Ganpat University
॥ विद्यया समाजोत्कर्षः ॥

Institute of Computer Technology

Name: Tushar Panchal

En.No: 21162101014

Sub: EADC (Enterprise Application Development for Cloud)

Branch: CBA

Batch:61

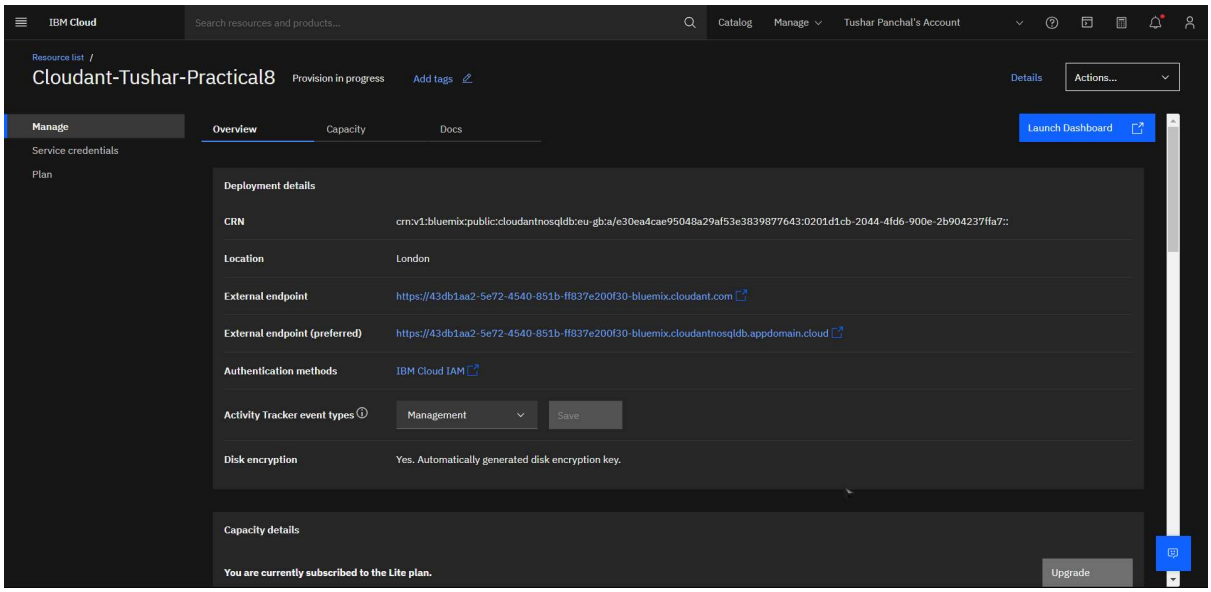
----------PRACTICAL 08----------

❖ Question :

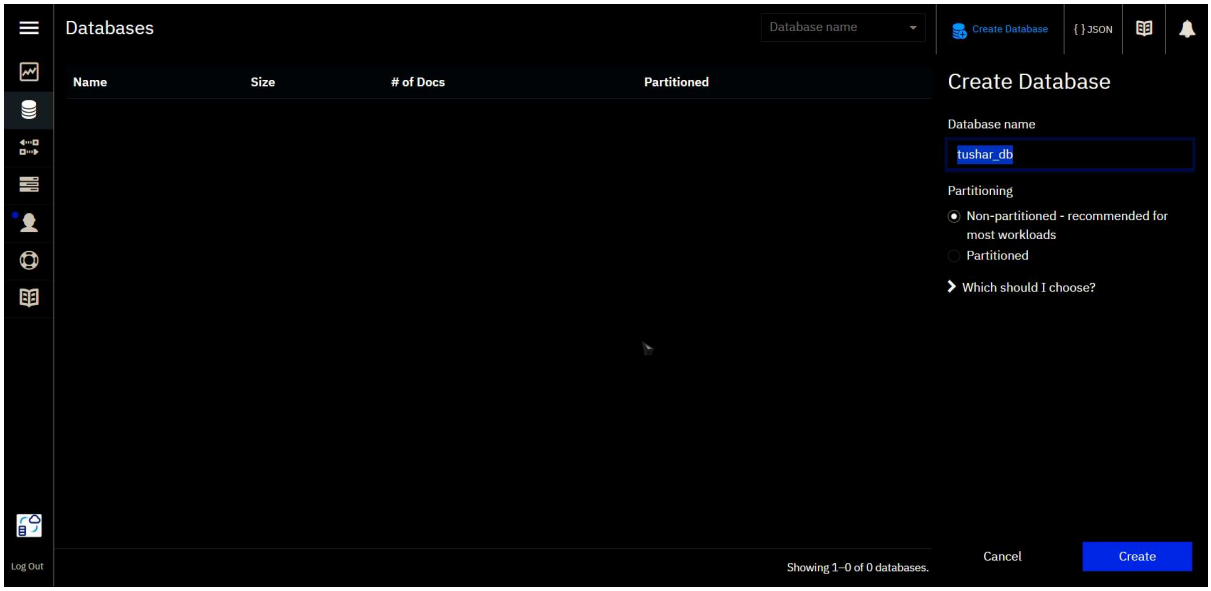
Cloudant database
in IBM Cloud integration with a third-party application (Postman)

1. Install third party application postman on
your system
2. Create database for novel shop and document
via creating an instance of Cloudant database.
4. Perform crud operations on database created using API's
5. Make PDF document for performed tasks

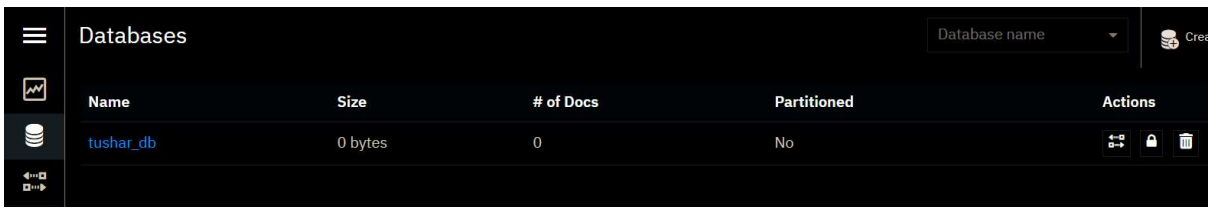
» **Step 1 :** navigate cloudant and Create cloudant service I created by name Cloudant-Tushar-Practical8 and after created lunch dashboard.



