

Mongodb LoopBack Connector

PavanKumar Meduri

3-2-2017

Api Connect Developer

Miracle Software Systems, Inc.

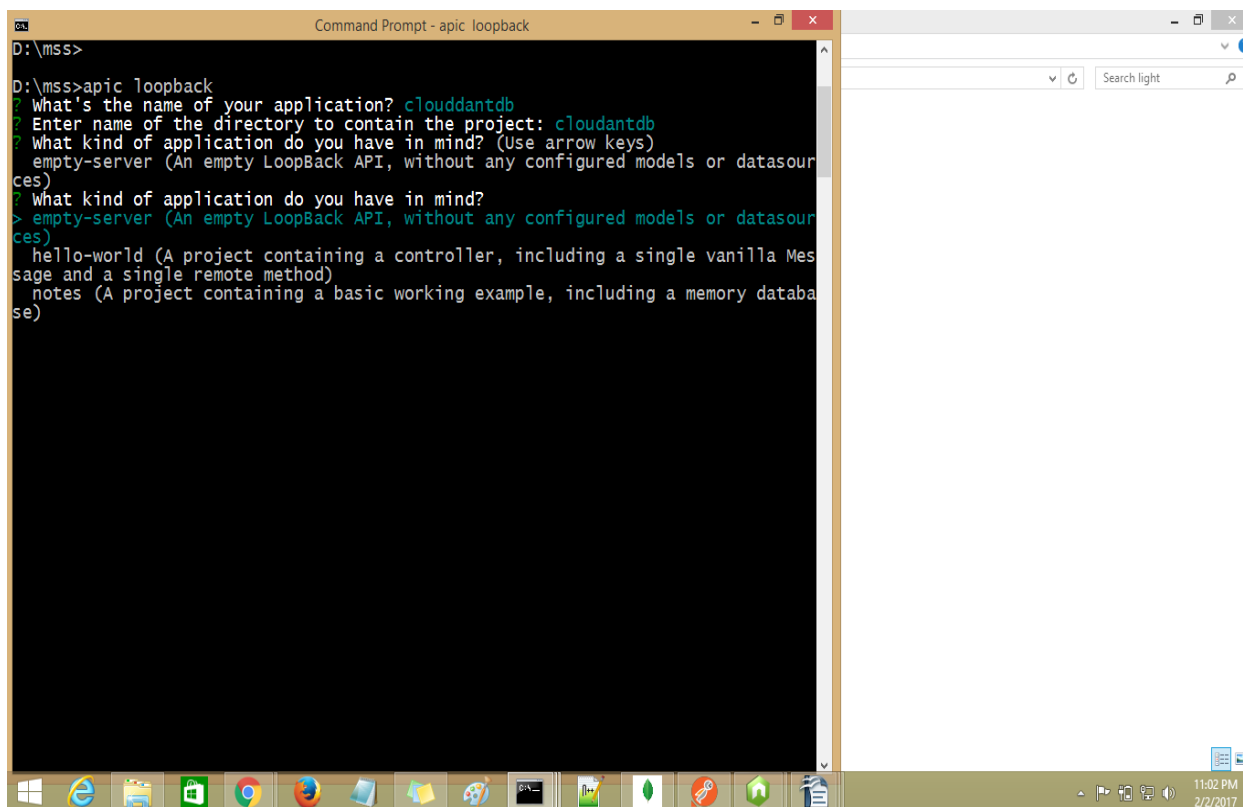
Creating the MongoDB LoopBack Connector

Creating the LoopBack Application

To create the LoopBack Application we need work on CommandLine Interface. In that we can give the command

