LOGPOINT

Integrations

Universal REST API

V3.3.0

CONTENTS

1 Universal REST API Fetcher **1**

2 Installing Universal REST API Fetcher **2**

3 Uninstalling Universal REST API Fetcher **3**

4 Configuring Universal REST API Fetcher **4**

4.1 Source 4
4.2 Connector 5
4.3 Endpoints 7
4.4 Routing 13
4.5 Normalization 15
4.6 Enrichment 16

5 Accessing the Universal REST API Fetcher Logs **17**

CHAPTER

ONE

UNIVERSAL REST API FETCHER

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

Prerequisite

Logpoint v7.4.0 and later

To install:

1. Download the .pak file from the *Download* section in Release Notes.
2. Go to *Settings >> System Settings* from the navigation bar and click **Applications**.
3. Click **Import**.
4. **Browse** to the downloaded .pak file.
5. Click **Upload**.

After installing Universal REST API Fetcher, you can find it under *Settings >> System Settings >> Plugins*.

CHAPTER

THREE

UNINSTALLING UNIVERSAL REST API FETCHER

To uninstall Universal REST API Fetcher, you must first remove its configurations and also uninstall Duo Security and Cybereason.

To remove Universal REST API Fetcher configurations:

1. Go to *Settings >> Log Sources* from the navigation bar.
2. Click the (⋮) icon of Universal REST API Fetcher and click **Delete**.
3. Click **Delete**.

To uninstall Universal REST API Fetcher, Duo Security and Cybereason:

1. Go to *Settings >> System Settings* from the navigation bar and click **Applications**.
2. Click the **Uninstall** (🗑) icon in **Actions** of Universal REST API Fetcher, Duo Security and Cybereason.

CHAPTER

FOUR

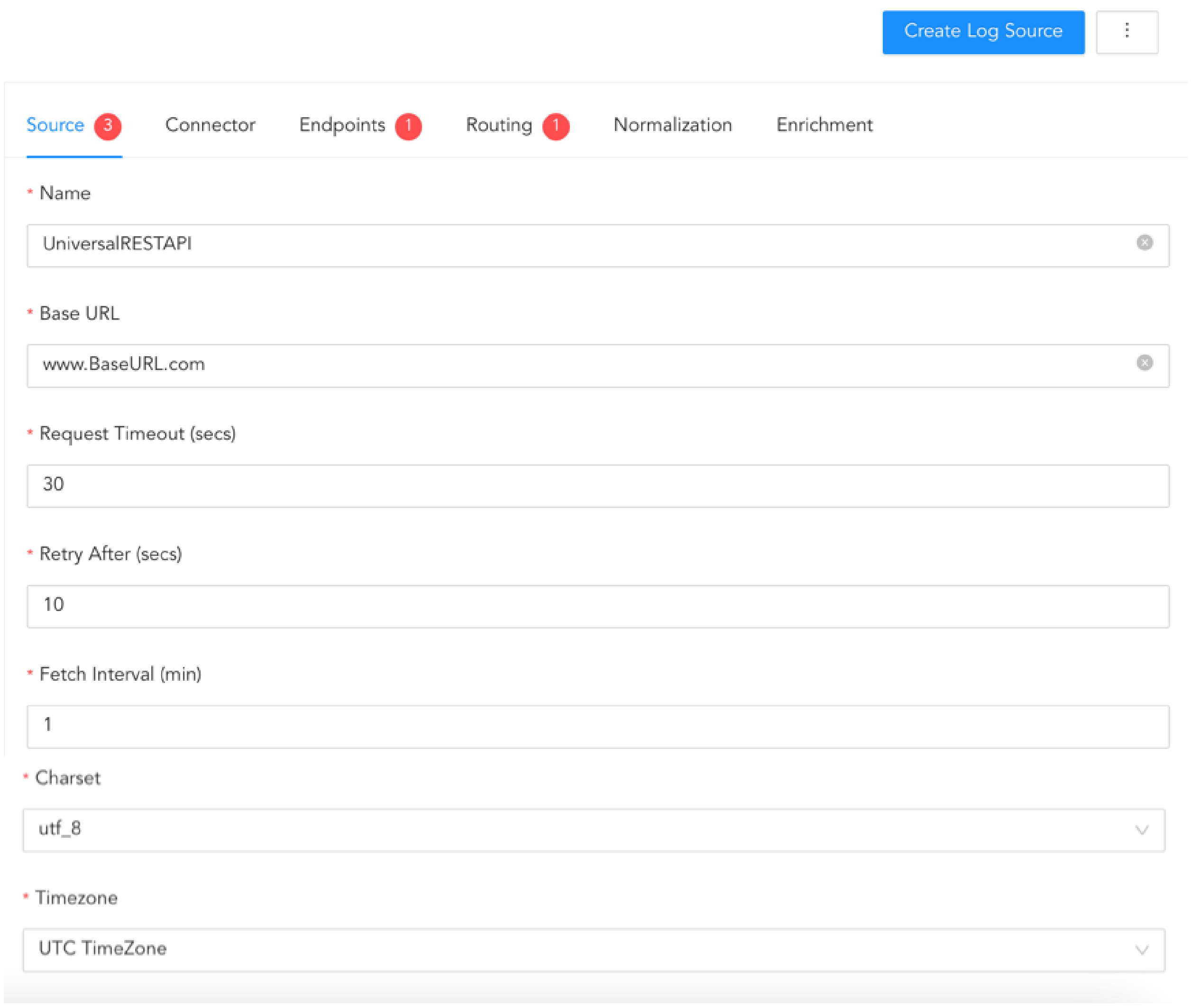
CONFIGURING UNIVERSAL REST API FETCHER

1. Go to *Settings >> Log Sources* from the navigation bar and click **Add Log Source**.
2. Click **Create New** and select **Universal Rest API**.

