Integration of IBM API and Strong Loop

Date: 3rd March, 2016

Santoshi and Geetha Krishna

API management Resources

Miracle Software Systems, Inc.

Contents

Introduction...3

Strong Loop:...3

IBM API Management:...3

Process of Integration:...4

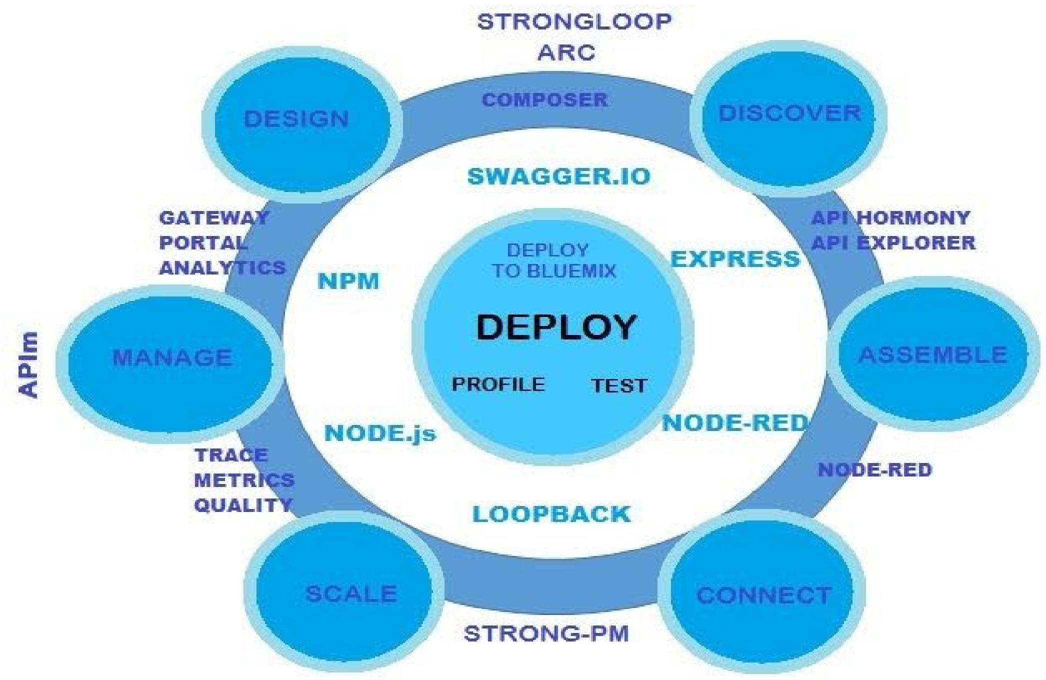
Testing:...18

Reference Links:...22

Introduction

Strong Loop:

- Used to develop a new Node application or API.
- It provides codeless API composition, security
- Can further push this REST API to IBM API Manager or Bluemix. whereas you can use the IBM bluemix and IBM API Manager to build and deploy it on different back ends such as Cloudant, Mongo DB and publish your API on a developer portal where it can be further developed by the developers.



IBM API Management:

IBM API Management (**IBM APIM**) is an API Management platform for use in the API Economy. IBM API Management enables users to create, assemble, manage, secure and socialize web application programming interfaces (APIs).

