Nitin Pai

Nitin.pai@centricsoftware.com

Version 1.1

Rest API – sHADOW TABLE

**Rest api for shadow table data**

**Requirements:**

1. To create a shadow table for Supplier and Material Business object given the set of attributes that are defined in the view.
2. To create a rest api that would help retrieve data from Centric.

**Assumptions:**

1. The view name on the centric8 ui is unique across the system.
2. Middleware team can access the centric server via proper authentication.
3. View name has a prefix of “mainetti” for identification purpose.
4. Data will be in the form of Json only.
5. Data integrity checks are complete. (Character limitations across systems, data type of attributes)

**Constraints:**

1. Addition/removal of fields to the view would require a rebuild of the shadow table and will be considered a change request.
2. Centric cannot massage the outbound data. (trimming of characters, applying separators like pipeline etc)

**Best Practices:**

1. Rest api call to Centric is not frequent. (4-hour frequency is suggested)
2. Always have a service account for integration purpose.

**Agreed Process:**

1. One time data pull from Centric through middleware. (may be during cut over)
2. Post cut over, at a regular frequency, hit the rest api and fetch the data.



**Details:**

The below steps help to fetch SUPPLIER data via rest api on an **incremental** basis. (only the changed data from the previous data pull)

NOTE: Create a view called “mainetti\_supplier\_view” in the supplier BO.

This will be created initially and will be for admin only.

**Step 1:**

**Getting connected** (screen shot as per postman app)

This would be the first api call to get connected to Centric.

**Rest API (POST call)**

https://mainetti-test.centricsoftware.com/csi-requesthandler/api/v2/session

A screenshot of a computer

Description automatically generated

**Step 2:**

**Calculate the incremental results of the custom view, updating your result node (GET Request)**

https://mainetti-test.centricsoftware.com/csi-requesthandler/api/v2/defects?skip=0&limit=10000&rest\_incremental=vw\_mainetti\_supplier\_view\_incremental