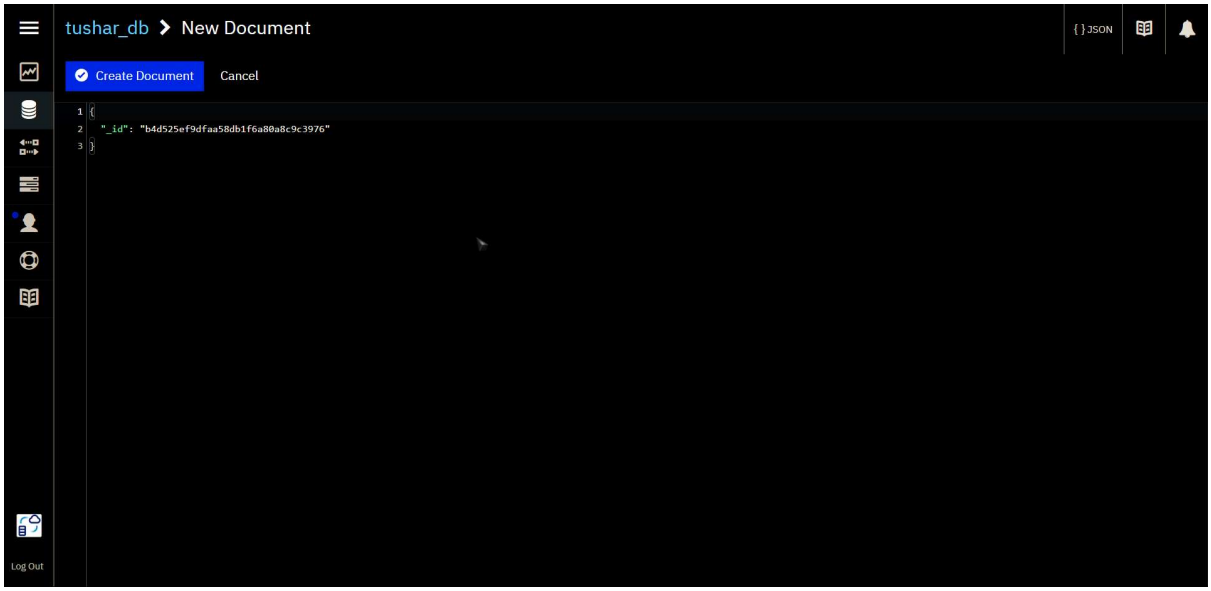
» **Step 2 :** after launch dashboard click on create database and give it name and set non-partitioned and create.



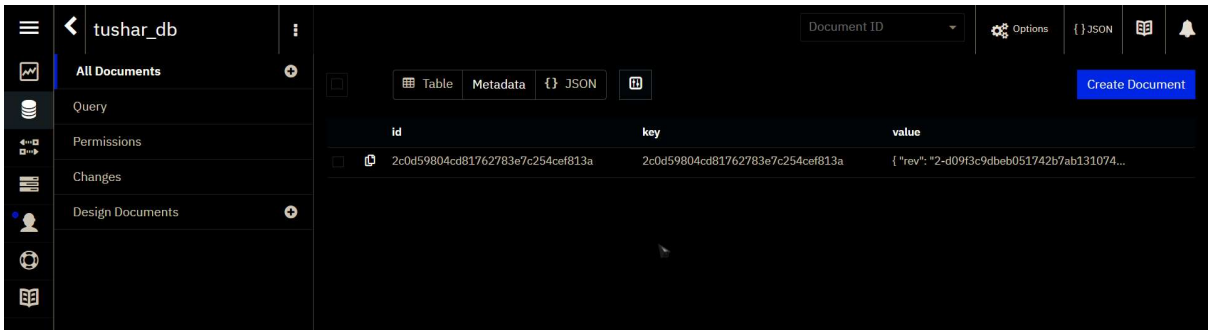
» **Step 3 :** Database created now click on database.



Now go to design documents and create Document

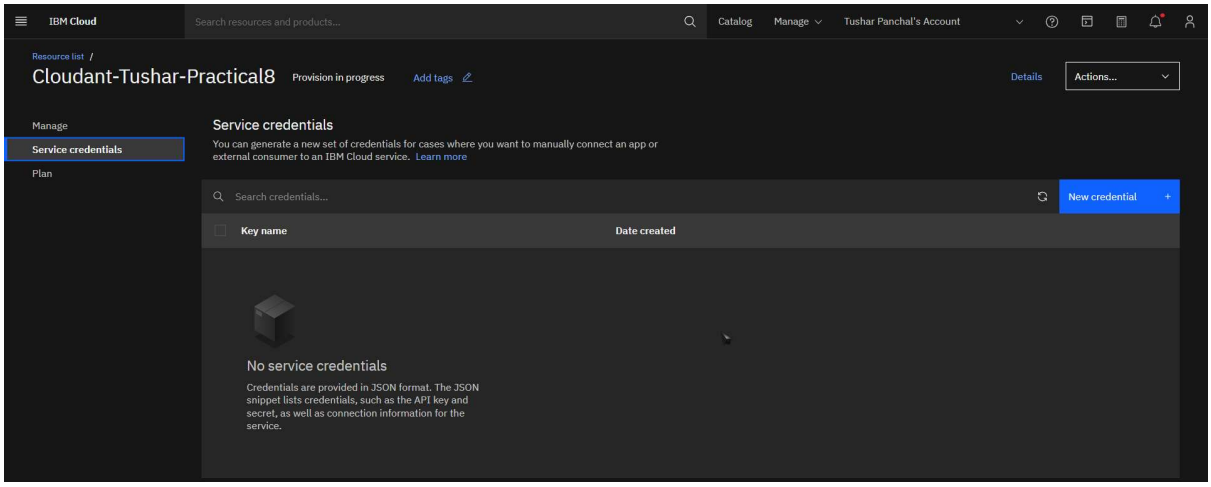


» **Step 4 :** Click on id that shown below and open it.

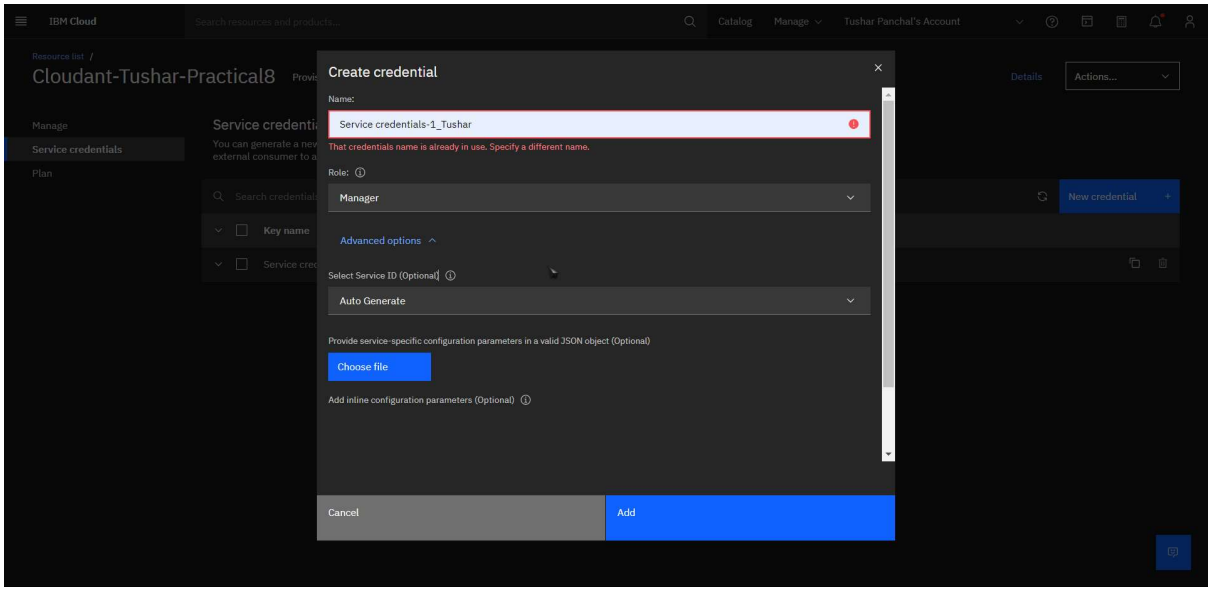


» **Step 5 :** Add name and age and save it.

» **Step 6 :** Go into service credentials and hit new credential.



» **Step 7 :** Give it name add role manager and set id auto generate and hit add button.