4.1 Source

In source, you can add details about the log source from where the Universal REST API Fetcher fetches logs.

1. Click **Source**.
2. Enter the Log Source's **Name**.
3. In **Base URL**, enter the RESTful API.
4. Enter **Request Timeout (secs)** for the API request.
5. In **Retry After(secs)**, enter the time to wait after an error or timeout.
6. Enter the **Fetch Interval (min)** and select **Charset**.
7. Select the **Timezone** of the log source if its response time is not in UTC. Universal REST API Fetcher automatically sets the time in case of UTC.



* Name

UniversalRESTAPI

* Base URL

www.BaseURL.com

* Request Timeout (secs)

30

* Retry After (secs)

10

* Fetch Interval (min)

1

* Charset

utf_8

* Timezone

UTC TimeZone

Fig. 1: Configuring Source

4.2 Connector

Connector is a pathway for transmitting logs from various sources to Logpoint. In connector, you can configure how the Universal REST API Fetcher and the log source communicate with each other.

1. Click **Connector**.
2. Select the **Authorization Type**.
   2.1. Select **No Auth** if no authentication is required.
   2.2. Select **Basic** to use a username and password to authenticate.
      2.2.1. In **Credentials**, enter the **Username** and **Password**.

2.3. Select **OAuth2** to authenticate using OAuth authentication. Enter the following details in **OAUTH 2.0 BASIC INFORMATION**.

2.3.1. Enter the **Token URL** of the server.
2.3.2. Select either **Client Credentials** or **Password Credentials** as the **Grant Type**.

2.3.2.1. If you select **Client Credentials**, enter the OAuth secret password in **Client Secret**.
2.3.2.2. If you select **Password Credentials**, enter the OAuth **Username** and **Password**.

2.3.3. In **Client ID**, enter the OAuth application ID or client ID. If the vendor requires a client secret, enter **Client Secret**.
2.3.4. In **API Key Prefix**, enter the prefix to add to the authorization header before the API Key or Token.
2.3.5. Select whether to send the client credentials as a basic auth header or in the body.
2.3.6. Enter the extra parameters key and its value in **ADDITIONAL BODY FOR OAUTH 2.0**.

2.4. Select the **API Key** to authenticate using an API Key.

2.4.1. In **Secret Key**, enter **API Key**. This API key is used in the authorization header.
2.4.2. In **API Key Prefix**, enter the prefix to add to the authorization header before API Key or Token. This is optional.