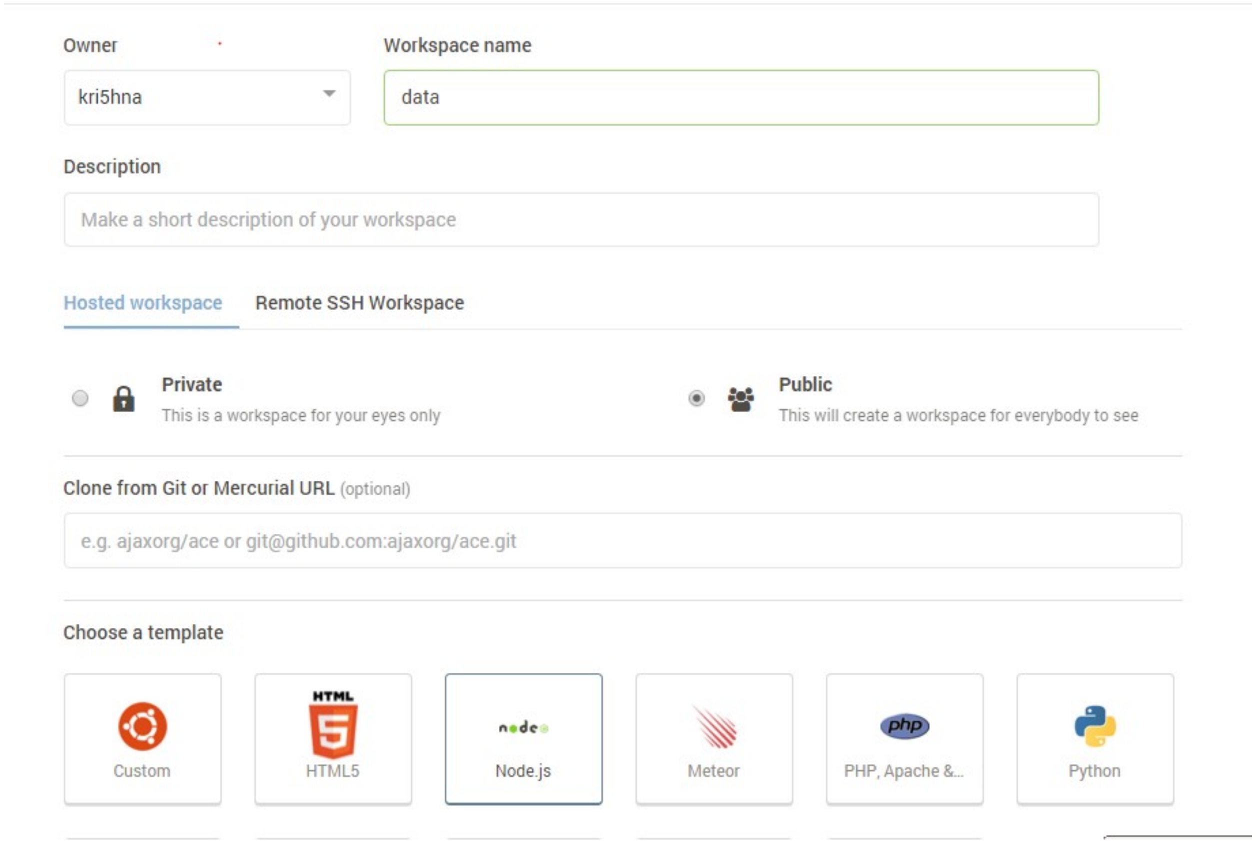
Why we should we integrate IBM API with Strong Loop?

This integration leads us to support “FULL LIFECYCLE of API’s” design, discover, assemble, connect, deploy, scale and manage along with the API composition in Swagger, Node.js and Express.js.

With Strong Loop’s Open API platform represented by Strong Loop Arc used within a cloud-based development platform like Blue mix, an enterprise can apply the Open API Method to rapidly iterate on micro services from new or existing assets to build new engaging application and reuse and wrap existing IT assets at the speed required by new users of digital enterprise services

Process of Integration:

Create an account in Cloud9 and create a work space using template Node.js:



Owner: kri5hna

Workspace name: data

Description: Make a short description of your workspace

Private — This is a workspace for your eyes only

[✓ Public — This will create a workspace for everybody to see]

Clone from Git or Mercurial URL (optional): e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git