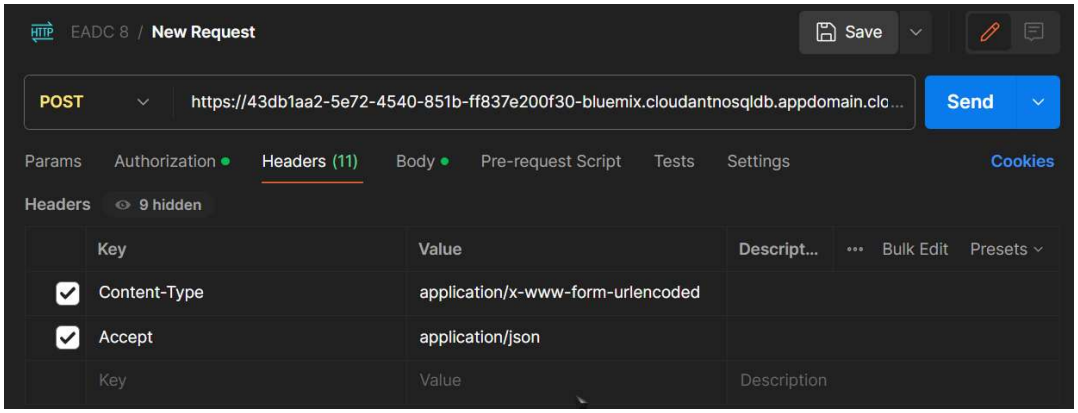


As you can see credential has been added successfully

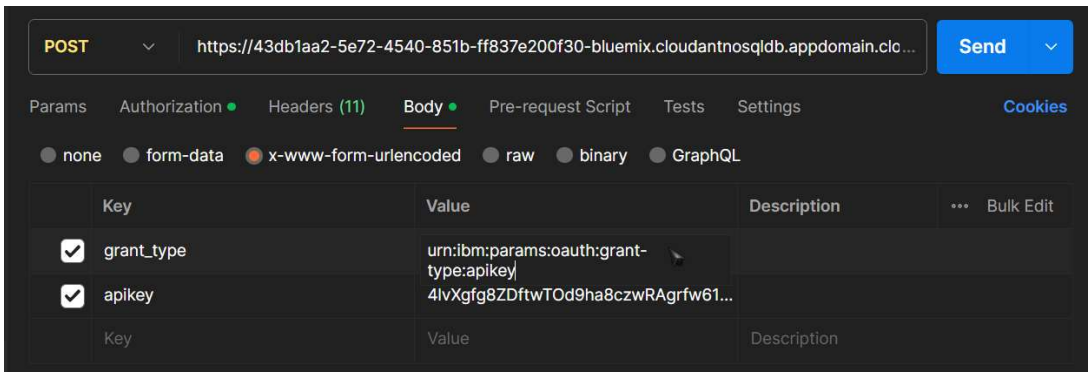
```
{
  "apikey": "4lvXgfg8ZDftwTOd9ha8czwRAgrfw615H1HcYt5GjpzB",
  "host": "43db1aa2-5e72-4540-851b-ff837e200f30-
bluemix.cloudantnosqldb.appdomain.cloud",
  "iam_apikey_description": "Auto-generated for key
crn:v1:bluemix:public:cloudantnosqldb:eu-
gb:a/e30ea4cae95048a29af53e3839877643:0201d1cb-2044-4fd6-900e-
2b904237ffa7:resource-key:69d7e3db-488c-4a5c-b99e-7162dcc169c6",
  "iam_apikey_id": "ApiKey-73ed4c7e-8f20-492a-9e3c-4e7622de41b2",
  "iam_apikey_name": "Service credentials-1_Tushar",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-
identity::a/e30ea4cae95048a29af53e3839877643::serviceid:ServiceId-
22f66608-2568-450b-991c-42cb495a55f3",
  "url": "https://43db1aa2-5e72-4540-851b-ff837e200f30-
bluemix.cloudantnosqldb.appdomain.cloud",
  "username": "43db1aa2-5e72-4540-851b-ff837e200f30-bluemix"
}
```


» **Step 8 :** Now to send request from postman we have to generate bearer token .

In header set this key and values:

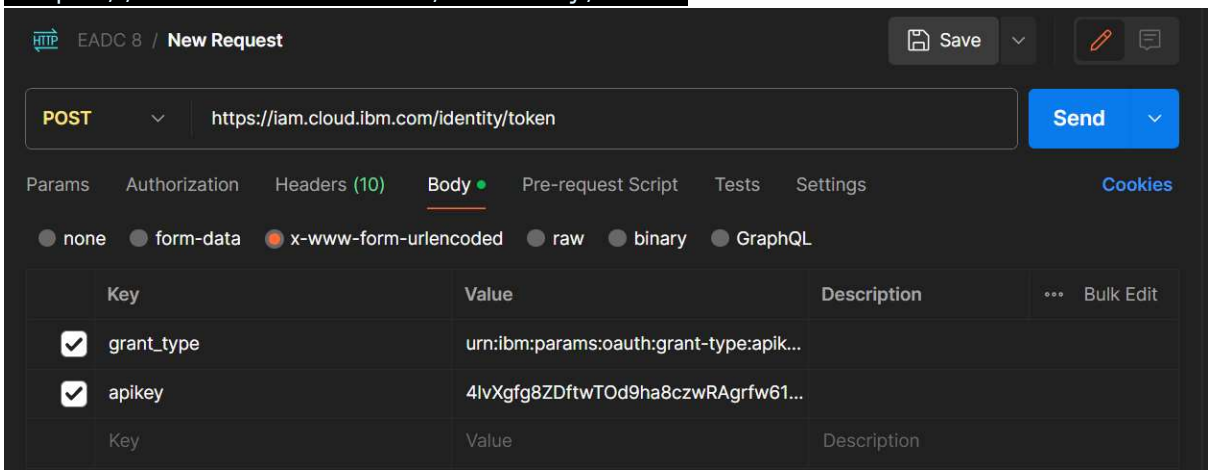


In body part set grant_type as mine but enter your own api key:

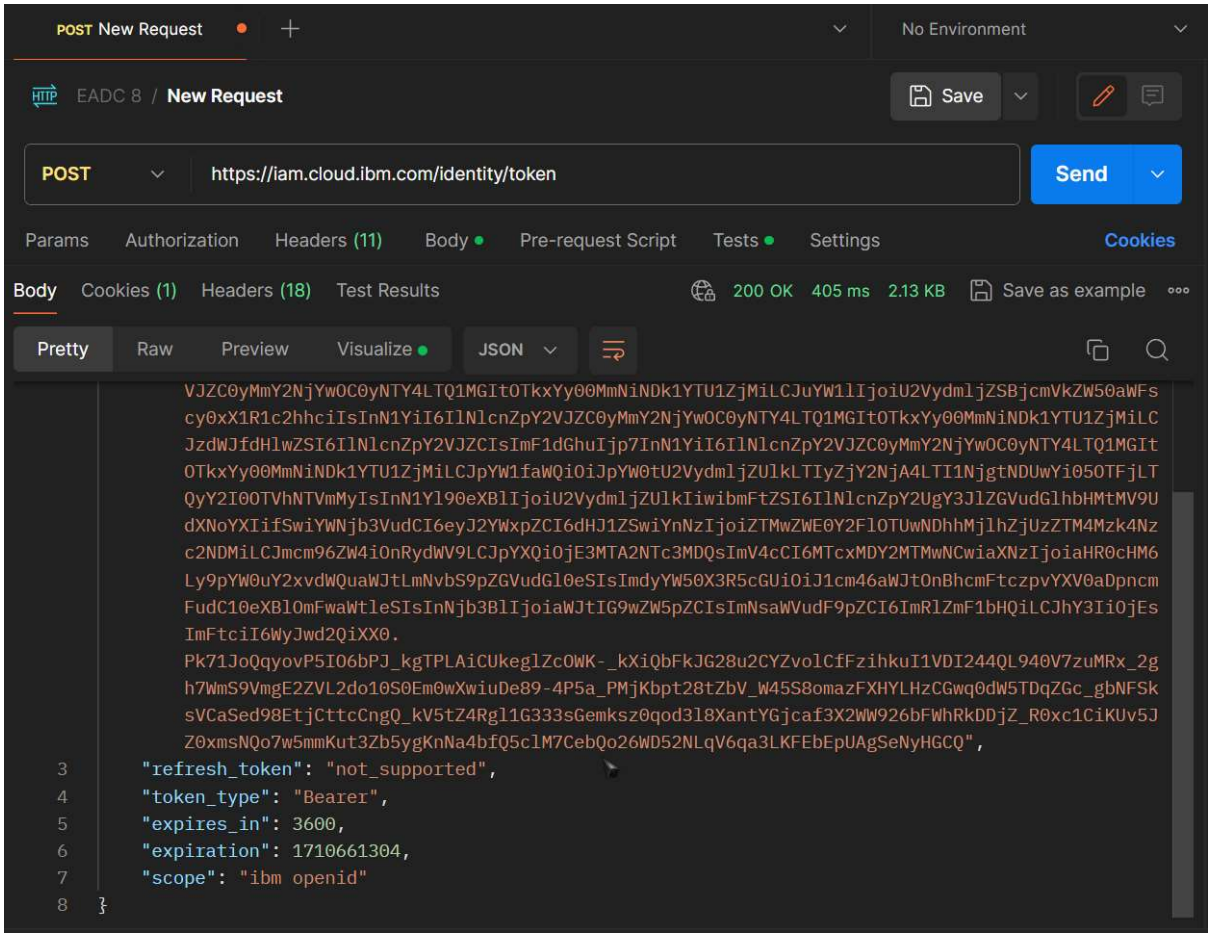


And in url enter this url

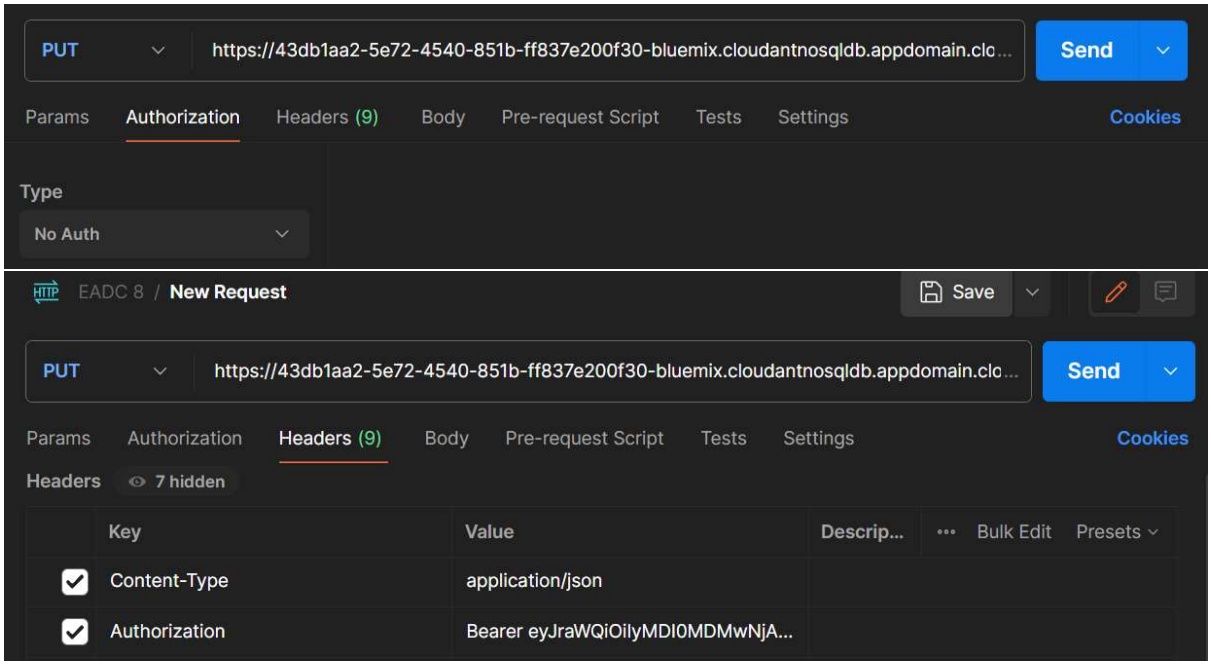
`https://iam.cloud.ibm.com/identity/token`



After send you will receive bearer token :

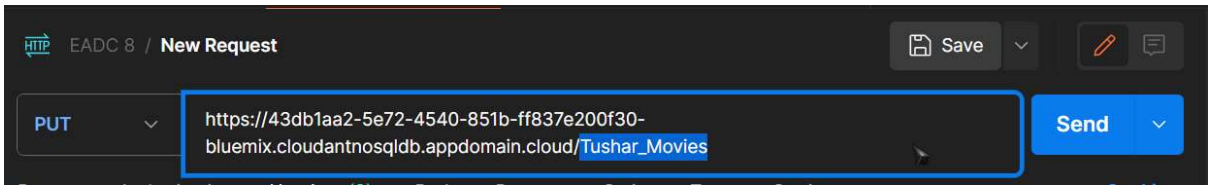


Now to send any request we have to configure headers set content type as json and in authorization enter that bearer access token that we have been created just before as shown in up and enter url from service credentials that we added and in authorization select no authorization:

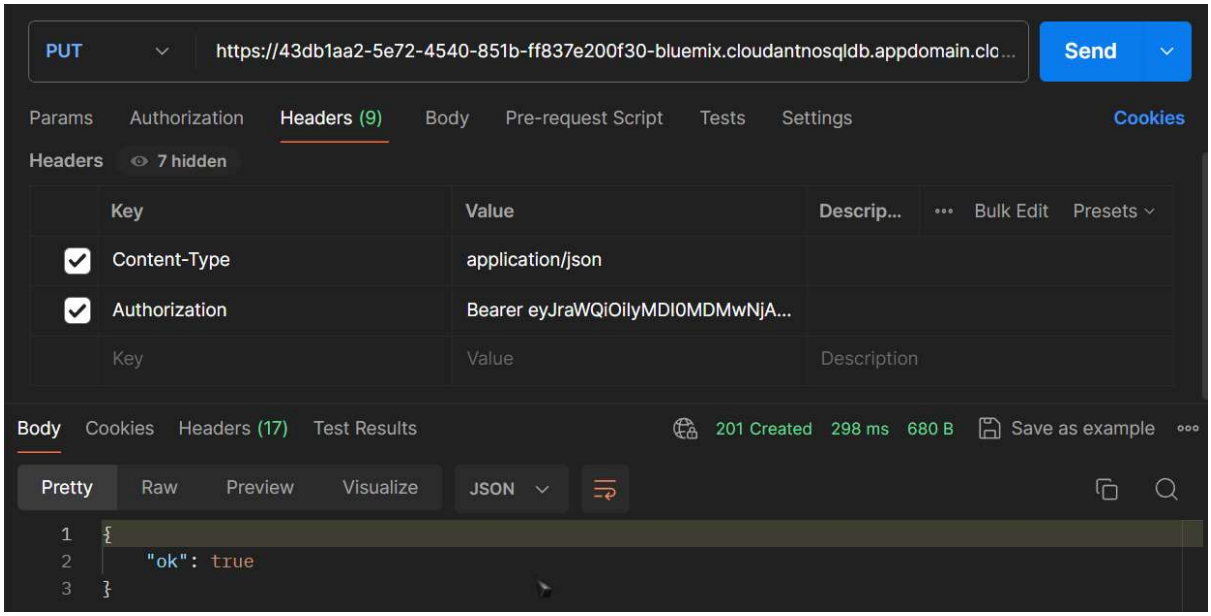


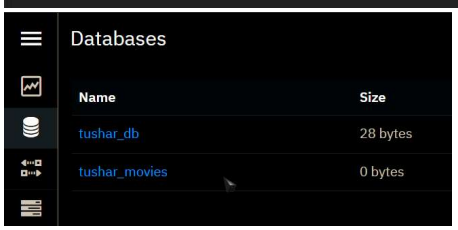
» **Step 9 :** set put request and create new database .

Name your Database at the end of endpoint url like these :

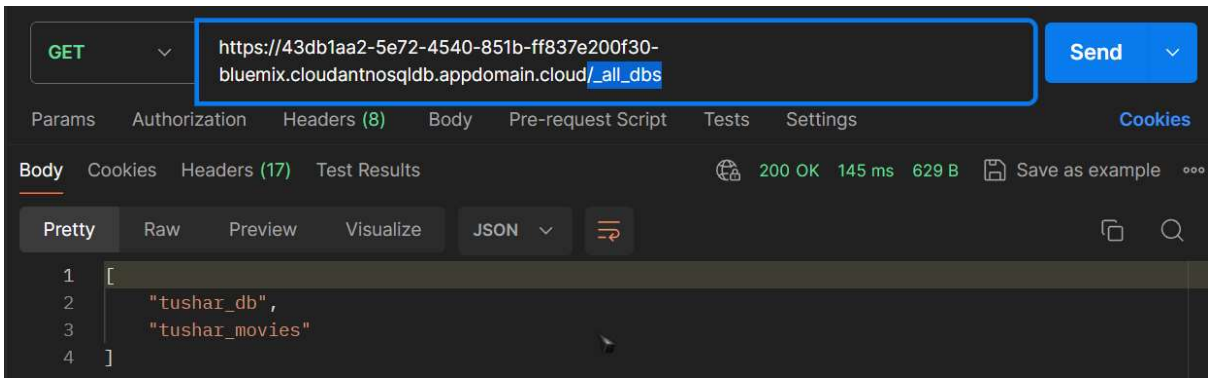


As you can see it is created showing true in output :

