

***Collibra - Okera Integration Design Document***

***V2.0***

***December 28th 2022***

Table of Contents

[1 Introduction 3](#_Toc123906721)

[2 Requirements 3](#_Toc123906722)

**3 Configuration Details**………………………………………………………………………………………………………………………3

**4 How to Run the Application**…………………………………………………………………………………………………………….3

5 Design………………………………………………………………………………………………………………………………………………6

5.1 Extracting from Collibra……………………………………………………………………………………………………………….6

5.2 Mapping and Transforming………………………………………………………………………………………………………….9

**5.3 High level flow**…………………………………………………………………………………………………………………………….10

**5.4** **Status API**…………………………………………………………………………………………………………………………………..11

**6 Open items**………………………………………………………………………………………………………………………………………11

# **Introduction**

This document describes the high-level design for asset ingestion from Collibra to Okera.

# **Requirements**

The DGC and Okera instance details are configured in the property file and can be modified before installation. The attributes of Collibra assets will be imported into Okera as tags.

1. **Configuration Details**

Below are details that user can modify in the configuration file (app. properties).

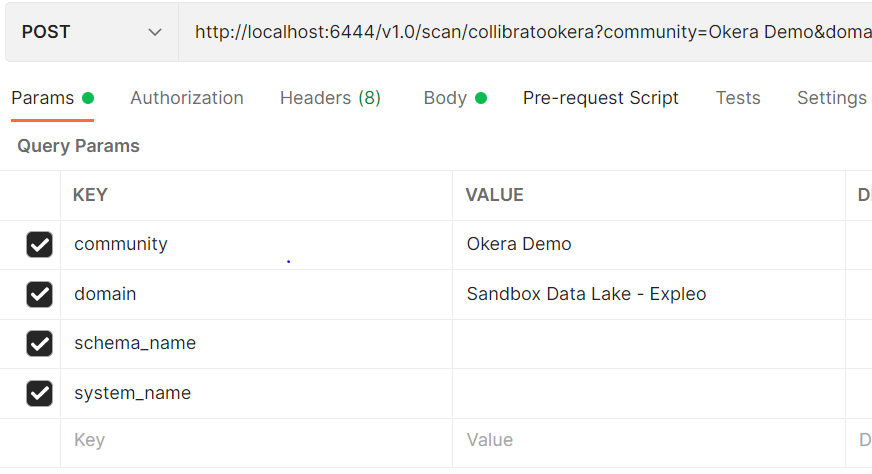
|  |  |  |  |
| --- | --- | --- | --- |
|  | **parameters** | **Default Value** | **Description** |
| **Collibra Configuration Details** | dgc.username | username | Username of the Collibra instance |
| dgc.password | password | A Collibra instance's password in base64 format. |
| dgc.baseurl | https://hostname:port | The Collibra instance's base URL (host and port). |
| **Okera Configuration Details** | okera.host | hostname | Host name of the Okera instance |
| okera.port | Port number | Port number of the Okera instance |
| okera.token | Token in string format | A token of Okera instance |
| okera.namespace | collibra\_classification | This namespace will be used for to update Collibra attributes and tags in Okera instances. |
| okera.namespace\_status | collibra\_status | This namespace will be used to update the status of table in Okera instance. |
| **Common Configuration details** | okera.sync description | false | If the user also wants to compare and modify the description or comments, then this parameter should be true; otherwise, it should be false. |
| dgc.loaded from edge | false | If assets are being extracted from the edge server, keep this parameter set to true; otherwise, set it to false. The user **can extract Collibra assets either from the Edge server or the Job server, not from a combination of both.** |
| dgc.full\_name\_delimiter | > | Delimiter use in Collibra instance for full names. |
| okera.dataset\_name\_delimiter | . | Delimiter use in Okera instance for the dataset name. |
| dgc.security\_classification\_as\_relation | false | If user want to give security classification tags as a relation, then set this parameter as ‘true’, otherwise set it as ‘false’ |