Eg: apic loopback

Then the command prompt can be shown as



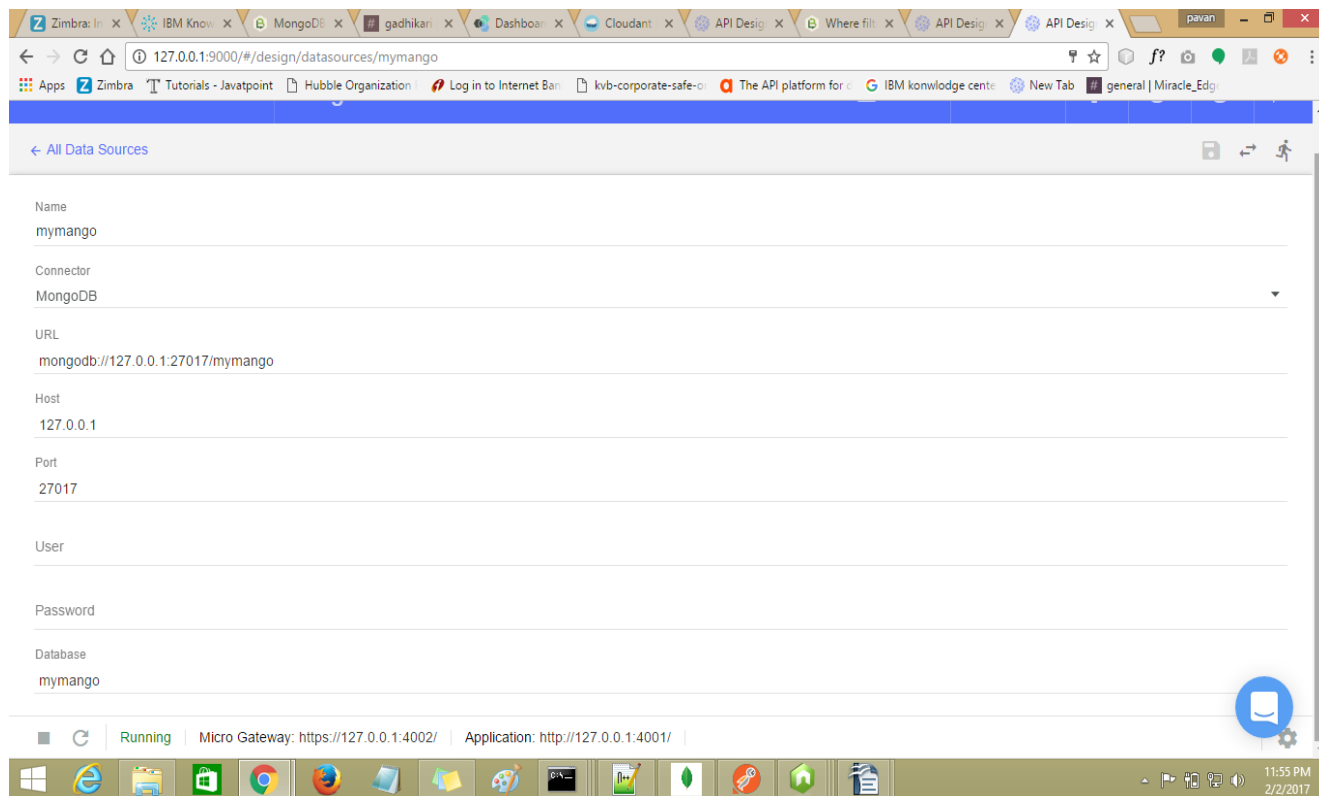
```
D:\mss>apic loopback
? What's the name of your application? cloudantdb
? Enter name of the directory to contain the project: cloudantdb
? What kind of application do you have in mind? (Use arrow keys)
? empty-server (An empty LoopBack API, without any configured models or datasources)
? What kind of application do you have in mind?
> empty-server (An empty LoopBack API, without any configured models or datasources)
? hello-world (A project containing a controller, including a single vanilla Message and a single remote method)
? notes (A project containing a basic working example, including a memory database)
```

In the Command Prompt select the Empty server and then press **ENTER**.

After creation of loopback project then change the directory current directory and then create **DataSource** and **Model** in the Api designer.

Creating the DATASOURCE

In the Api Designer click on DataSources and then press on **+ADD** button.



Then give the Appropriate name for the given fields like

Name :-Describes name of the datasource.

Connector:-Describes the list of loopback connectors.