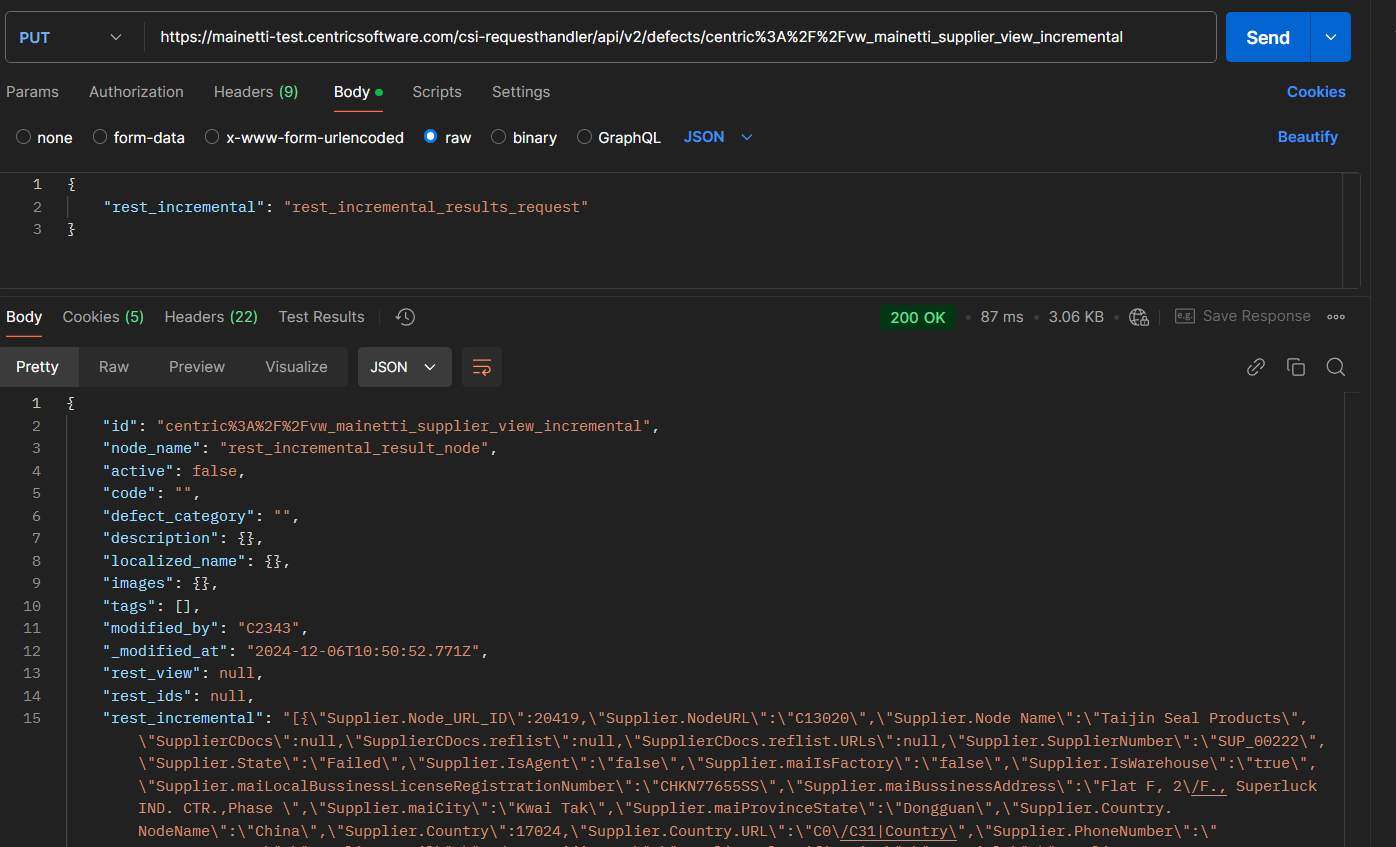
A screenshot of a computer

Description automatically generated

**Step 3:**

**Request the latest result node: (PUT request)**

<https://mainetti-test.centricsoftware.com/csi-requesthandler/api/v2>



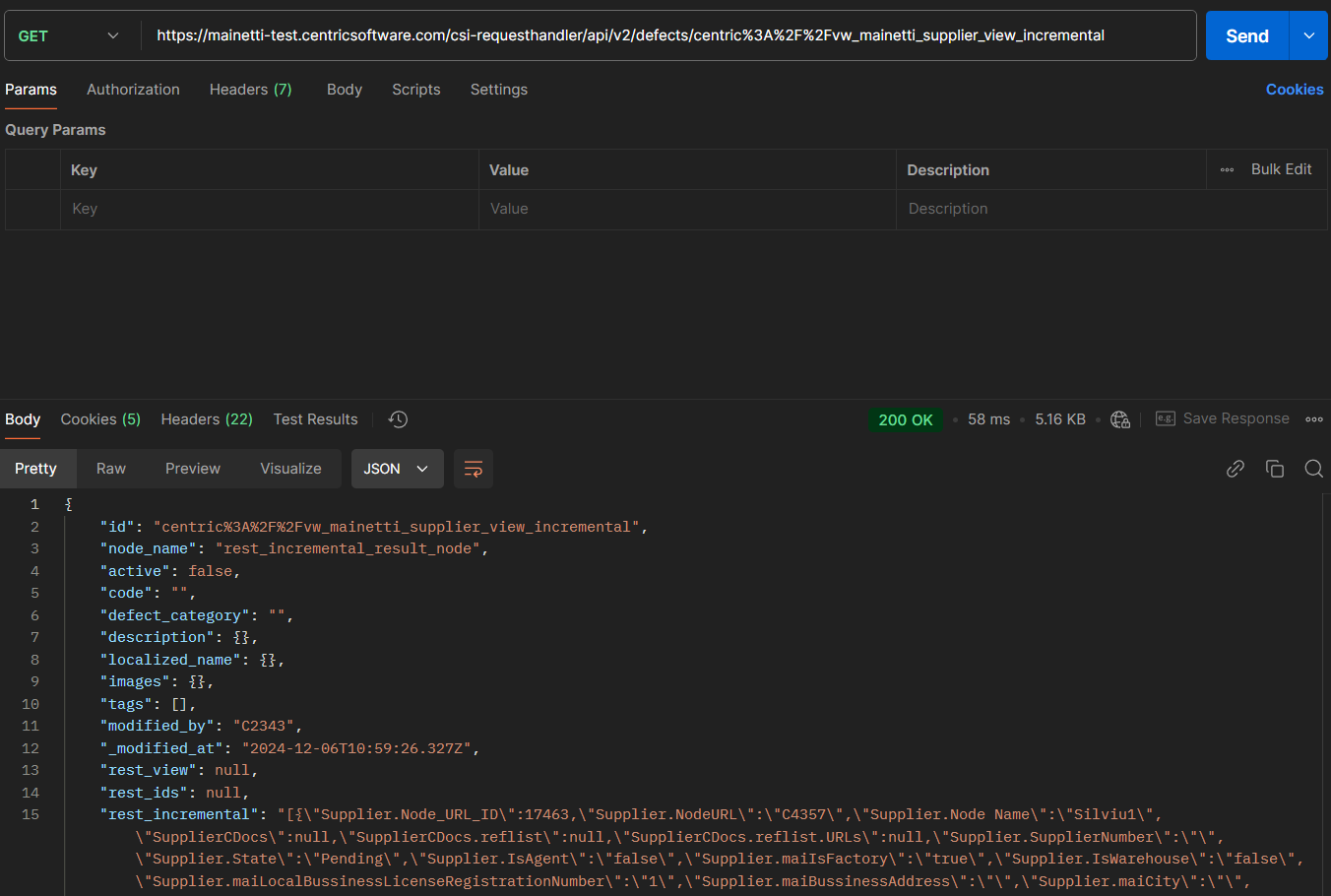
**Step 4:**

**Retrieve the result node from the database (GET Request)**

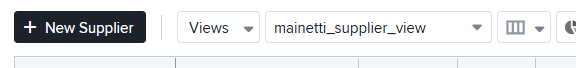
Fire a “GET” request to the Centric server by using the below:

Example:

<https://mainetti-test.centricsoftware.com/csi-requesthandler/api/v2/defects/centric%3A%2F%2Fvw_mainetti_supplier_view_incremental>



The part of the above api highlighted in yellow “mainetti\_supplier\_view” is the name of the custom view on the UI, like what is displayed below.



Ideally, the result obtained is as below: (representation purpose only)