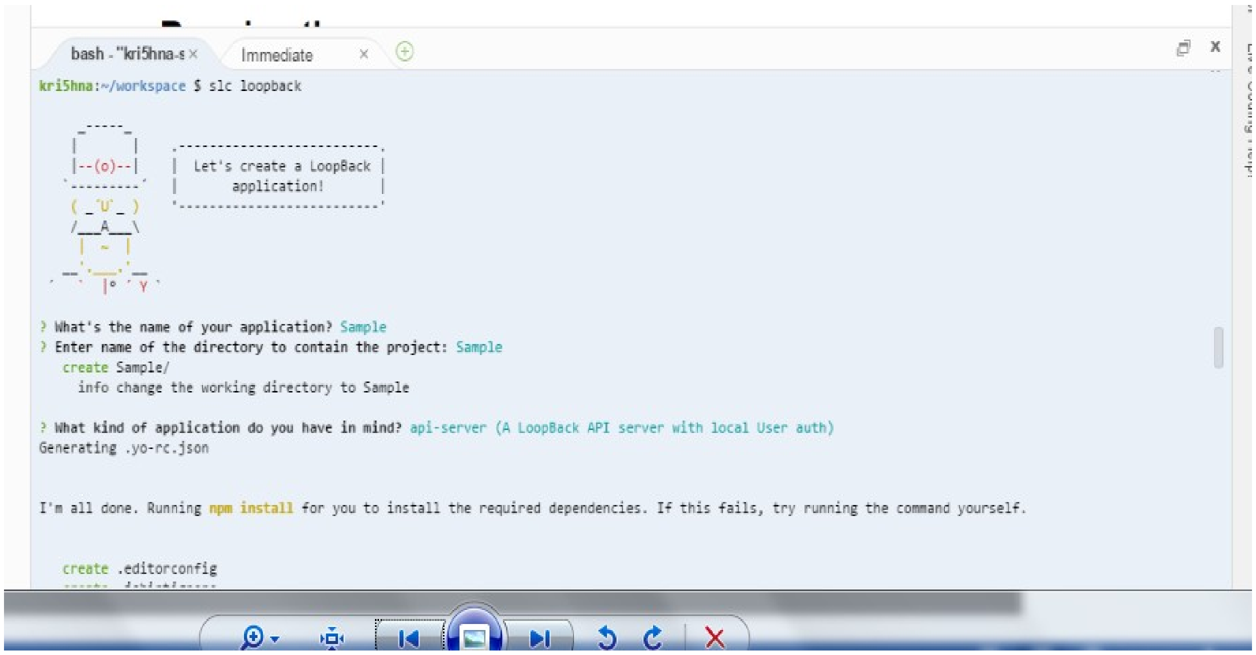
Choose a template: Custom, HTML5, [✓ Node.js], Meteor, PHP, Apache &..., Python

Install the Strong Loop using the following commands:

- **npm install –g strongloop:** using this command we can create the strong loop environment globally.

Create a Loop Back Application using the following command:

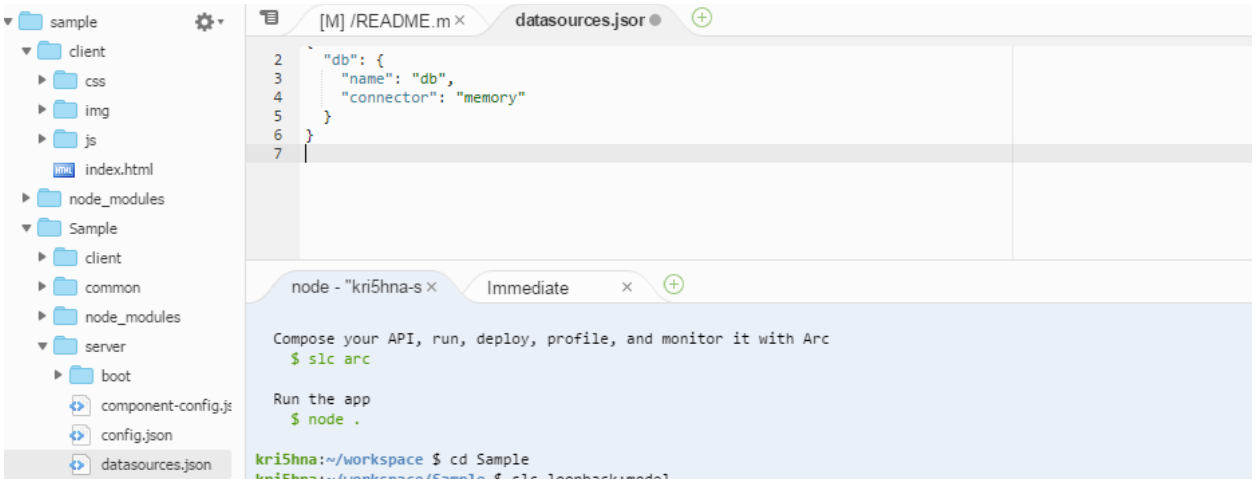
- **“slc loopback”** and follow the prompts(console) to create the application



```
kri5hna:~/workspace $ slc loopback

     _-----_
    |       |    .--------------------------.
    |--(o)--|    |  Let's create a LoopBack |
   `---------´   |       application!       |
    ( _´U`_ )    '--------------------------'
    /___A___\
     |  ~  |
   __'.___.'__
 ´   `  |° ´ Y `

? What's the name of your application? Sample
? Enter name of the directory to contain the project: Sample
   create Sample/
     info change the working directory to Sample

? What kind of application do you have in mind? api-server (A LoopBack API server with local User auth)
Generating .yo-rc.json


I'm all done. Running npm install for you to install the required dependencies. If this fails, try running the command yourself.


   create .editorconfig
```