URL:-Describes the URL is In the form of

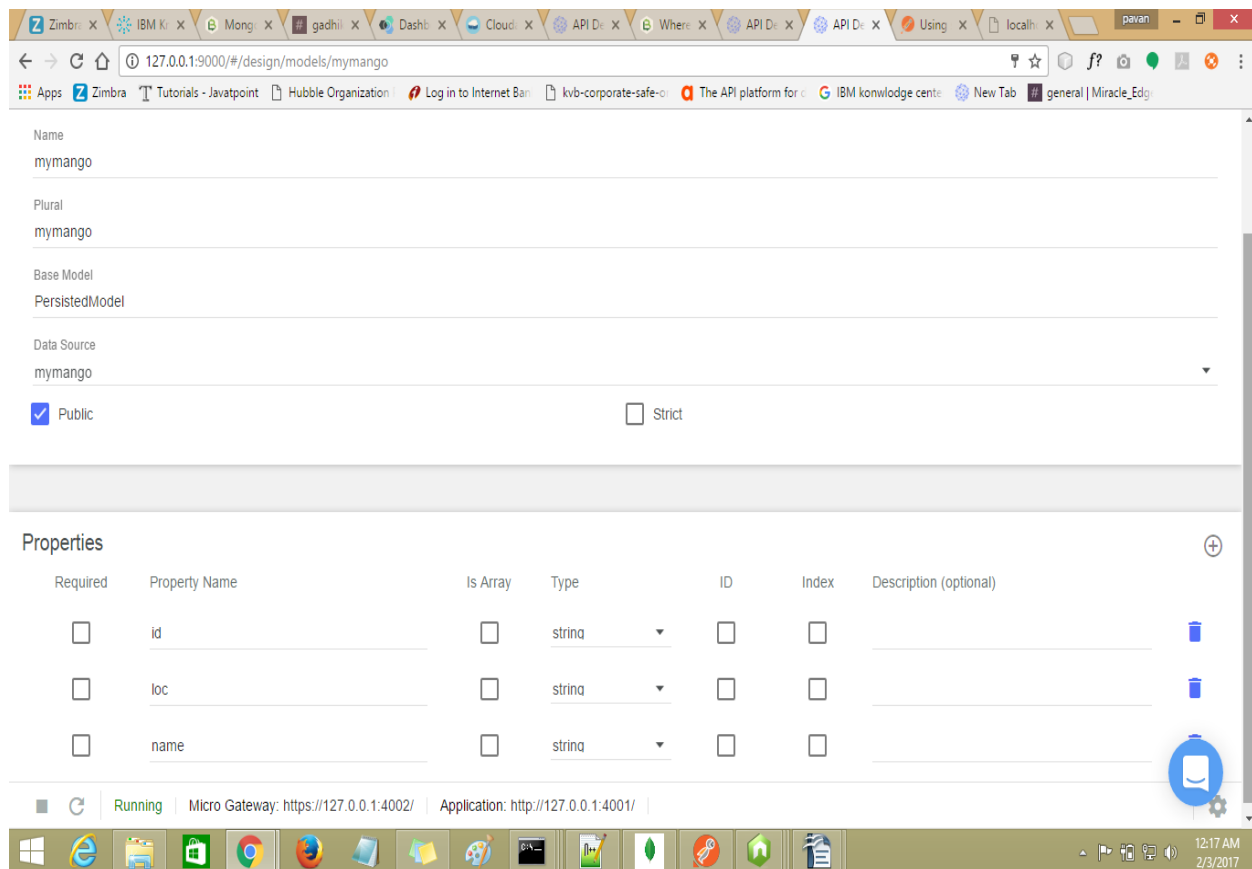
mongodb://username:password@host:port/database

DataBase:-Describes the name of the database.

*Creating the **MODEL***

For creating the model select **MODELS** and then click on **+ADD**.