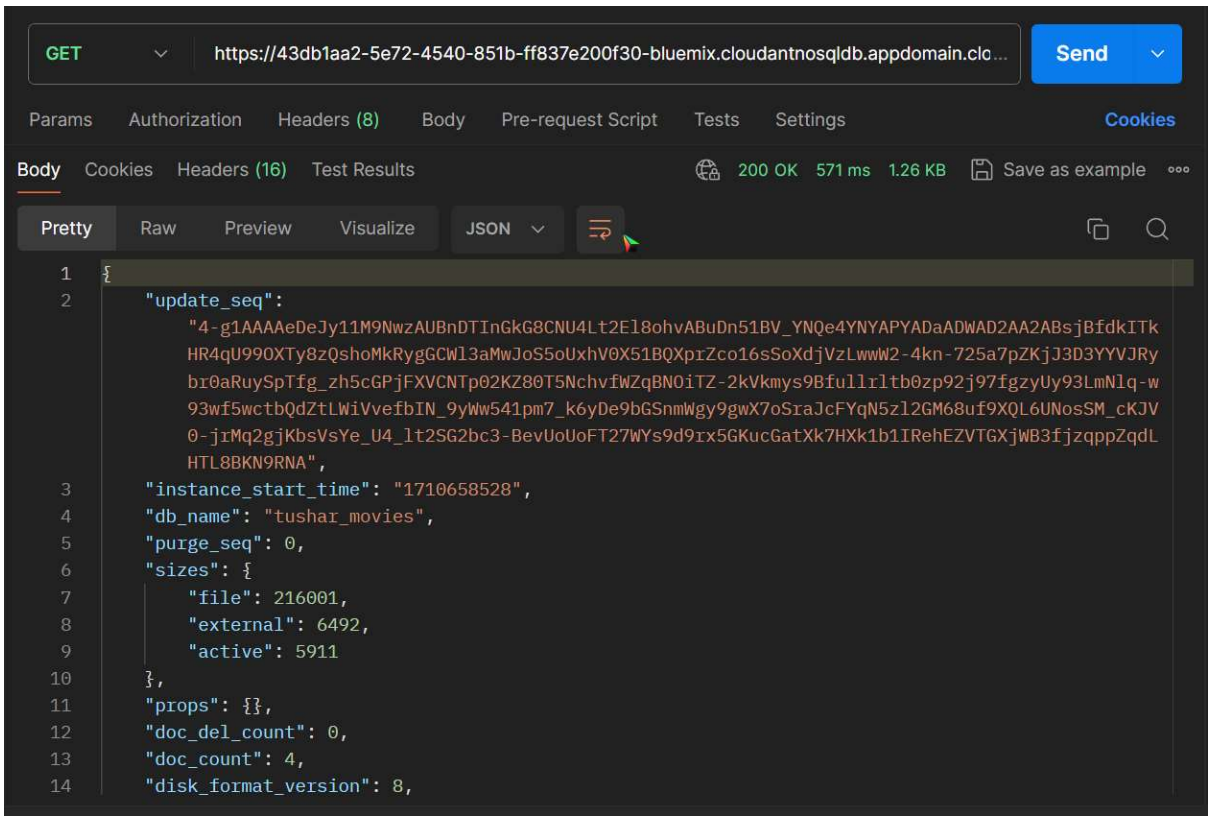




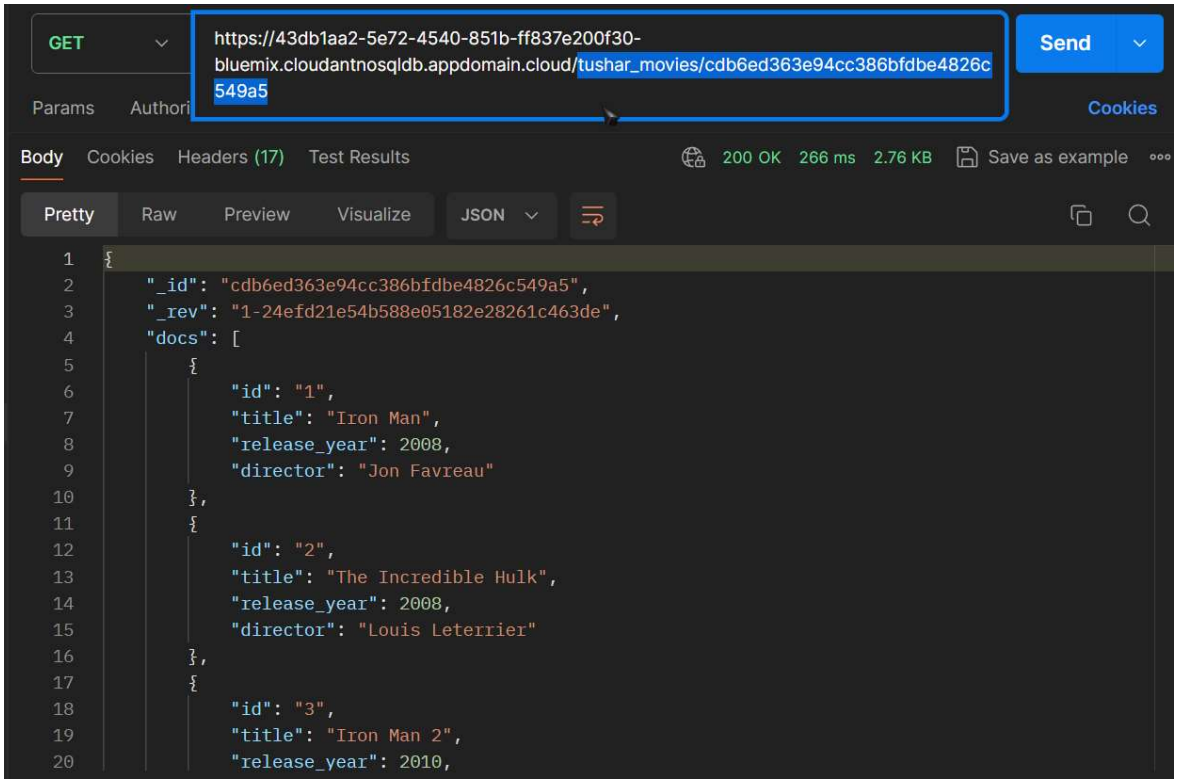
» **Step 10 :** to check all database available in use **“all_dbs”** at the end of endpoint and set method as **get**.



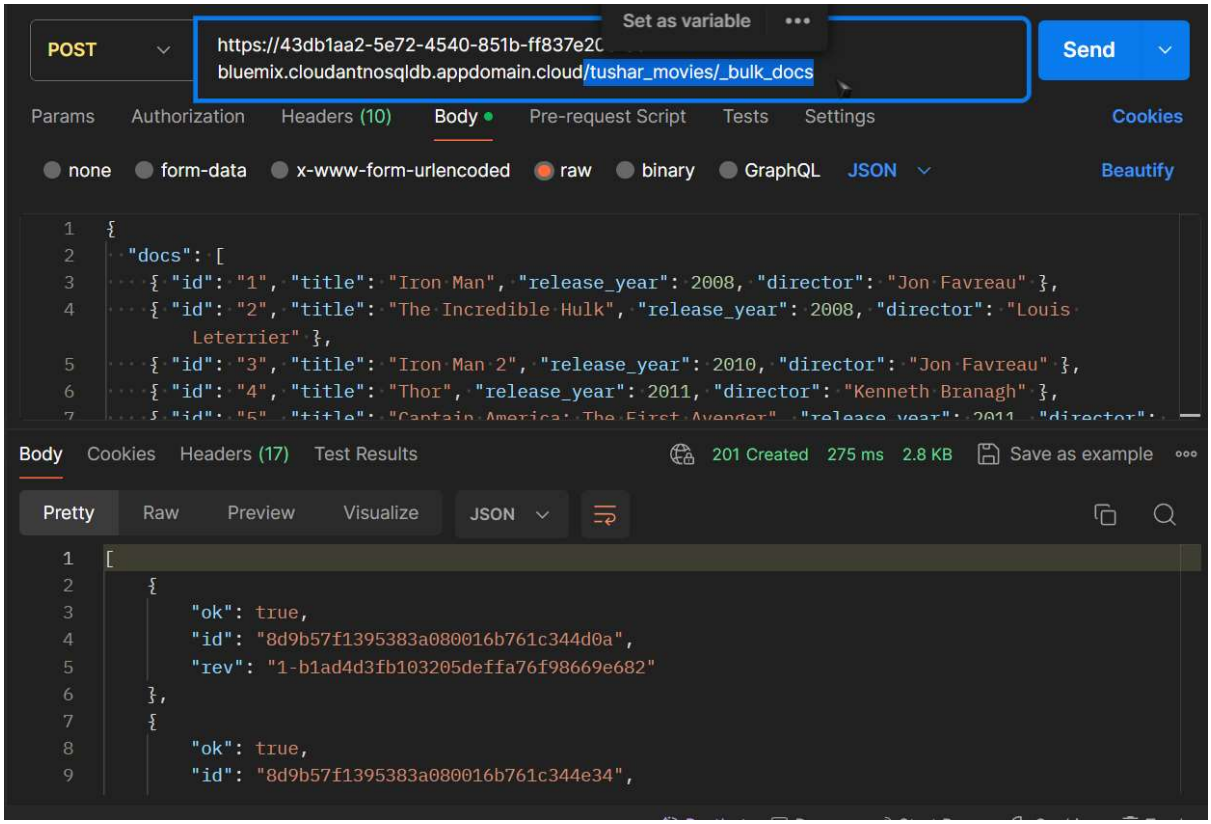
To get all details of database add database name at the end of endpoint :



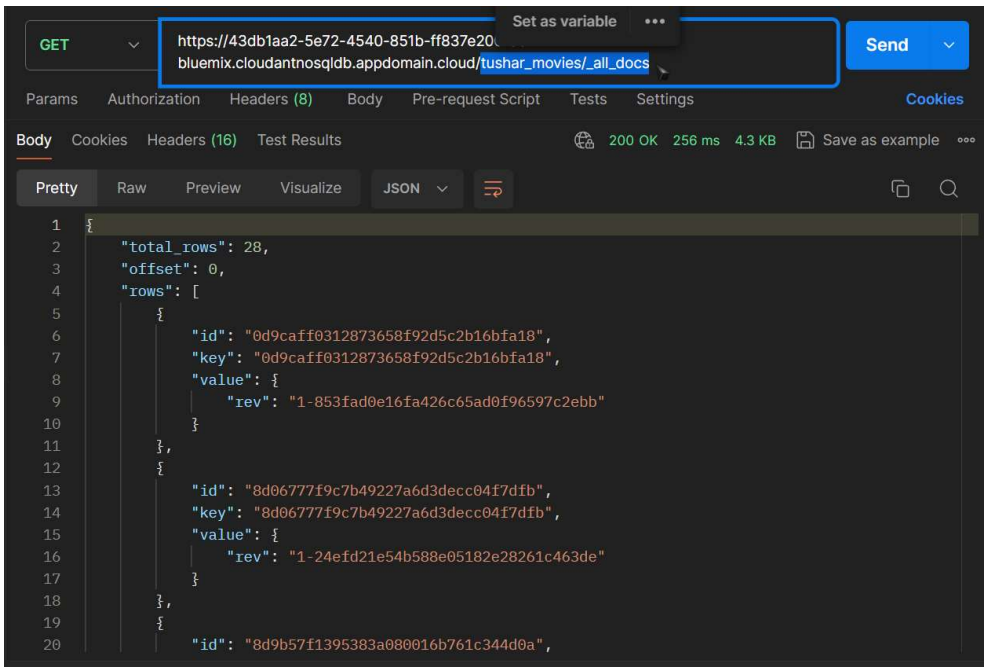
» **Step 11 :** To get single data information add database name and id of document at the end point.