2.5. Select **Digest** to authenticate using digest access authentication.

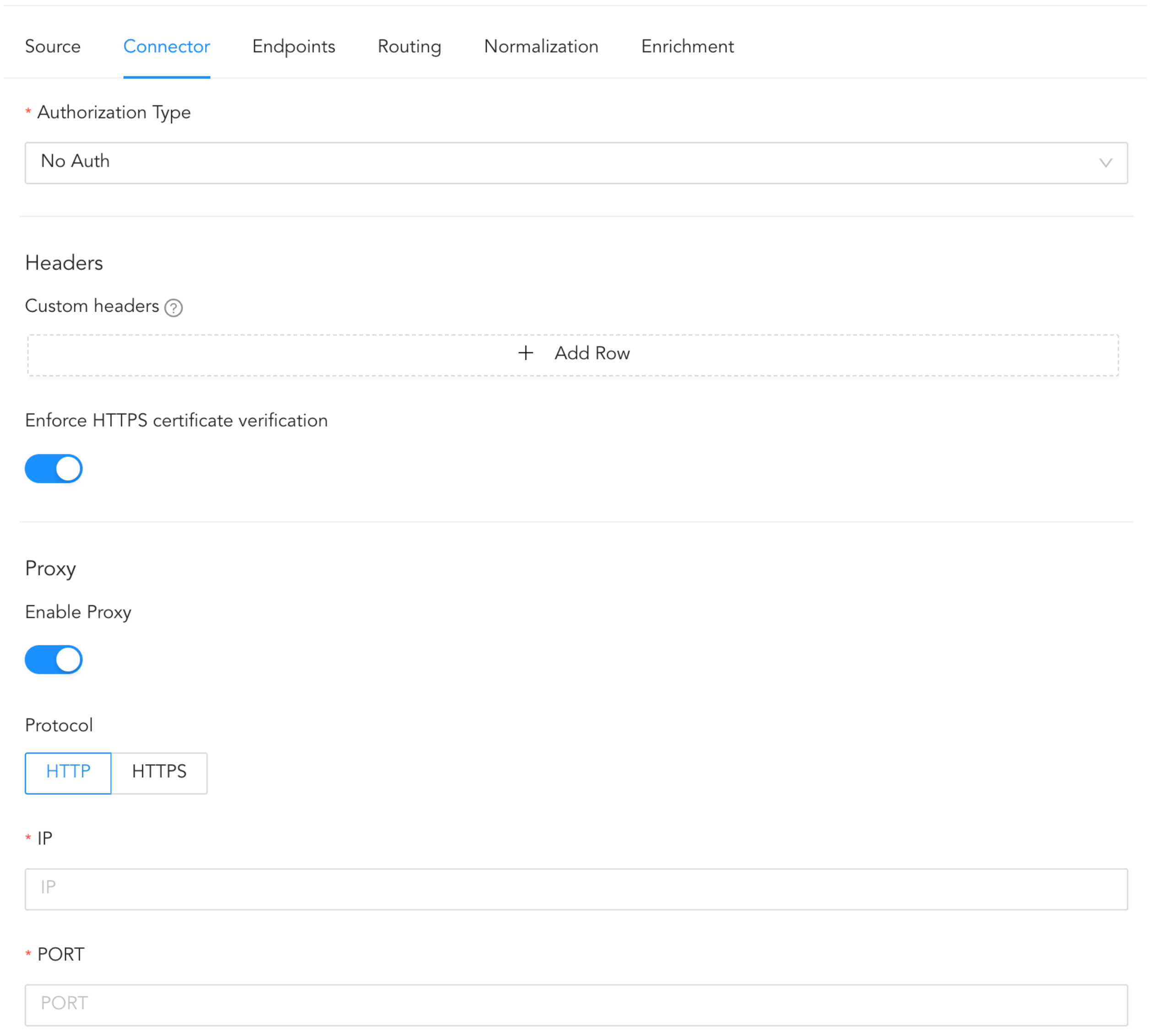
2.5.1. In **Credentials**, enter the **Username** and **Password**.

2.6. Select **Custom** to authenticate using integration that applies custom authentication mechanisms and request handling.

2.6.1. Select a **Product**. Here, you can see the integrations supported by Universal REST API Fetcher, such as Duo Security and Cybereason, that require custom authentication mechanisms and request handling. They must also be installed on Logpoint.

3. Enter the RESTful API custom headers in **Key** and **Value**.
4. Enable **Enforce HTTPS certificate verification** to enable a secure connection.
5. **Enable Proxy** to use a proxy server.

5.1. Select either **HTTP** or **HTTPS** protocol.
5.2. Enter the proxy server **IP** address and the **PORT** number.



* Authorization Type

No Auth

Headers

Custom headers

+ Add Row

Enforce HTTPS certificate verification

Proxy

Enable Proxy

Protocol

HTTP | HTTPS

* IP

IP

* PORT

PORT

Fig. 2: Configuring Connector

4.3 Endpoints

In enpoints, you can configure details about the log source endpoints.

1. Click **Endpoints** and **Add Row**.
2. Enter the endpoint's **Name**.
3. Select the HTTP request **Method**.
   3.1. If you select **GET**, continue to Step 4.
   3.2. If you select **POST**, enter the **Post request body** in JSON format.

Example:

```
{
  "filters": [
    {
      "fieldName": "<field>",
      "operator": "<operator>",
      "values": "[value]"
    }
  ],
  "search": "<value>",
  "sortingFieldName": "<field>",
  "sortDirection": "<sort direction>",
  "limit": "<limit>",
  "offset": "<page number>"
}
```

Important: You can define the time range for fetching logs in the **Post request body**:

- Use the Jinja keyword *{{start}}* for beginning of the log fetching window.
- Use *{{end}}* for endpoint of the time range.

These values are dynamically replaced with actual timestamps during log fetching. For endpoints where specifying *{{end}}* is optional, omitting it causes the current time to be used instead.

Example:

```
{
  "filters": [
    {
      "fieldName": "StartTimestamp",
      "operator": "equals",
      "values": "{{start}}"
    },
    {
      "fieldName": "EndTimestamp",
      "operator": "equals",
      "values": "{{end}}"
    }
  ]
}
```

In this example, *StartTimestamp* and *EndTimestamp* are the beginning and end of the fetch window.

4. Enter the **Endpoint** part of the previously *added* Base URL .

5. Enter a **Description** for the endpoint.
6. Under **Headers**, click **+ Add Row**. Enter the **Key** and **Value** for each custom header.

Note:

- Avoid using common log filtering fields like *start_date* or *end_date* as headers.
- Do not add *Authorization* as a custom header.

7. In **Query Parameters**, click **+ Add Row**. Enter the **Key** and **Value** as required by the API.

Example:

For a query like */api/alerts?$filter=(severity eq 'High') or (severity eq 'Medium')*, enter:

- **Key**: *$filter*
- **Value**: *(severity eq 'High') or (severity eq 'Medium')*

Query parameters are sent as part of the request URL.

Important: To define a time range using query parameters:

- Use *{{start}}* for the start timestamp.
- Use *{{end}}* for the end timestamp.