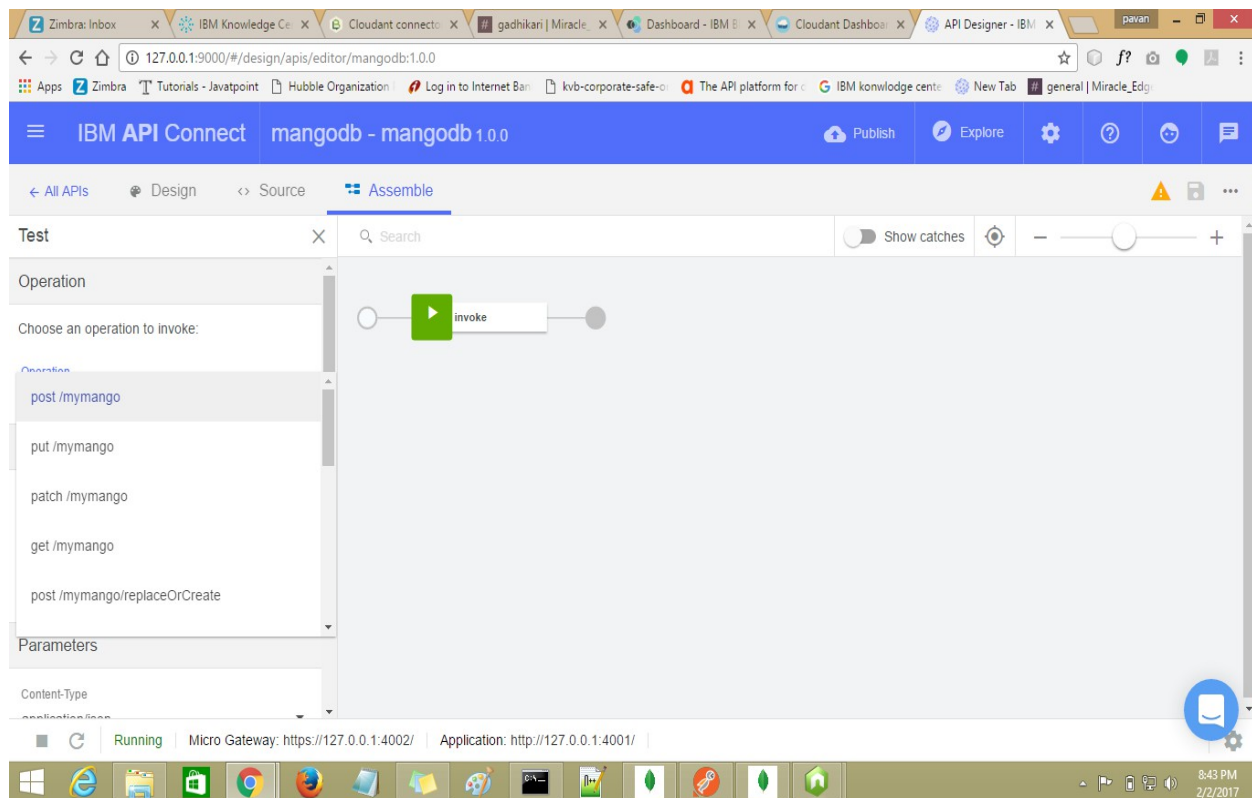
In the UI we need to Add the some properties for that click on + sign,then for each data we need to select the respetive datatype.



After creating the DataSource and model we need to perform the GET,POST,UPDATE,DELETE operation.

POST Operation on MongoDB DataBase

For the Post operation we need to select the Invoke operation from the drop down menu.



Then it will show two options for us.

- 1.Show Schema
- 2.Generate

when we click on the **Show Schema** then it will shows the schema that we are created.

When we are click on the Generate then it will gives the DEFAULT values according to our Schema.