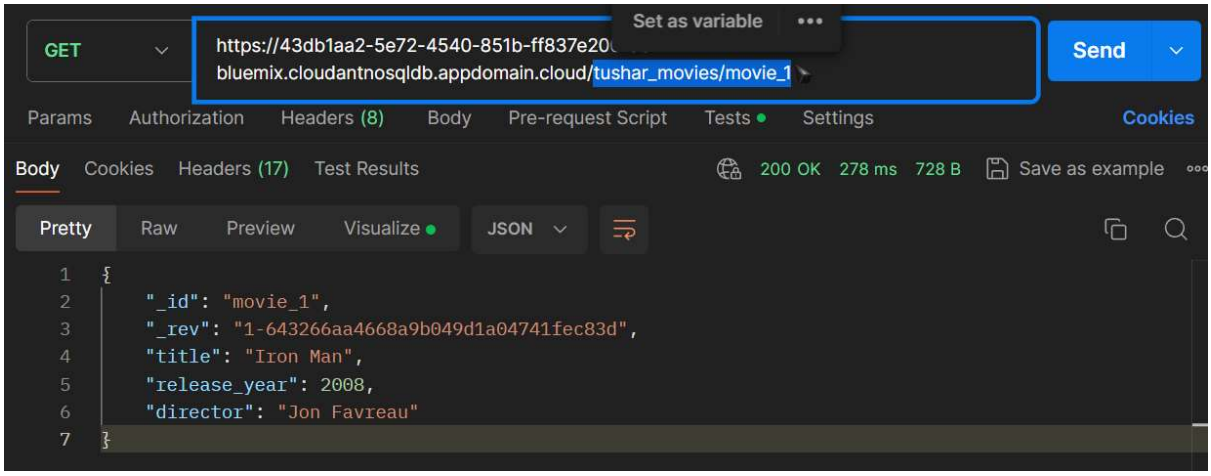
» **Step 12 :** To get data in bulk set **post** method and add db name and this **"_bulk_docs"** at the endpoint.

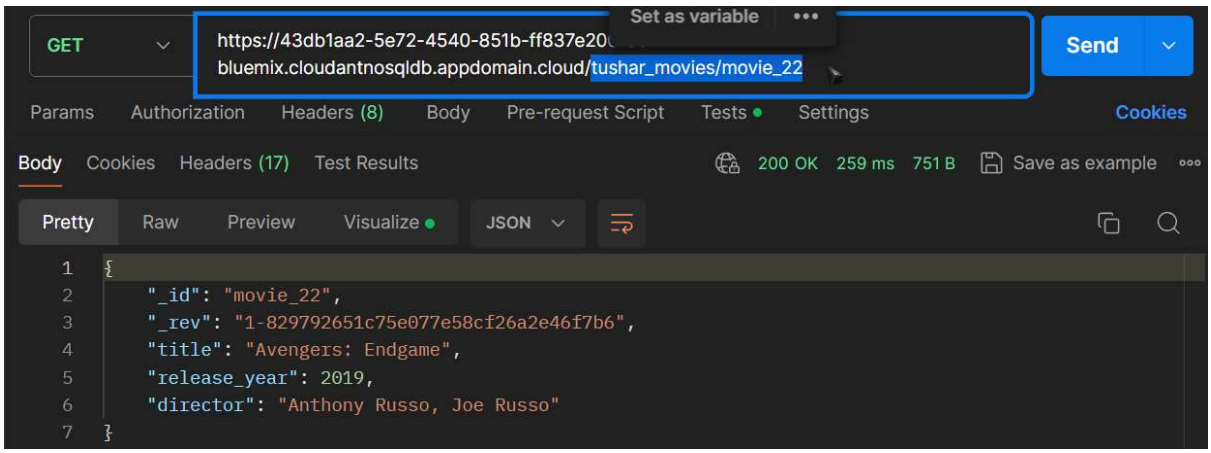


» **Step 13 :** To get all docs id & key set **get** method and add db name and this **"_all_docs"** at the endpoint.



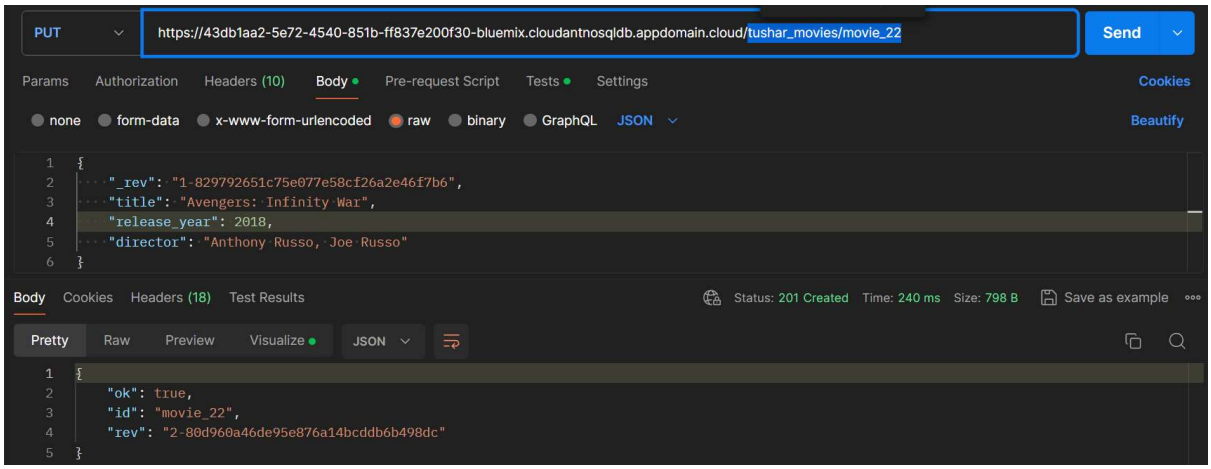
» **Step 14 :** To get any particular data set **get** method and add db name and document id at the endpoint.