Now go to workspace

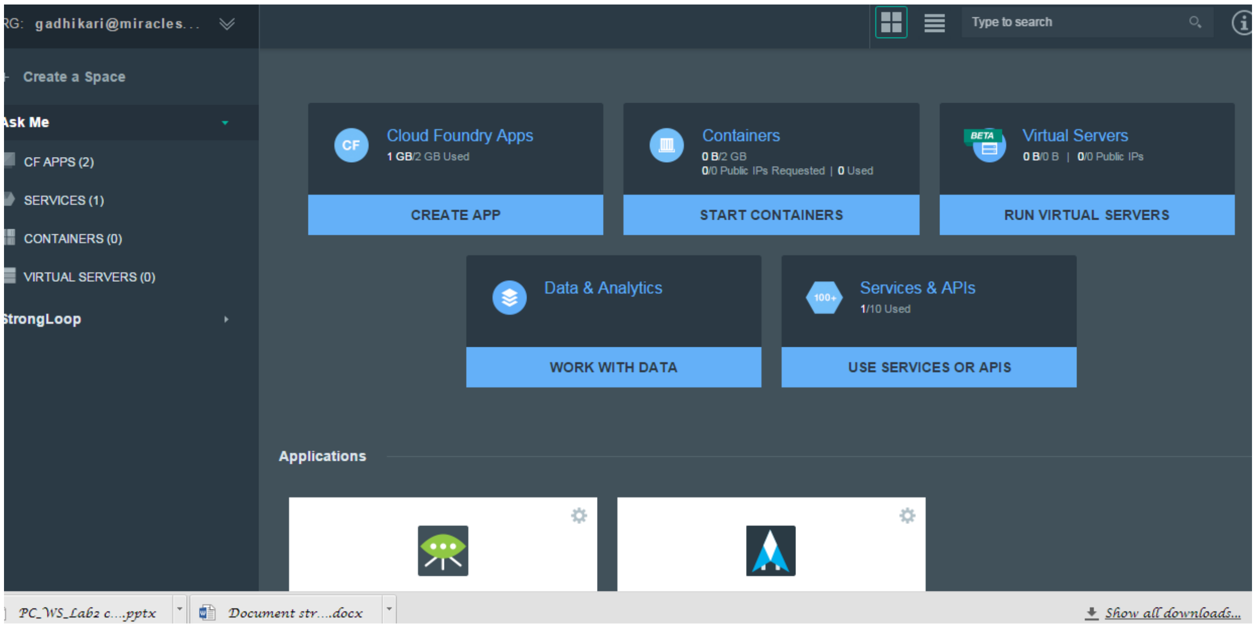
Cloud9 directoryApplicationnode modules server datasources.jcon