IBM API Connect mangodb - mangodb 1.0.0

Test

Model instance data

```
data
{
  "id": "91",
  "name": "praveen",
  "loc": "guntur"
}
```

Show schema | Generate

☐ Repeat

Repeat the API invocation a set number of times, or until the stop button is clicked

Stop after: ☒ Stop on error

10

Invoke

Running | Micro Gateway: https://127.0.0.1:4002/ | Application: http://127.0.0.1:4001/

Then click on the **INVOKE** ,if it is success it gives response 200 OK

IBM API Connect mangodb - mangodb 1.0.0

Test

Response

Status code: 200 OK

Response time: 60ms

Headers:

```
x-ratelimit-remaining: 98
content-type: application/json; charset=utf-8
x-ratelimit-limit: 100
```

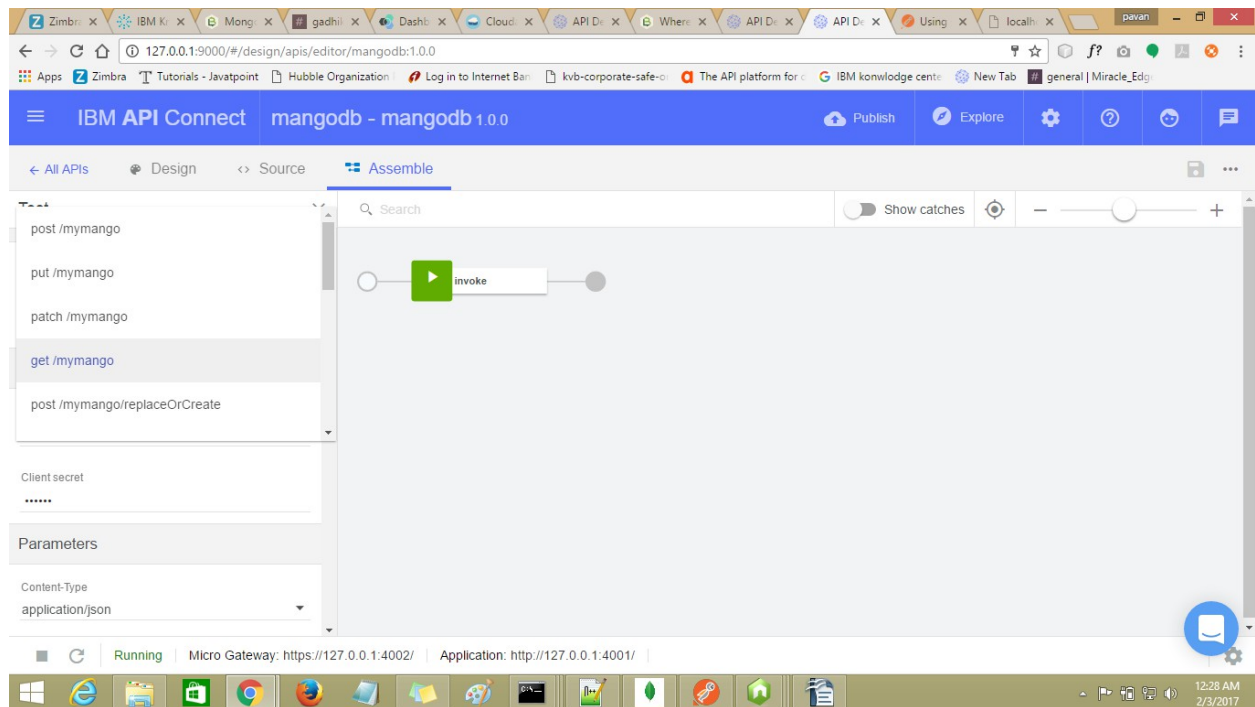
Body:

```
{
  "id": "91",
  "name": "praveen",
  "loc": "guntur"
}
```