Example:

Key	Value
StartTimestamp	*{{start}}*
EndTimestamp	*{{end}}*

8. In **Increment Value / Check Sum**, enter the path to the field that tracks progress in log fetching.

Example:

If the field is *event_date* within an *Events* object, enter *Events.event_date*. This field's value from the most recent log is stored in the CheckSum database. During the next collection cycle, it becomes the *{{start}}* value, ensuring no duplicate logs are collected.

9. Enter the **Response Key**. This is used to locate and extract log records from the API response.

10. Enter the **Custom Date Format** expected in the API response.

Some of them are:

Date Type	Format	Example
UTC	%Y-%m-%dT%H:%M:%SZ	2023-04-27T07:18:52Z
ISO-8601	%Y-%m-%dT%H:%M:%S%z	2023-04-27T07:18:52+0000
RFC 2822	%a, %d %b %Y %H:%M:%S %z	Thu, 27 Apr 2023 07:18:52 +0000
RFC 850	%A, %d-%b-%y %H:%M:%S UTC	Thursday, 27-Apr-23 07:18:52 UTC
RFC 1036	%a, %d %b %y %H:%M:%S %z	Thu, 27 Apr 23 07:18:52 +0000
RFC 1123	%a, %d %b %Y %H:%M:%S %z	Thu, 27 Apr 2023 07:18:52 +0000
RFC 822	%a, %d %b %y %H:%M:%S %z	Thu, 27 Apr 23 07:18:52 +0000
RFC 3339	%Y-%m-%dT%H:%M:%S%z	2023-04-27T07:18:52+00:00
ATOM	%Y-%m-%dT%H:%M:%S%z	2023-04-27T07:18:52+00:00
COOKIE	%A, %d-%b-%Y %H:%M:%S UTC	Thursday, 27-Apr-2023 07:18:52 UTC
RSS	%a, %d %b %Y %H:%M:%S %z	Thu, 27 Apr 2023 07:18:52 +0000
W3C	%Y-%m-%dT%H:%M:%S%z	2023-04-27T07:18:52+00:00
YYYY-DD-MM HH:MM:SS	%Y-%d-%m %H:%M:%S	2023-27-04 07:18:52
YYYY-DD-MM HH:MM:SS am/pm	%Y-%d-%m %I:%M:%S %p	2023-27-04 07:18:52 AM
DD-MM-YYYY HH:MM:SS	%d-%m-%Y %H:%M:%S	27-04-2023 07:18:52
MM-DD-YYYY HH:MM:SS	%m-%d-%Y %H:%M:%S	04-27-2023 07:18:52

11. In **Logs Filtering Parameters**, select the parameters to filter the incoming logs.

    11.1. Select a **Data format**.

    11.1.1. Select **ISO Date** to represent data using the International Standards Organization (ISO) format of "yyyy-MM-dd". Example: 2017-06-10.

    Note: If you select **ISO Date**, then its value must be in the string format in the **Post request body**.

11.1.2. Select **UNIX Epoch** to represent data using the UNIX epoch time format. It is a system for measuring time as the number of seconds that have elapsed since January 1, 1970, at 00:00:00 UTC (Coordinated Universal Time). Example: 1672475384.

11.1.3. Select **UNIX Epoch (ms)** to represent data using the UNIX epoch time format with milliseconds precision. It is a system for measuring time as the number of milliseconds that have elapsed since January 1, 1970, at 00:00:00 UTC (Coordinated Universal Time). Example:1672475384000.

11.1.4. Select **Custom Format** to define your own format for representing the data. The custom format can be created using *Date/Time patterns*.

11.1.5. Select **Unique ID** to represent data using a unique ID.

Note: If you select **Unique ID** here, then its value must be in the number format in the **Post request body**.