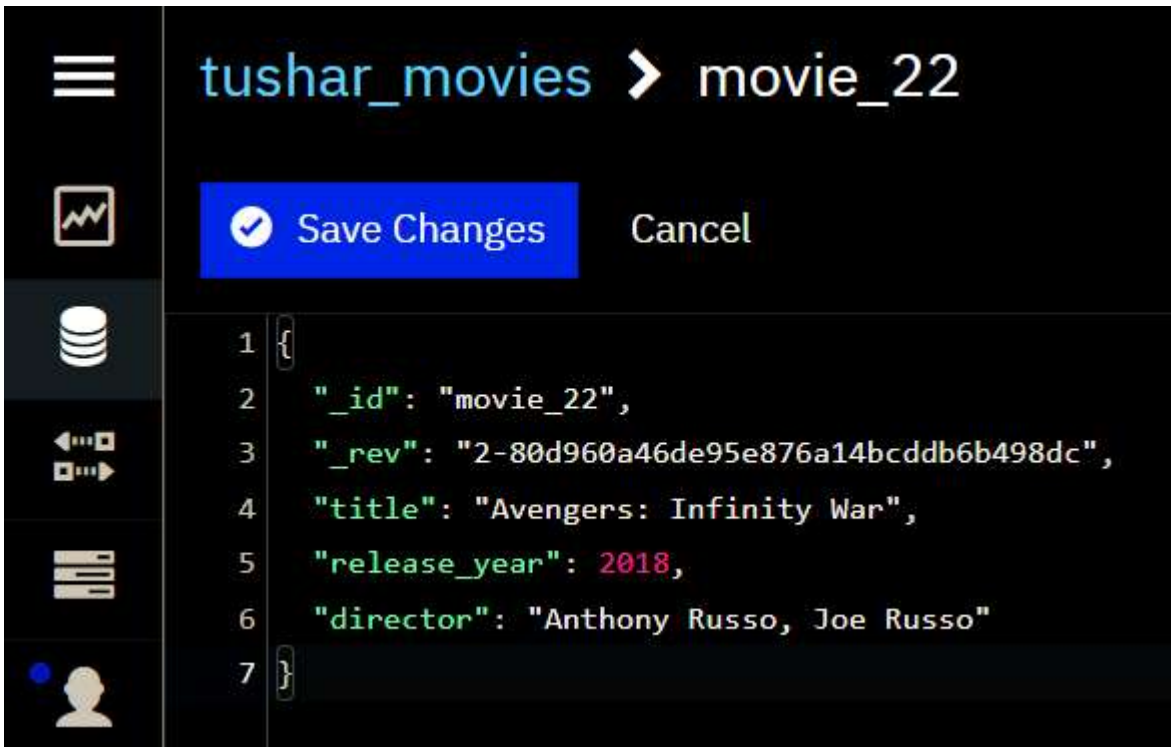




» **Step 15 :** Updating the data set put method here I updating title and year so I adding raw json for that.

add db name and document id at the endpoint

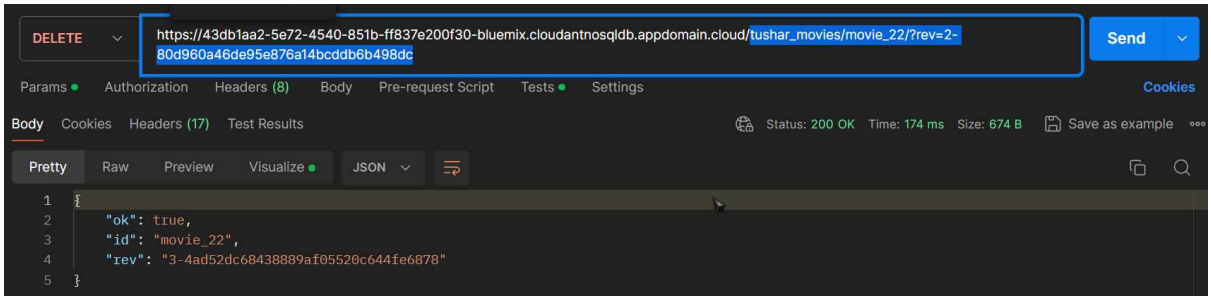




data has been updated successfully.

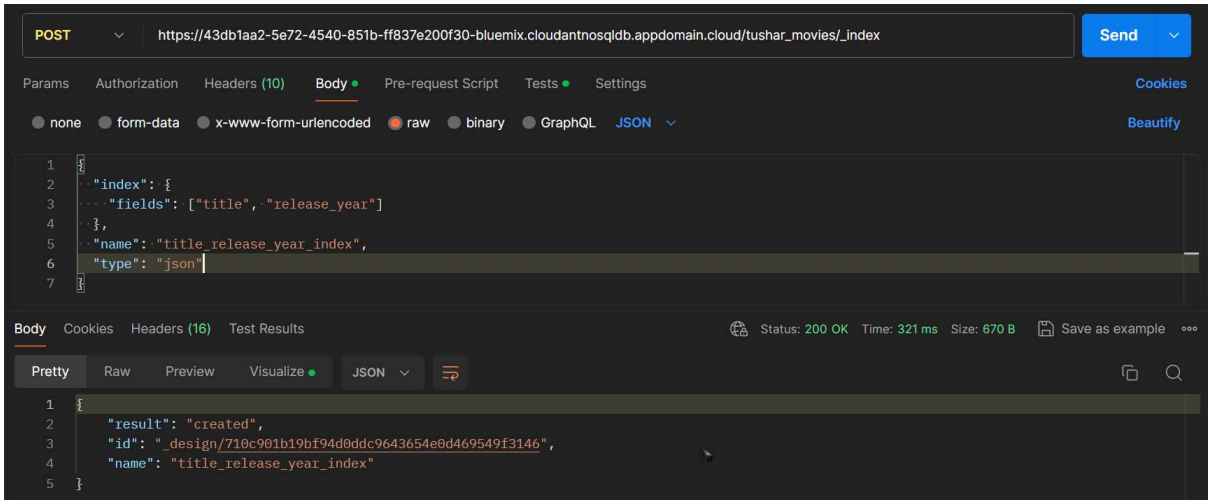
» **Step 16 :** To delete the the data that we created use this command :- DELETE request for ***$URL/$DATABASE/$DOCUMENT_ID?rev=$REV***

Add database name and document id and rev id at the end of endpoint and select DELETE method



» **Step 17 :** Create new index using send a POST request to **$URL/$DATABASE/_index** Go in to body and add data and send add below data in body , row , json

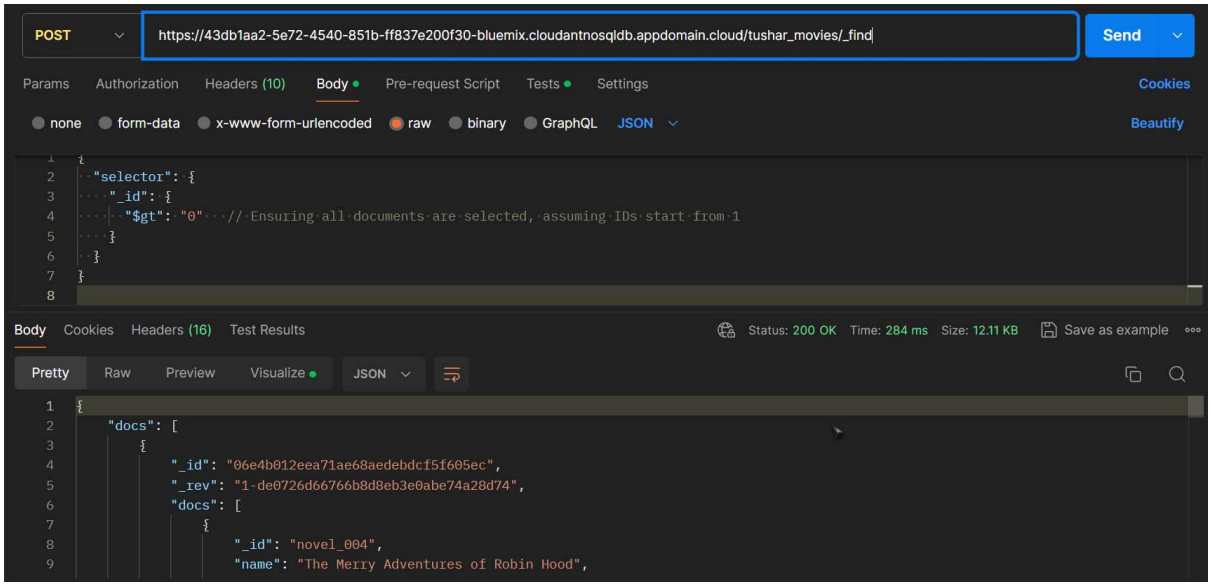
```
{
  "index": {
    "fields": ["title", "release_year"]
  },
  "name": "title_release_year_index",
  "type": "json"
}
```



» Step 18 : POST request to $URL/$DATABASE/_find

```
{
  "selector": {
    "_id": {
      "$gt": "0"    // Ensuring all documents are selected, assuming IDs start from
1
    }
  }
}
```

Here as you can see we find data where id is greater than 0:



Here as you can see I find all avengers movies :