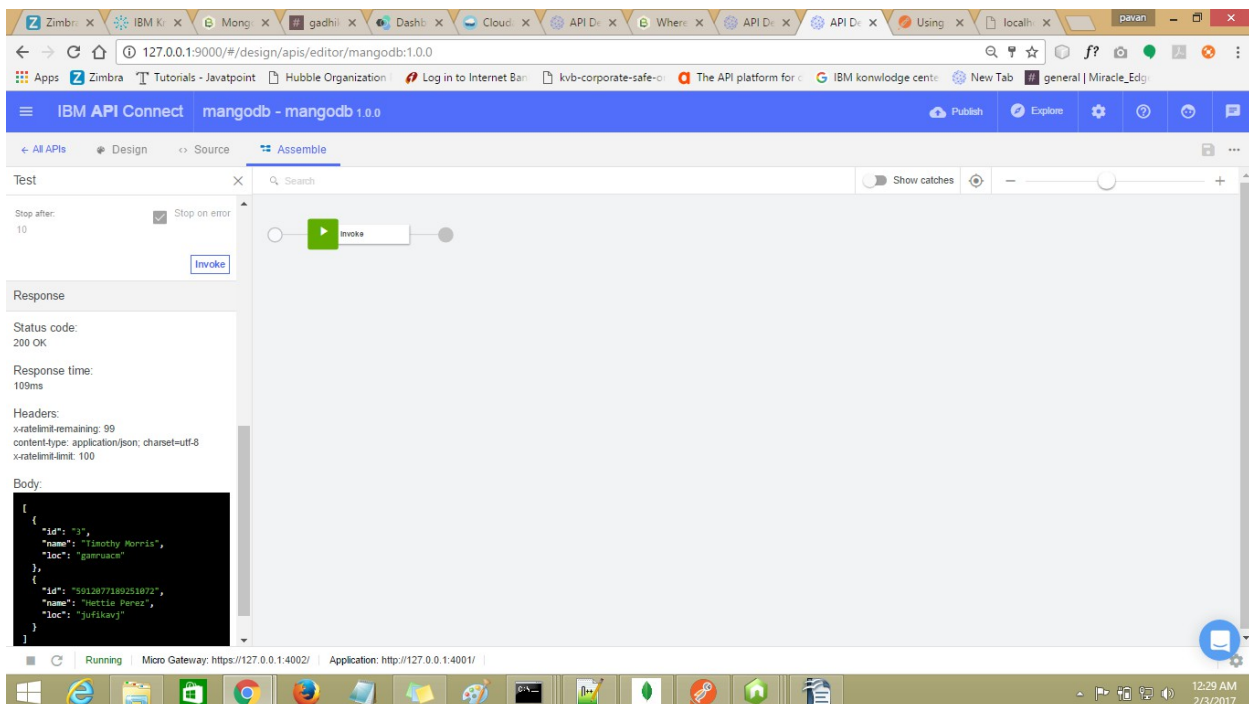
Running | Micro Gateway: https://127.0.0.1:4002/ | Application: http://127.0.0.1:4001/

GET Operation on MongoDB DataBase:

For the Get operation we need to select the invoke operation from the dropdown menu.