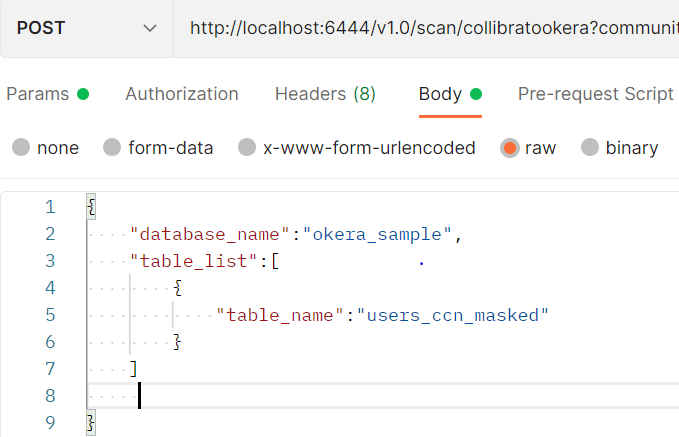
1. **How to Run the Application:**

To run the application, follow the below steps:

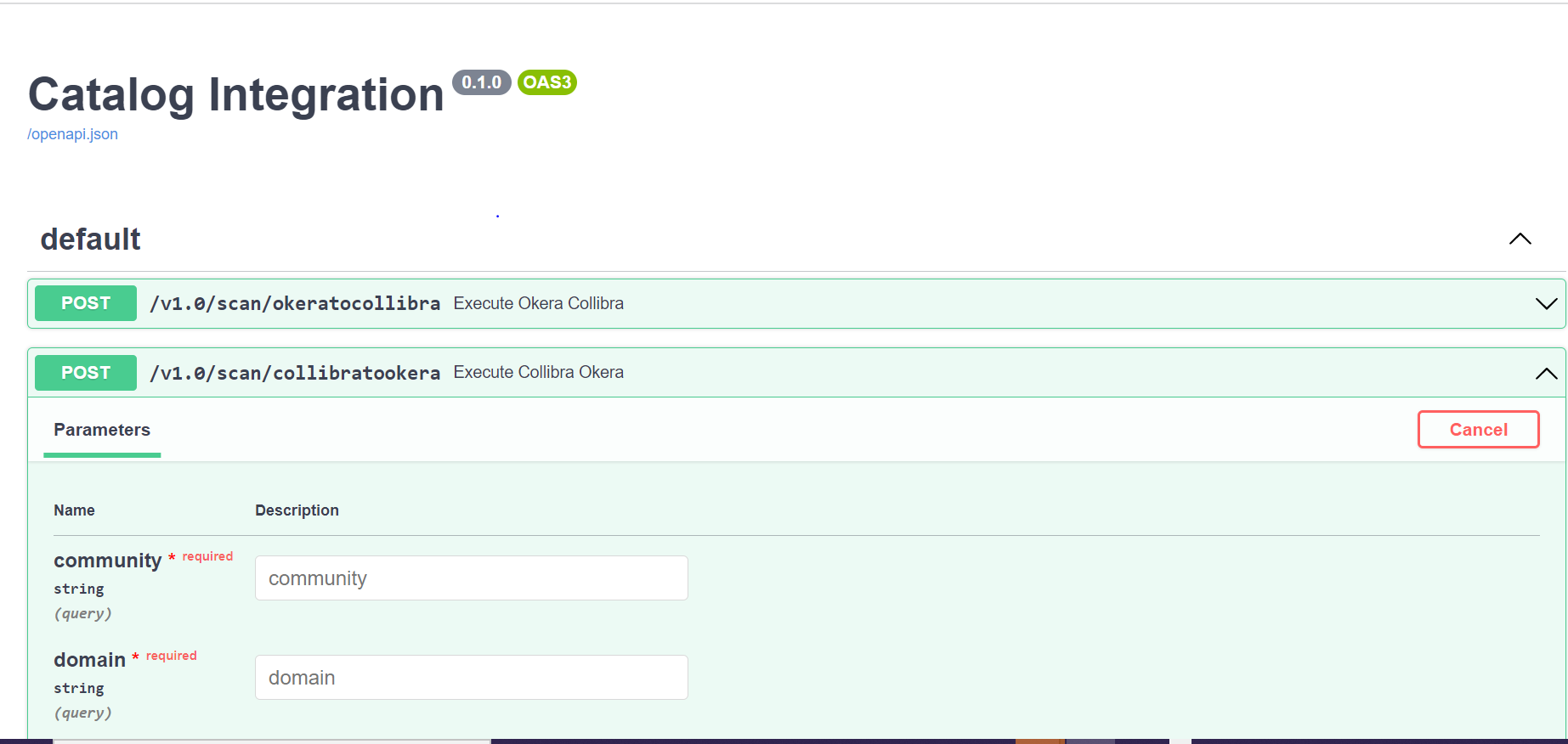
* Change the property file to suit your needs.
* After you have set the property file, run the following command in Terminal: <python main.py>
* Give the following parameters in the Postman/Swagger UI:
  + - Community: **Required**; community name
    - Domain: **Required**; domain name
    - Schema\_name: Optional, name of the schema.
    - System\_name: Optional, name of the system.
* Provide the following parameters in the body of the postman/swagger UI.
* Database\_name: **Required**, database name.
* Table\_name: Optional. If the user wants to fetch only specific tables, then give table names. If not, the application will retrieve all tables from the specified database.