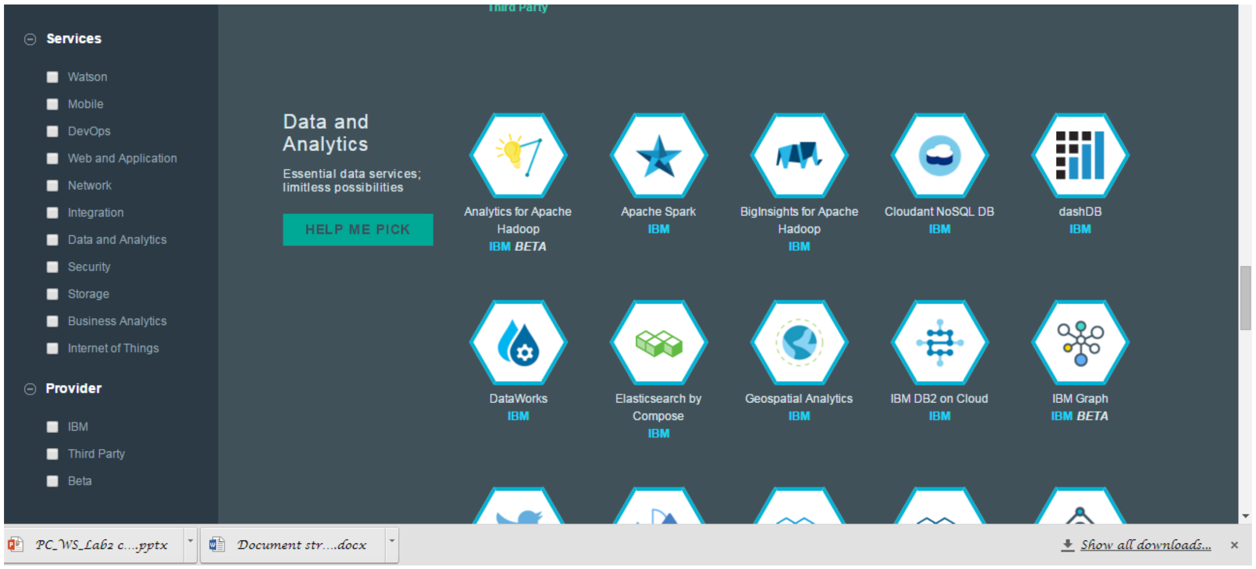
```
  "db": {
    "name": "db",
    "connector": "memory"
  }
}
```

```
Compose your API, run, deploy, profile, and monitor it with Arc
  $ slc arc

Run the app
  $ node .

kri5hna:~/workspace $ cd Sample
```

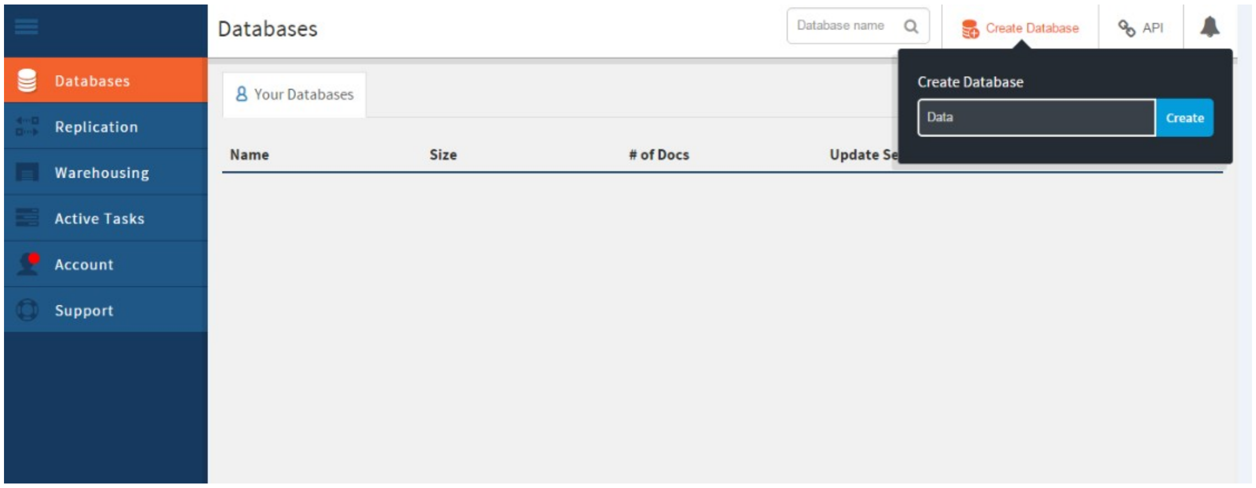
For cloudant credentials go to your IBM Bluemix Dashboard Services&API's