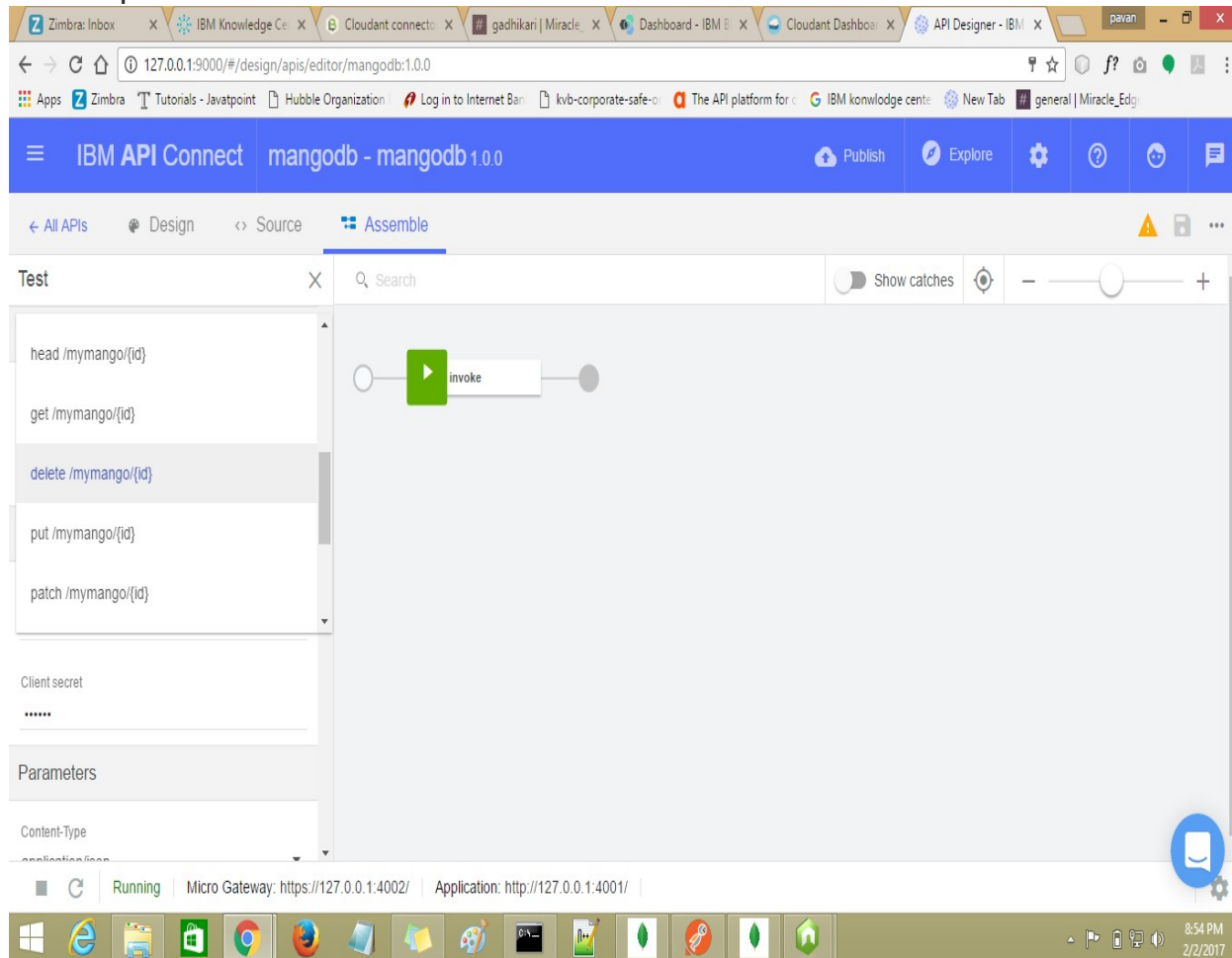
And then click on invoke tab.



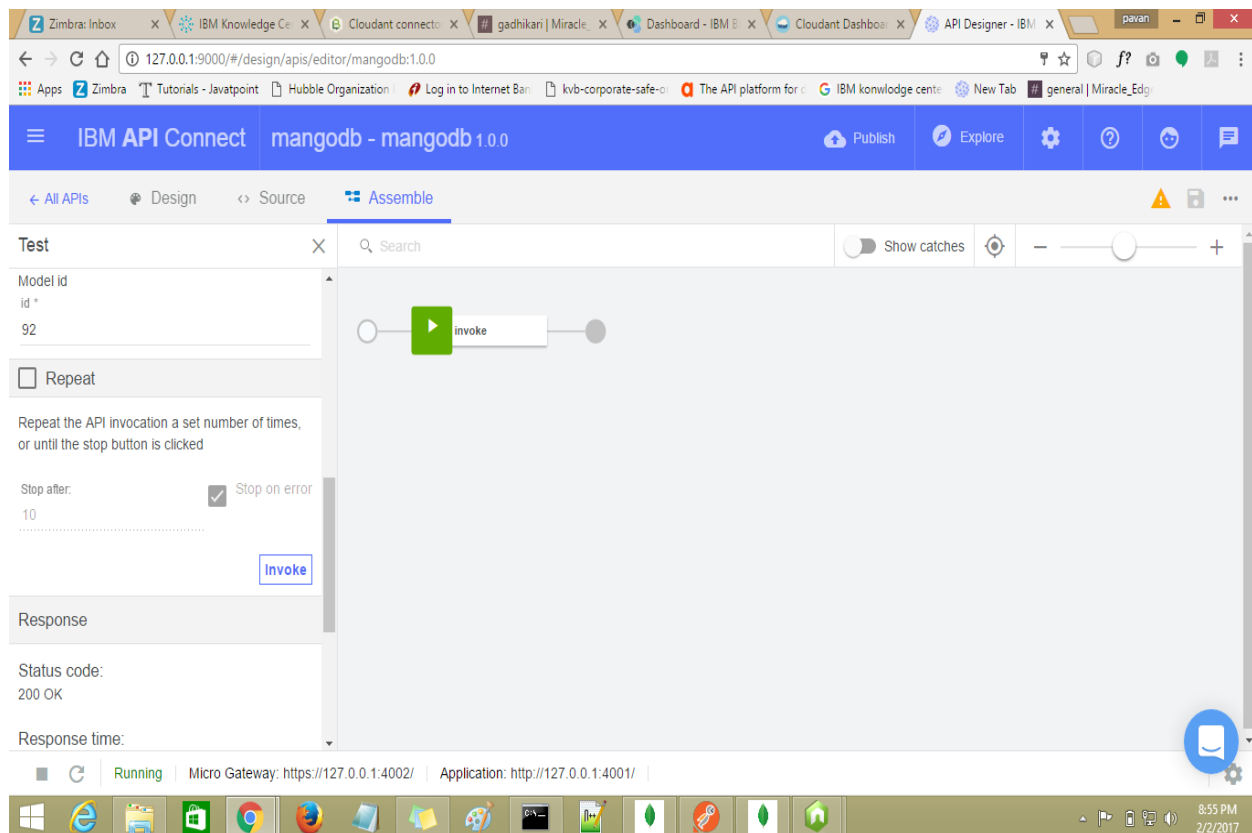
Then the data present in the database can be displayed.