{

    "id": "centric%3A%2F%2Fvw\_mainetti\_supplier\_view\_incremental",

    "node\_name": "rest\_incremental\_result\_node",

    "active": false,

    "code": "",

    "defect\_category": "",

    "description": {},

    "localized\_name": {},

    "images": {},

    "tags": [],

    "modified\_by": "C2343",

    "\_modified\_at": "2024-11-26T13:08:32.082Z",

    "rest\_view": null,

    "rest\_ids": null,

    "rest\_incremental": "[{\"Supplier.Node\_URL\_ID\":20438,\"Supplier.NodeURL\":\"C13236\",\"Supplier.Node Name\":\"Thimm Packaging\",\"SupplierCDocs\":\"\",\"SupplierCDocs.reflist\":\"20597\",\"SupplierCDocs.reflist.URLs\":\"C14013\",\"Supplier.SupplierNumber\":\"SUP\_03001\",\"Supplier.State\":\"Active\",\"Supplier.IsAgent\":\"true\",\"Supplier.maiIsFactory\":\"true\",\"Supplier.IsWarehouse\":\"true\",\"Supplier.maiLocalBussinessLicenseRegistrationNumber\":\"RO1092660\",\"Supplier.maiBussinessAddress\":\"STR. SURA MICII NR. 13\",\"Supplier.maiCity\":\"Sura Mica\",\"Supplier.maiProvinceState\":\"Sibiu\",\"Supplier.Country.NodeName\":\"Romania\",\"Supplier.Country\":17099,\"Supplier.Country.URL\":\"C0\/C136|Country\",\"Supplier.PhoneNumber\":\"+400754666888\",\"Supplier.Email\":\"Vanzari@thimm.ro\",\"Supplier.Classification\":\"Materials\",\"Supplier.maiClassificationSubType\":\"Packaging\",\"Supplier.maiCurrency.NodeName\":\"RON\",\"Supplier.maiCurrency\":17296,\"Supplier.maiCurrency.URL\":\"C4155\",\"Supplier.maiVAT\":1.900000000000000e-001,\"Supplier.maiDefaultPaymentTerm\":\"Open Account\",\"Supplier.TradeTerm\":\"CPT\",\"Supplier.maiContactPersonAgent\":\"Dorin Micu\",\"Supplier.maiBankKey\":\"JJJSSLYH88\",\"Supplier.maiBankaccountnumber\":\"RO90BRDE6667700BBERE\",\"Supplier.maiExpiryDateBussinessLicense\":null,\"Supplier.maiExpiryDateBussinessLicense UTC\":null},{\"Supplier.Node\_URL\_ID\":21232,\"Supplier.NodeURL\":\"C14936\",\"Supplier.Node Name\":\"Example\",\"SupplierCDocs\":null,\"SupplierCDocs.reflist\":null,\"SupplierCDocs.reflist.URLs\":null,\"Supplier.SupplierNumber\":\"\",\"Supplier.State\":\"Pending\",\"Supplier.IsAgent\":\"true\",\"Supplier.maiIsFactory\":\"false\",\"Supplier.IsWarehouse\":\"false\",\"Supplier.maiLocalBussinessLicenseRegistrationNumber\":\"\",\"Supplier.maiBussinessAddress\":\"\",\"Supplier.maiCity\":\"\",\"Supplier.maiProvinceState\":\"\",\"Supplier.Country.NodeName\":null,\"Supplier.Country\":1,\"Supplier.Country.URL\":\"centric:\",\"Supplier.PhoneNumber\":\"\",\"Supplier.Email\":\"\",\"Supplier.Classification\":\"\",\"Supplier.maiClassificationSubType\":\"\",\"Supplier.maiCurrency.NodeName\":null,\"Supplier.maiCurrency\":1,\"Supplier.maiCurrency.URL\":\"centric:\",\"Supplier.maiVAT\":0.000000000000000e+000,\"Supplier.maiDefaultPaymentTerm\":\"\",\"Supplier.TradeTerm\":\"\",\"Supplier.maiContactPersonAgent\":\"\",\"Supplier.maiBankKey\":\"\",\"Supplier.maiBankaccountnumber\":\"\",\"Supplier.maiExpiryDateBussinessLicense\":null,\"Supplier.maiExpiryDateBussinessLicense UTC\":null}]"

}

Fetch the “rest\_incremental” value from the above data set which is the actual data.

Note that the data is in the form of a json.

|  |  |  |
| --- | --- | --- |
| Data request scenarios | Process | Comments |
| First Time Request | Carry out Step 1, 2,3,4 | You need to get connected. |
| After the first-time request | Carry out Step2,3,4 | No need to get connected, assuming connection is still active |
| Refresh of shadow table | Carry out the step mentioned below and then Step 2,3,4 | When there is addition of new attribute, deletion of attribute from the views |

Note that considering there is a 4-hour frequency in the data pull from centric using rest api, then there would be 19 rest api calls in a day.

|  |  |
| --- | --- |
| First call | 4 api calls |
| Second, Third, Fourth, Fifth, Sixth call | 3 api calls every time |
|  | Total: 19 calls |

**Shadow table refresh:**

Fire a PUT api call with the parameter: rest\_incremental:”deploy” and carry out Step 2,3,4

A screenshot of a computer

Description automatically generated

**FULL DATA ACCESS:**

Fire the below api as a GET request.

<https://mainetti-test.centricsoftware.com/csi-requesthandler/api/v2/defects?skip=0&limit=10000&rest_view=vw_mainetti_supplier_view>

The above api will fetch all the records in the supplier node.