Now go to Data&Analytics select Cloudant NoSQL DB

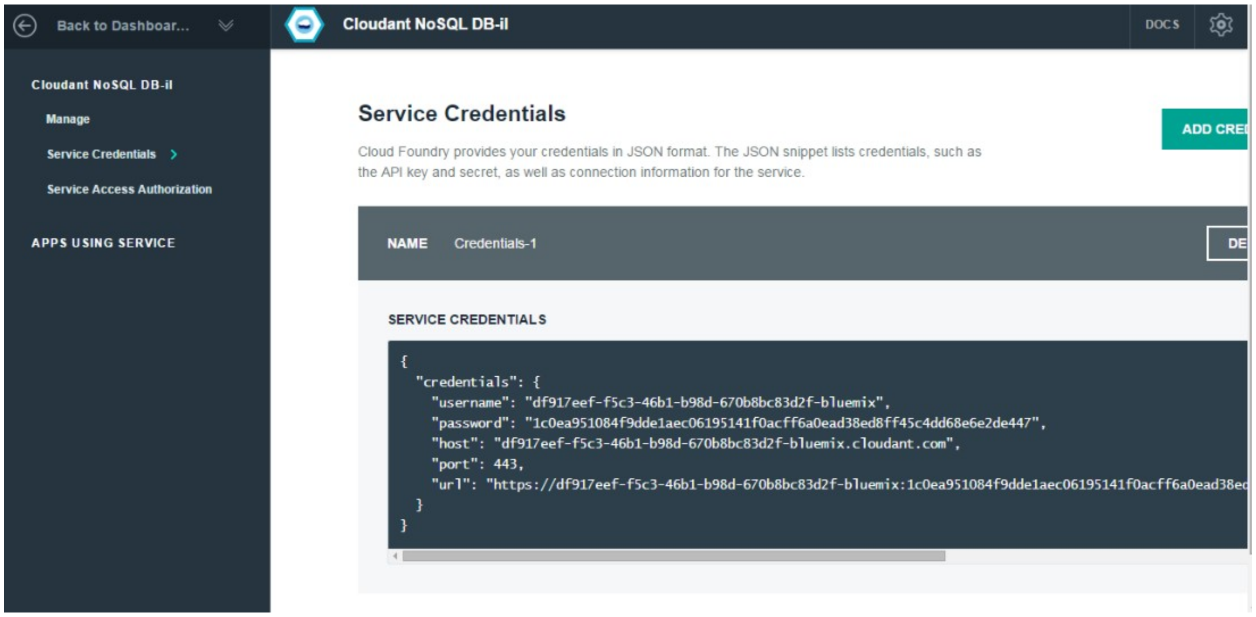


Create a service and Launch it and now create a database



Now go to dashboard service credentials add credentials.

and copy username and password to datasources.json



Service Credentials

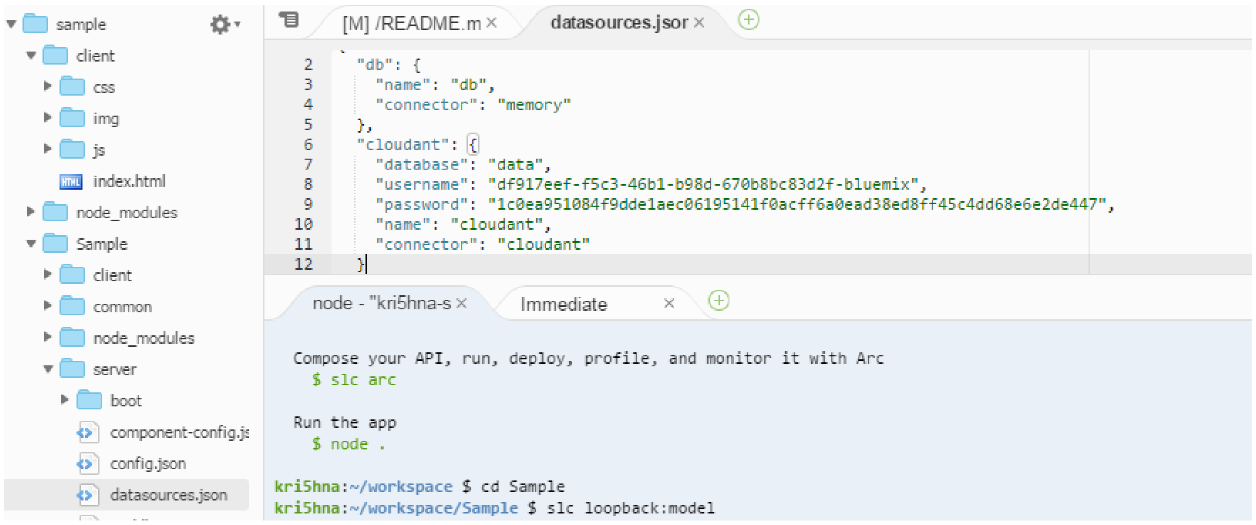
Cloud Foundry provides your credentials in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