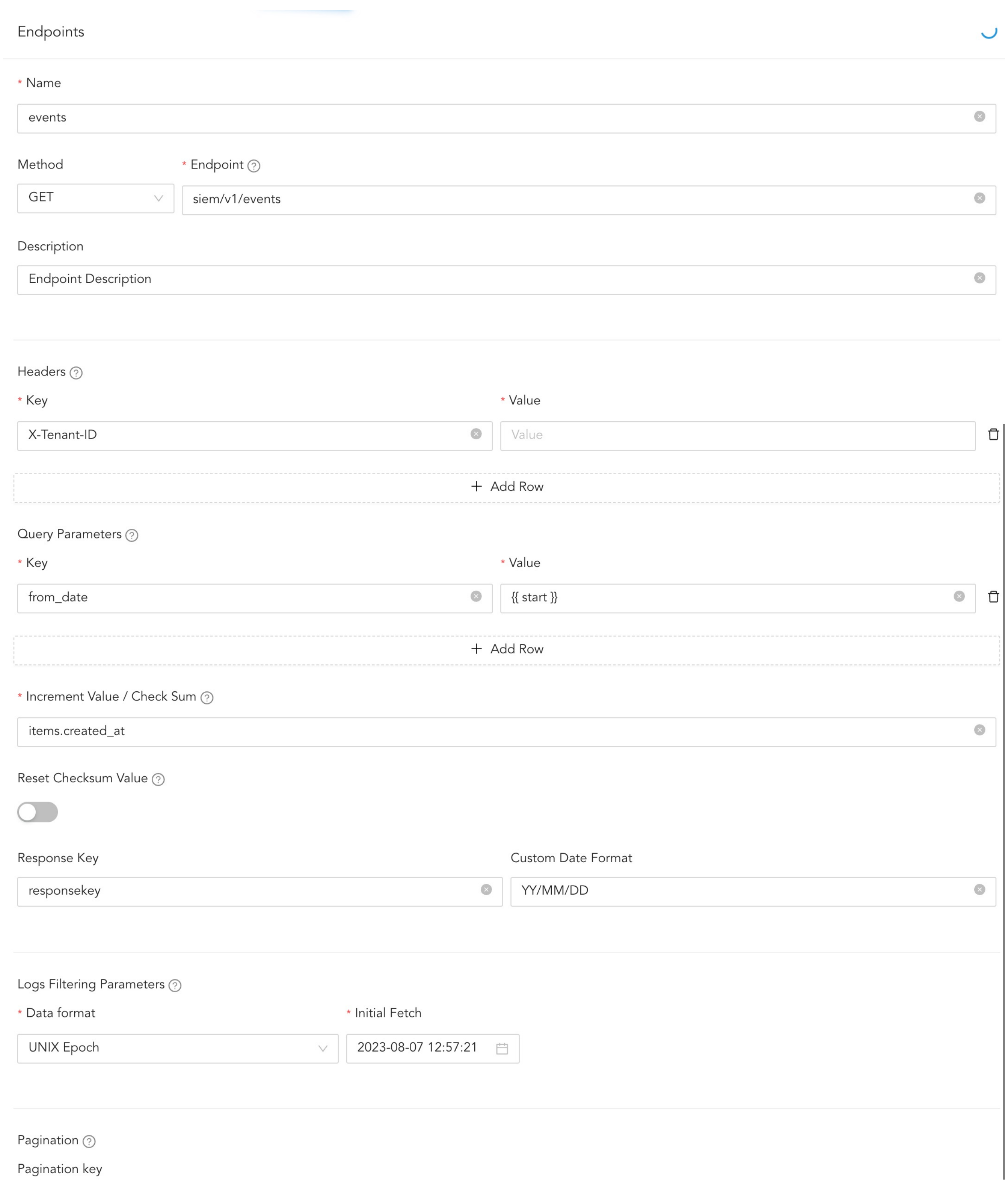
12.2. Select an **Initial Fetch** date. Logs are fetched for the first time from this date.

13. In **Pagination Key**, enter the location of the following page URL from the response if the API supports pagination.

    For example, if the data from the RESTful API looks like the following, the pagination key is *metadata.links.next*.

```
"metadata": {

"links": {

        "self": "https://api.com/audit_logs",

        "next": "https://api.com/audit_logs?offset=500"

      }

      }
```