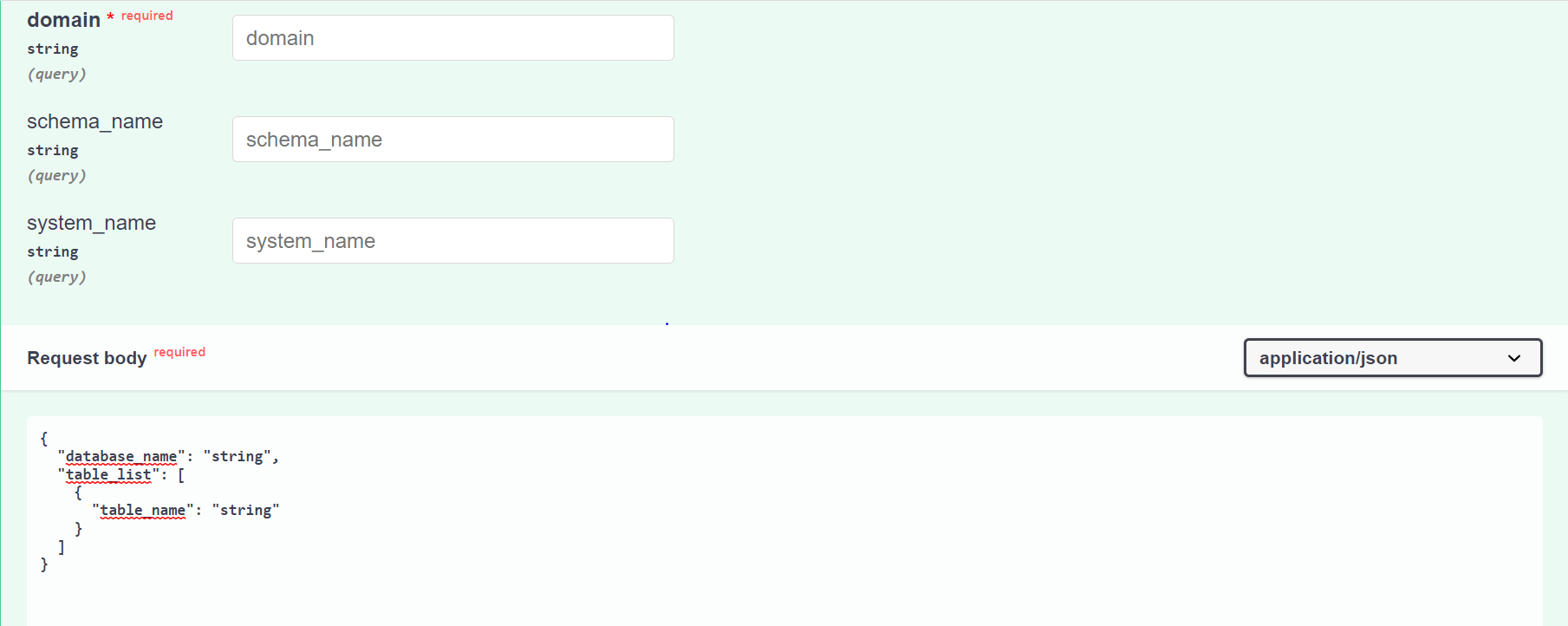
**4.1 Using Postman:**





* 1. **Using Swagger UI:**
* After running main.py, go to your browser and use <http://localhost:port/docs> (port = Collibra port number) to use the Swagger UI.





