time from this date. Mandatory Field.
pagination_key	Pagination Key	String	Location of the following page URL from the response if the API supports pagination. Mandatory Field. For example, if the data

4.4 Routing

In routing, you can create repos and routing criteria for Universal REST API Fetcher. Repos are locations where incoming logs are stored and routing criteria are created to determine the conditions under which these logs are sent to repos.

Parameters:

Field	Label in UI	Type	Description
repo_name	Repo name	String	Name of the repo where incoming logs are stored. Mandatory Field.
path	Path	String	Location to store incoming logs. Mandatory Field.
retention	Retention (Days)	String	Number of days logs are kept in a repository before they are automatically deleted. Mandatory Field.
remote_logpoint	Remote logpoint	String	Remote Logpoint. Optional Field.
key and value	Key and Value	String	The key-value pair is used to apply routing criteria to logs. Optional Field.
operation	Operation	String	Operation for logs that have the key-value pair. Optional Field.
repository	Repository	String	Repo to store logs. Optional Field.

4.5 Normalization

In normalization, you can select normalizers for the incoming logs. Normalizers transform incoming logs into a standardized format for consistent and efficient analysis.

Parameters:

Field	Label in UI	Type	Description
normalizers	Normalizer	String	Enter the name of the normalizer for the incoming logs. Mandatory Field.
type	Type	String	Type of the normalizer. Mandatory Field.

4.6 Enrichment

In enrichment, you can select an enrichment policy for the incoming logs. Enrichment policies are used to add additional information to a log, such as user information, device type or geolocation, before analyzing it. For more information on enrichment, go to Enrichment Policies.

Parameters:

Field	Label in UI	Type	Description
EnrichmentPolicy	Enrichment Policy	String	Enter the name of the enrichment policy for the incoming logs. Optional Field.

Request Example:

```
{
   "data":{
      "name": "logsource_name",
      "type": "UniversalRestApi",
      "vendor_name": "",
      "logo": "",
      "description": "",
      "documentation_link": "",
      "config": {
         "Source": {
            "name": "source_name",
            "base_url": "https://10.45.9.123",
            "request_timeout": 30,
            "retry_after": 10,
            "interval": 30,
            "charset": "utf_8"
         },
         "Connector": {
```

(continues on next page)

(continued from previous page)


```
            "auth_type": "none",
            "custom_headers": [
            {
               "key": "id",
               "value": "15"
            }
            ],
            "enforce_https": true,
            "enable_proxy": false,
            "protocol": "http"
         },
         "Endpoints": [
            {
            "endpoint_name": "getApps",
            "method": "get",
            "endpoint": "/apps",
            "endpoints_custom_headers": [],
            "query_params": [],
            "incremental_value_response_field": "event",
            "log_filter_params_dataformat": "iso",
            "log_filter_params_from_value": "2023-10-05 11:13:47",
            "id": "bbf30918-8605-4f1f-8d7c-93ce3489d57e"
            }
         ],
         "RoutingPolicy": {
            "routing_criterion": [],
            "catch_all": "_logpoint"
         },
         "NormalizationPolicy": {
            "normalizers": [
            {
               "name": "ThycoticSecretServerCompiledNormalizer",
               "type": "compiled"
            },
            {
               "name": "JSONCompiledNormalizer",
               "type": "compiled"
            }
            ]
         },
         "EnrichmentPolicy": "642beb329fab980b50e4bb7e"
      }
   }
}
```


Success Response:

```
{
  "status": "Success",
  "message": "monitorapi/{pool_UUID}/{logpoint_identifier}/orders/{request_id}"
 }
```

4.7 Editing a Universal REST API Fetcher Configuration

You can edit a Universal REST API Fetcher configuration in a Fabric-enabled Logpoint using the **PluginConfiguration - Edit** API.

Endpoint URL:

```
https://api-server-host-name/configapi/{pool_UUID}/{logpoint_identifier}/LogSources/{id}
```

Method:

```
PUT
```

Parameters:

Field	Label in UI	Type	Description
id	–	String	Universal REST API Fetcher UUID. Obtain it using the **Devices - List** API. Mandatory

Request Example:

```
{
  "name": "UniversalRESTApi",
  "type": "UniversalRestApi",
  "vendor_name": "",
  "logo": "",
  "description": "",
  "config": {
   "Source": {
    "name": "UniversalRESTApi",
    "base_url": "https://api.stripe.com",
    "request_timeout": 30,
    "retry_after": 10,
    "interval": 15,
    "charset": "utf_8",
    "timezone": "UTC"
   },
```

(continues on next page)

(continued from previous page)


```
  "Connector": {
    "source_type": "DuoSecurityFetcher",
    "auth_type": "none",
    "custom_headers": [
      {
        "key": "NAME",
        "value": "USER"
      }
    ],
    "enforce_https": true,
    "enable_proxy": false,
    "protocol": "http"
  },
  "Endpoints": [
    {
      "endpoint_name": "EndpointName",
      "method": "post",
      "endpoint": "v1",
      "endpoints_custom_headers": [],
      "query_params": [],
      "incremental_value_response_field": "end_Data",
      "log_filter_params_dataformat": "iso",
      "log_filter_params_from_value": "2024-10-18 14:16:13",
      "id": "ffdc23c8-4269-4c8f-a5e4-02ec32112238",
      "fetch_status": "None",
      "last_fetch_attempt": "2024/10/24 09:06:21"
    }
  ],
  "RoutingPolicy": {
    "routing_criterion": [],
    "catch_all": "default"
  },
  "NormalizationPolicy": {
    "normalizers": [
      {
        "name": "LEEFCompiledNormalizer",
        "type": "compiled"
      },
      {
        "name": "SendMailCompiledNormalizer",
        "type": "compiled"
      }
    ]
  },
  "EnrichmentPolicy": "62e8a4fb82c2c51db550ba3d"
```


(continues on next page)

(continued from previous page)

```
 },
 "documentation_link": "",
 "id": "LogSources/0d9da10e-a925-4e53-aa67-b8fdc65bd2ce"
}
```

Success Response:

```
{
  "status": "Success",
  "message": "monitorapi/{pool_UUID}/{logpoint_identifier}/orders/{request_id}"
}
```

4.8 Deleting a Universal REST API Fetcher Configuration

You can delete a Universal REST API Fetcher configuration in a Fabric-enabled Logpoint using the **PluginConfiguration - Trash** API.

Endpoint URL:

```
https://api-server-host-name/configapi/{pool_UUID}/{logpoint_identifier}/LogSources/{id}
```

Method:

```
Delete
```

Parameters:

FIELD	LABEL IN UI	TYPE	DESCRIPTION	REQUIRED
id	–	String	Universal REST API Fetcher UUID. Obtain it using the **Devices - List** API.	Mandatory

Success Response:

```
{
  "status": "Success",
  "message": "monitorapi/{pool_UUID}/{logpoint_identifier}/orders/{request_id}"
}
```