***DELETE** Operation on MongoDB DataBase:*

To perform the delete operation we need to select the reselective operation from the dropdown list.



We can perform the **Delete** operation based on the MODEL ID. Then we need to give the model ID. Then click on INVOKE Method, if it is succeeded it gives response 200,OK.



UPDATE operation on Mongodb DataBase:

To Perform the UPADTE the operation select the respective operation from the drop down menu. For the update operation we can use PUT operation Also Based on the ID.

Local Gateway <https://127.0.0.1:4002/>

mangodb 1.0.0

Operations

- POST /mymango
- PUT /mymango**
- PATCH /mymango
- GET /mymango
- POST /mymango/replaceOr...
- POST /mymango/upsertWit...
- GET /mymango/{id}/exists
- HEAD /mymango/{id}
- GET /mymango/{id}
- DELETE /mymango/{id}
- PUT /mymango/{id}
- PATCH /mymango/{id}
- POST /mymango/{id}/replace
- GET /mymango/findOne

Client secret: SECRET

Content-Type: application/json

Accept: application/json

Parameters

Model instance data

```
data
{
  "id": "3",
  "name": "Lawrence Roy",
  "loc": "huole"
}
```

Show schema | Generate

Call operation

Running | Micro Gateway: <https://127.0.0.1:4002/> | Application: <http://127.0.0.1:4001/>

Then click on Invoke operation,when it gives success it gives response 200 OK.

Local Gateway <https://127.0.0.1:4002/>

mangodb

Operations

- POST /mymango
- PUT /mymango**
- PATCH /mymango
- GET /mymango
- POST /mymango/replaceOr...
- POST /mymango/upsertWit...
- GET /mymango/{id}/exists
- HEAD /mymango/{id}
- GET /mymango/{id}
- DELETE /mymango/{id}
- PUT /mymango/{id}
- PATCH /mymango/{id}
- POST /mymango/{id}/replace
- GET /mymango/findOne

Request

```
PUT https://localhost:4002/api/mymango
Content-Type: application/json
Accept: application/json
X-IBM-Client-Id: default
X-IBM-Client-Secret: SECRET
```

Response

Code: 200 OK

Headers:

```
x-ratelimit-remaining: 99
content-type: application/json; charset=utf-8
x-ratelimit-limit: 100
```

```
{
  "id": "3",
  "name": "Lawrence Roy",
  "loc": "huole"
}
```

Call operation

Running | Micro Gateway: <https://127.0.0.1:4002/> | Application: <http://127.0.0.1:4001/>