# **Design**

## **5.1 Extraction from Collibra**

The HTTP GET method will fetch the list of datasets from the source. The source URL can be configured in the property file.

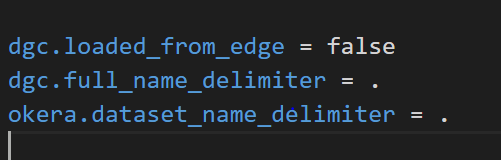
User can fetch and load metadata from Collibra in two ways:

1. **Using Job server:**

* If using the job server, the following changes should be made to the configuration (app.properties) file:

“dgc.load\_from\_edge”: parameter should be false.

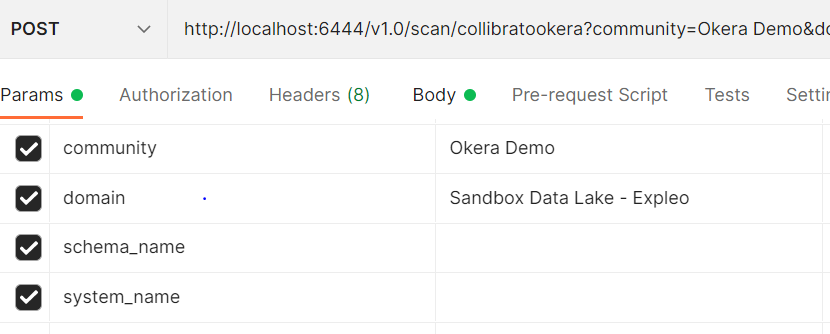
“dgc.full\_name\_delimiter”: delimiter use for full name attribute in Collibra.



* When fetching the details from the Job server, below parameters should be given as an input to postman/Swagger UI:

community: **Required**. Name of the community.

Domain: **Required**. Name of the domain.



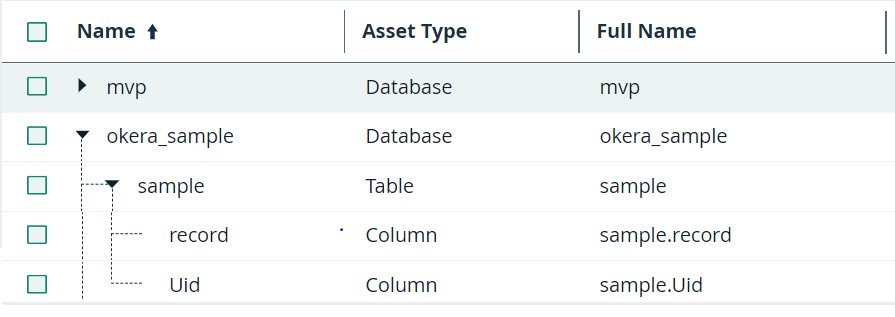
* + - * In Collibra instance, The asset's full name must be in the following format:

Database name: **Required** . Same as display name

Table name: optional. Same as display name

Column name: <Table\_name.Column\_name>

Example of ‘okera\_sample’ database:

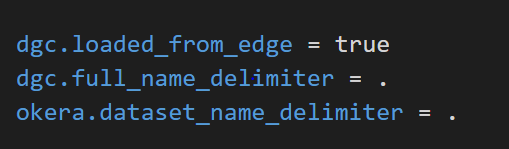


1. **Using Edge server:**

* When fetching the details from the Edge server, below parameters should be given as an input to postman/Swagger UI:

“dgc.load\_from\_edge” parameter should be true.

“dgc.full\_name\_delimiter”: delimiter use for full name attribute in Collibra.