14. Click **Save Changes**.



Endpoints

* Name

events

Method: GET

* Endpoint: siem/v1/events

Description

Endpoint Description

Headers

* Key	* Value
X-Tenant-ID	Value

Add Row

Query Parameters

* Key	* Value
from_date	{{ start }}

Add Row

* Increment Value / Check Sum

items.created_at

Reset Checksum Value

Response Key	Custom Date Format
responsekey	YY/MM/DD

Logs Filtering Parameters

* Data format	* Initial Fetch
UNIX Epoch	2023-08-07 12:57:21

Pagination

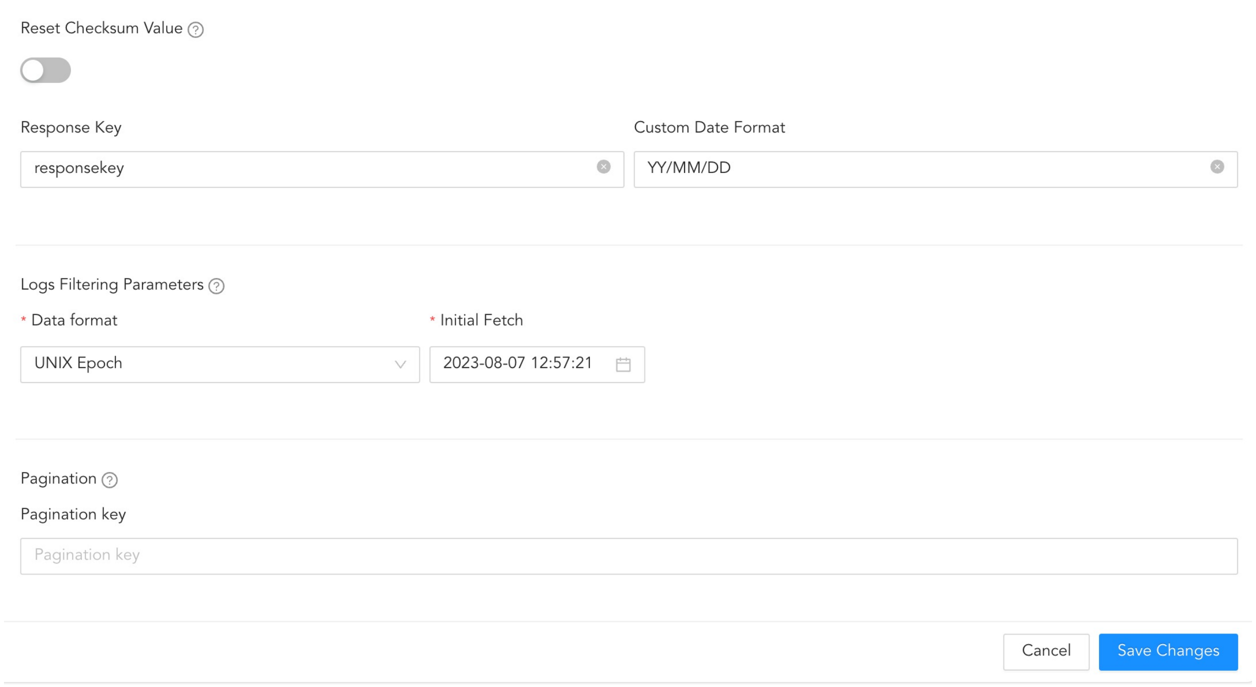
Pagination key

Fig. 3: Configuring Endpoint

To edit the endpoint configuration, click the (⋮) icon under **Action** and click **Edit**. Make the necessary changes and click **Save Changes**.

To delete the endpoint configuration, click the (⋮) icon under **Action** and click **Delete**.

To reset the Checksum values, toggle **Reset**.


Reset Checksum Value

Response Key: responsekey

Custom Date Format: YY/MM/DD

Logs Filtering Parameters

* Data format: UNIX Epoch

* Initial Fetch: 2023-08-07 12:57:21

Pagination

Pagination key: Pagination key

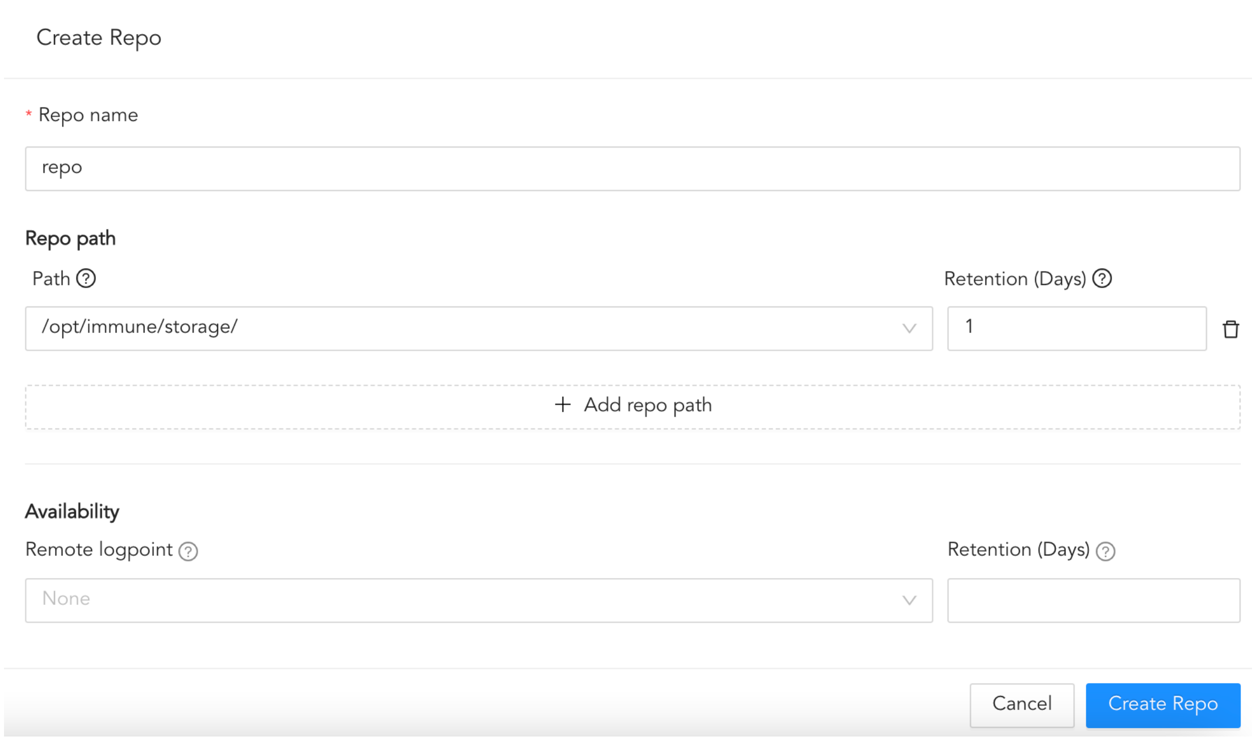
Fig. 4: Reseting Checksum

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 is created to determine the conditions under which these logs are sent to repos.

To create a repo:

1. Click **Routing** and **+ Create Repo**.
2. Enter a **Repo name**.
3. In **Path**, enter the location to store incoming logs.
4. In **Retention (Days)**, enter the number of days logs are kept in a repository before they are automatically deleted.
5. In **Availability**, select the **Remote logpoint** and **Retention (Days)**.
6. Click **Create Repo**.