* When fetching the details from the edge server, below parameters should be given as an input to the postman/swagger UI:

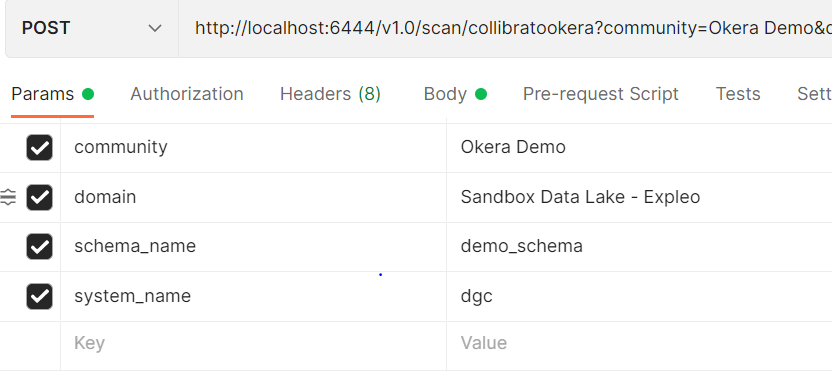
Community: **Required**. Name of the community.

Domain: **Required**. Name of the domain.

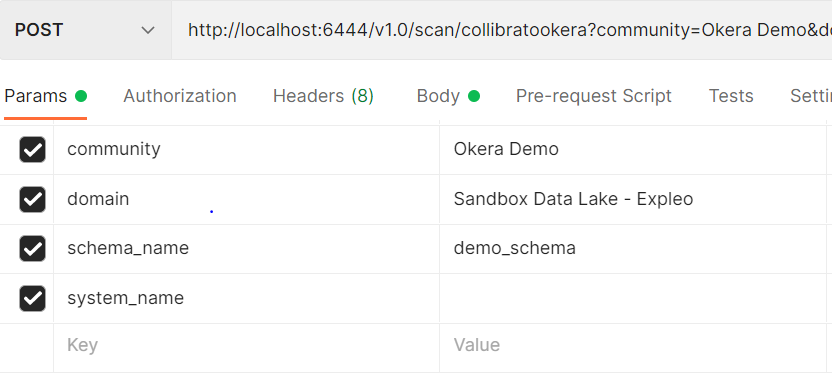
Schema\_name : **Required**. Name of the Schema

System\_name: System name is **optional**. Leave this parameter blank if the user does not want to provide/use a system name.

Example with System name:



Example without system name:



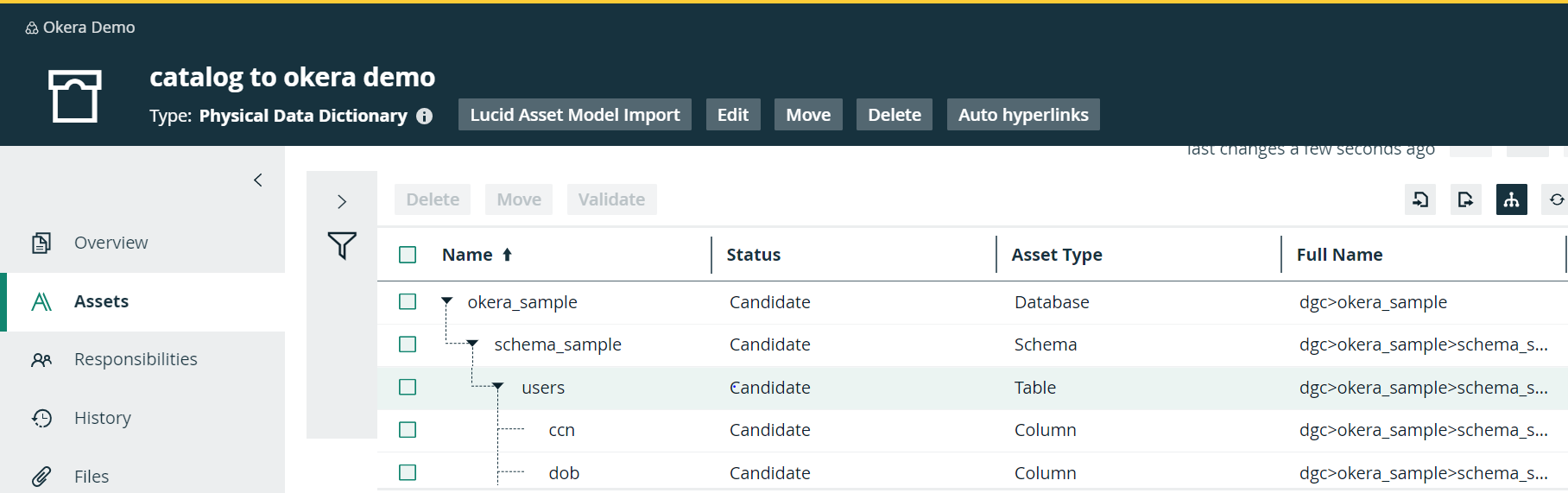
* + - * In Collibra instance, the asset's full name must be in the following format:

Database name: Same as the display name. If system name is specified, then <system\_name.database\_name>

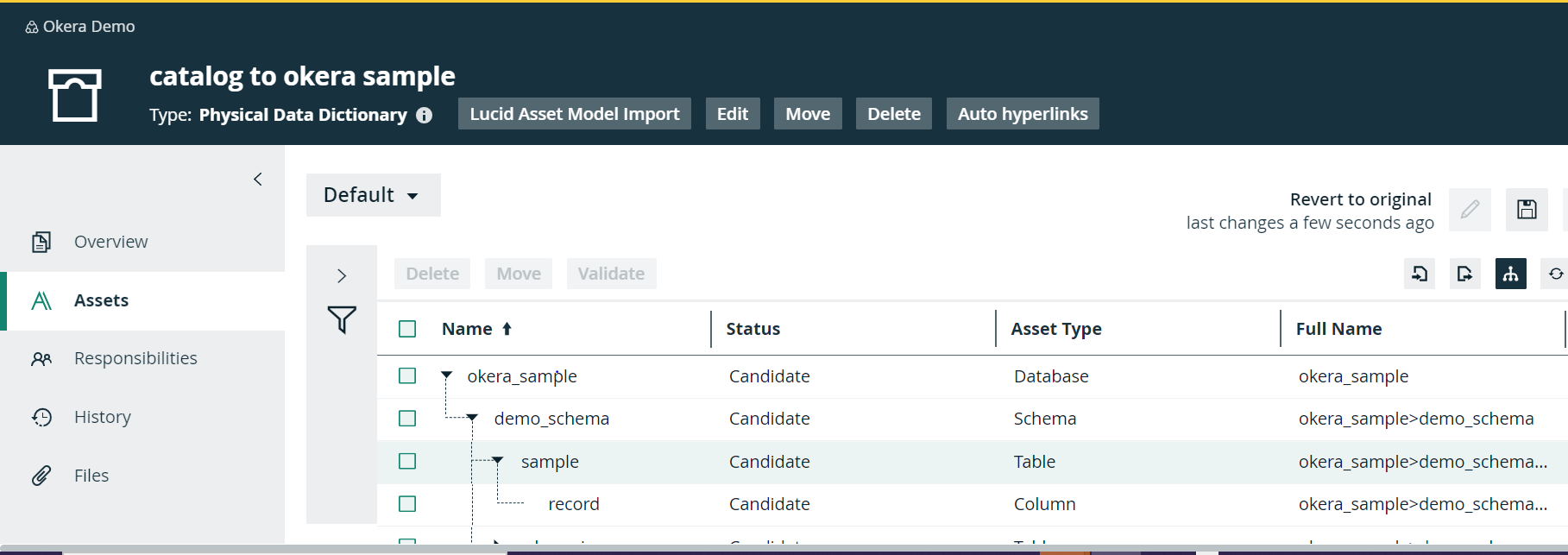
Table name: <Database\_name.Schema\_name.Table\_name>. If system name is specified, then <System\_name.Database\_name.Schema\_name.Table\_name>.

Column name: <Database\_name.Schema\_name.Table\_name.Column\_name>. if system name is specified then <System\_name.Database\_name.schema\_name.Table\_name.Column\_name>.

Example of “okera\_sample” database with system name:



Example of “okera\_sample” database without system name:



## **5.2 Mapping and Transformation**

Mapping between Okera Assets and DGC Assets

|  |  |  |  |
| --- | --- | --- | --- |
| **Okera (Target) Asset types** | **Collibra Parent Asset Types** | **If assets created from job server** | **If assets created from edge server** |
| OkeraDatabase | Database | Full name to be qualified same as name | Full name to be qualified as  “(system\_name > Database)” |
| Okera schema | Schema | Full name to be qualified same as name | Full name to be qualified as  “(system\_name > Database > Schema)” |
| OkeraModel | Table | Full name to be qualified same as name | Full name to be qualified as  “(system\_name > Database > Schema > table)” |
| OkeraColumn | Column | Full Name to be qualified as “(Table name > Column name)” | Full name to be qualified as  “(system\_name > Database > Schema > table > column\_name)” |