Repo name: repo

Path: /opt/immune/storage/ Retention (Days): 1

Remote logpoint: None

Fig. 5: Creating a Repo

In **Repo**, select the created repo to store logs.

To create Routing Criteria:

1. Click **+ Add row**.
2. Enter a **Key** and **Value**. The routing criteria is only applied to those logs which have this key-value pair.
3. Select an **Operation** for logs that have this key-value pair.

    3.1. Select **Store raw message** to store both the incoming and the normalized logs in the selected repo.
    3.2. Select **Discard raw message** to discard the incoming logs and store the normalized ones.
    3.3. Select **Discard entire event** to discard both the incoming and the normalized logs.

5. In **Repository**, select a repo to store logs.

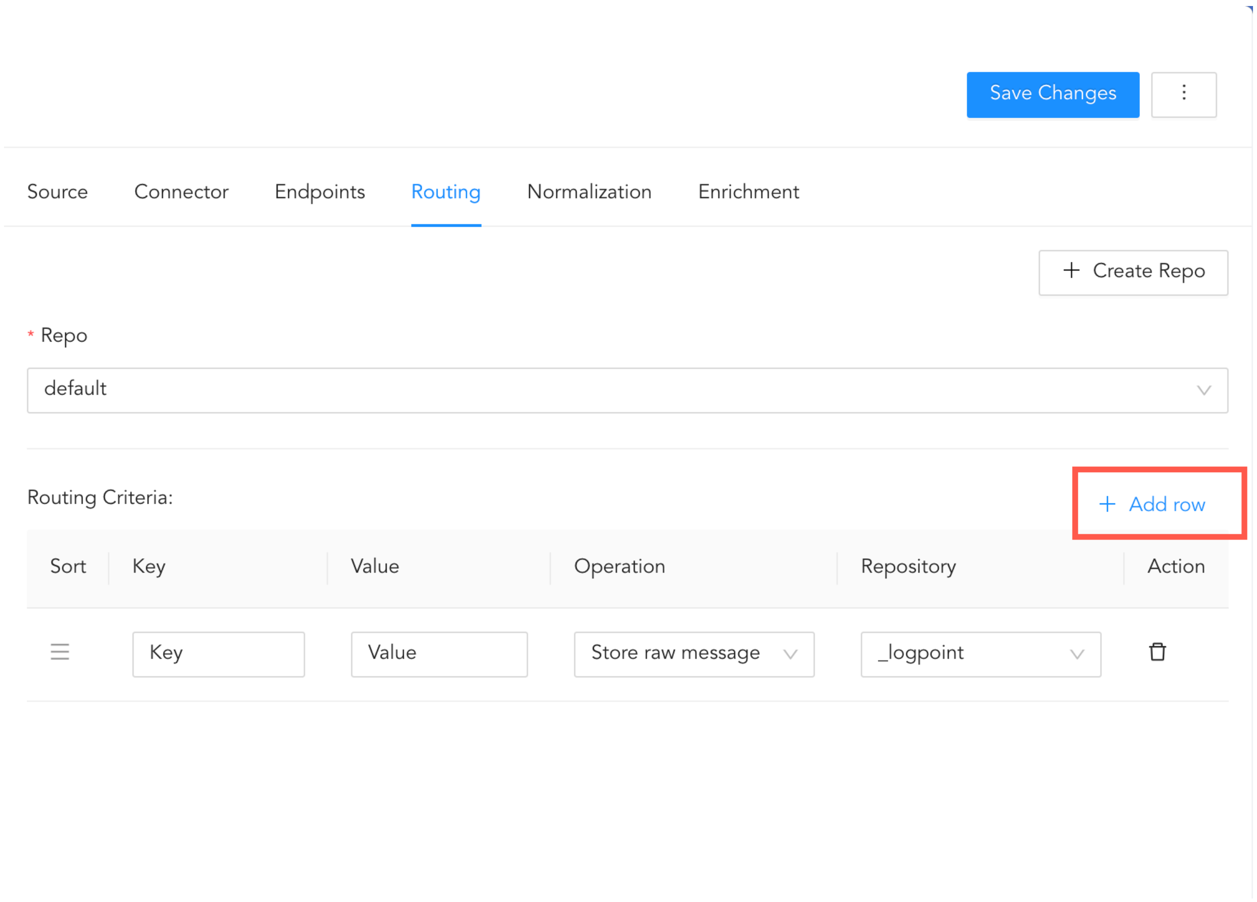


Fig. 6: Creating a Routing Criteria

Click the (🗑) icon under **Action** to delete the created routing criteria.

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.

1. Click **Normalization**.
2. You can either select a previously created normalization policy from the **Select Normalization Policy** dropdown or select a **Normalizer** from the list and click the swap(>) icon.