## **5.3 High Level Flow**

5.3.1 Collibra to Okera:

* Fetch the metadata and tags for table or database mention in a configurable file from Collibra.
* Fetch the metadata from the Okera datasets or tables mentioned in the configurable file for the corresponding Collibra metadata asset.
* Compare the tags of the Collibra asset and the Okera dataset. If any difference is found, perform an assign or unassign action accordingly to modify the tags in Okera.
* Only tags within namespaces that are defined in the configuration file are modified. Existing tags (without namespace) in Okera are never overwritten.
* Compare the description or comments (if specified in the configurable file) of Collibra Asset and Okera Database. If a discrepancy is discovered, change the description in Okera to match the Collibra Asset description.

Fetch data from Collibra json

Collibra

Perform actions to modify tags, desc in Okera

Fetch data from Okera

Okera

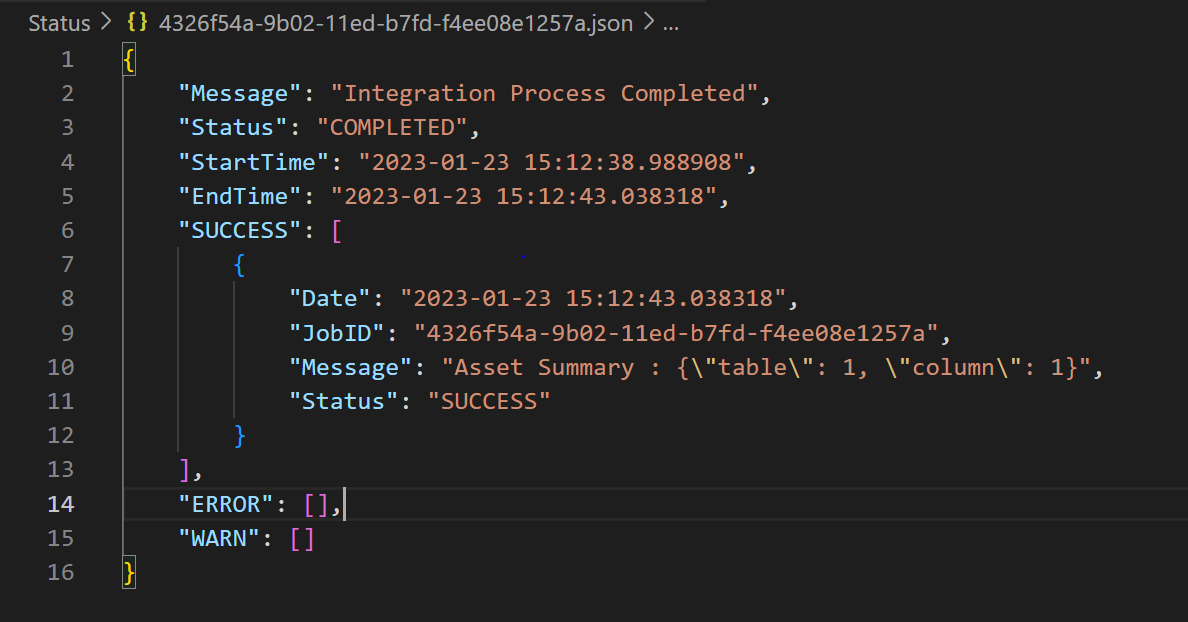
If difference found

Compare Collibra metadata to Okera

## **5.4 Status API:**

* After executing the script, in the Status folder, the status of the execution will be stored with sync\_id/job\_id as a file name.
* This file will contain the execution status along with details.
* If data is fetched from both instances, i.e., Collibra and Okera, and if the data is correct, then the status of the execution will be "SUCCESS" and in the message, table count, and column count will get mentioned.
* If data is not able to be fetched either from Collibra or Okera, or if the data is not correct, the status of the execution will be "ERROR" with the corresponding message.

Example of Status API:



# **Open items:**

* Endpoint expose in https.
* Basic auth for endpoints.