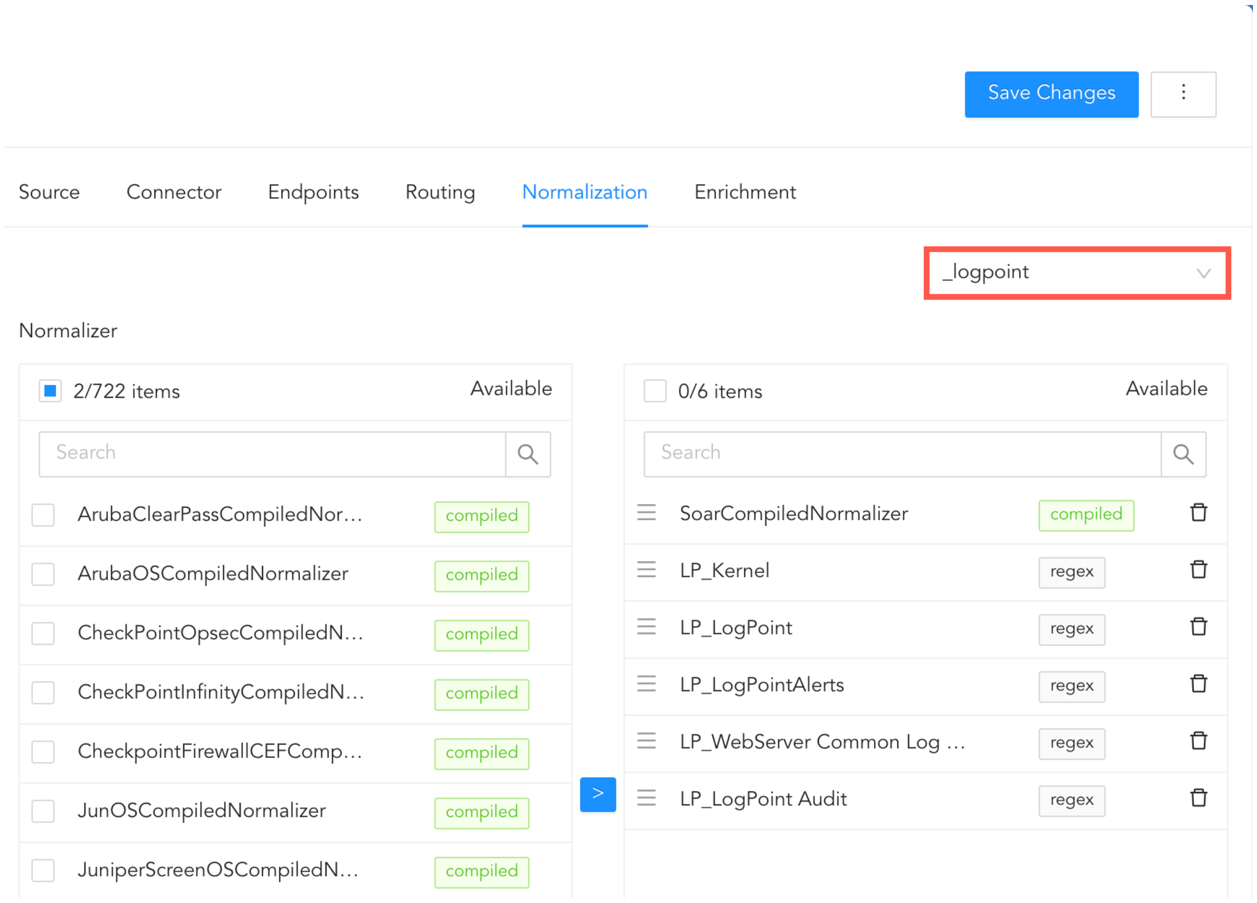


Fig. 7: Adding Normalizers

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.

1. Click **Enrichment**.
2. Select an **Enrichment Policy**.

Click **Create Log Source** to save the configurations of Source, Connector, Endpoints, Routing, Normalization and Enrichment.

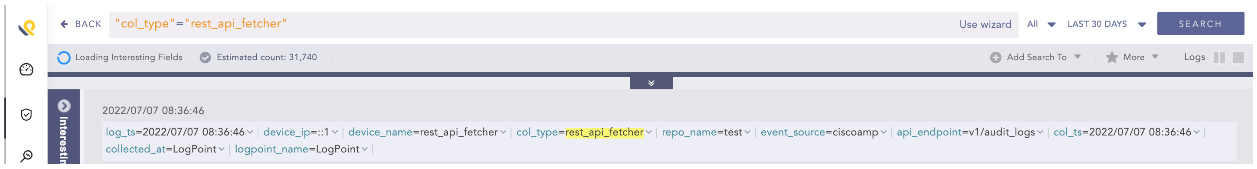
CHAPTER

FIVE

ACCESSING THE UNIVERSAL REST API FETCHER LOGS

Use the following query to access the logs:

```
col_type = rest_api_fetcher
```



```
"col_type"="rest_api_fetcher"
Estimated count: 31,740
2022/07/07 08:36:46
log_ts=2022/07/07 08:36:46 | device_ip=::1 | device_name=rest_api_fetcher | col_type=rest_api_fetcher | repo_name=test | event_source=ciscoamp | api_endpoint=v1/audit_logs | col_ts=2022/07/07 08:36:46 |
collected_at=LogPoint | logpoint_name=LogPoint |
```

Fig. 1: Universal REST API Fetcher Sample Log