NAME Credentials-1

SERVICE CREDENTIALS

```
{
  "credentials": {
    "username": "df917eef-f5c3-46b1-b98d-670b8bc83d2f-bluemix",
    "password": "1c0ea951084f9dde1aec06195141f0acff6a0ead38ed8ff45c4dd68e6e2de447",
    "host": "df917eef-f5c3-46b1-b98d-670b8bc83d2f-bluemix.cloudant.com",
    "port": 443,
    "url": "https://df917eef-f5c3-46b1-b98d-670b8bc83d2f-bluemix:1c0ea951084f9dde1aec06195141f0acff6a0ead38ed
  }
}
```

Change the database specified in datasources.json to the name you've given during creation of IBM Cloudant database.



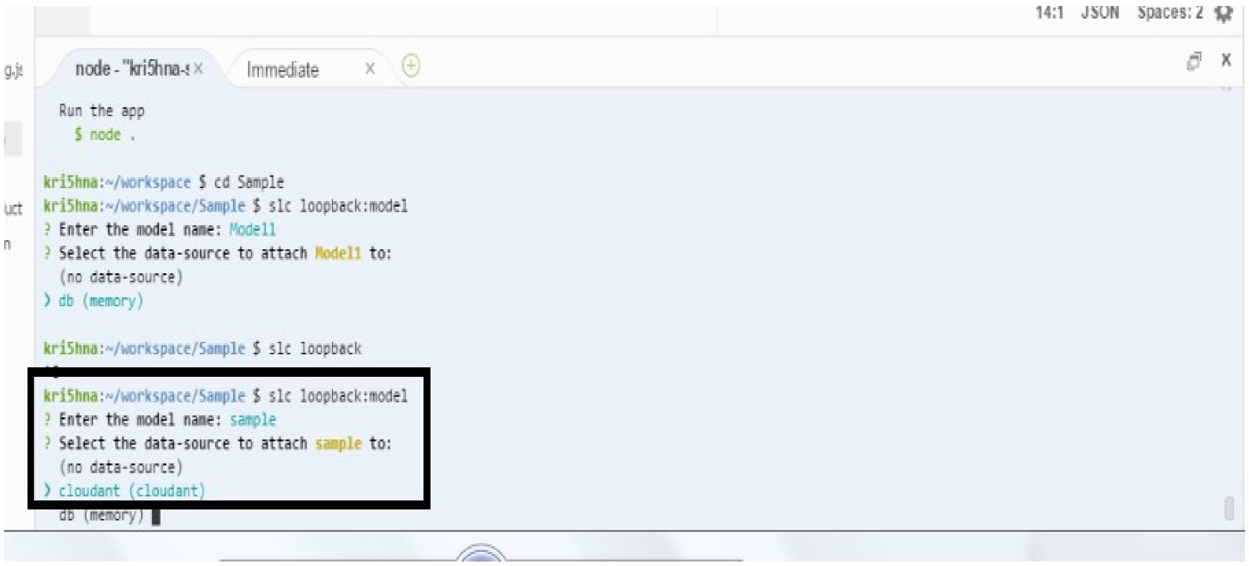
```
  "db": {
    "name": "db",
    "connector": "memory"
  },
  "cloudant": {
    "database": "data",
    "username": "df917eef-f5c3-46b1-b98d-670b8bc83d2f-bluemix",
    "password": "1c0ea951084f9dde1aec06195141f0acff6a0ead38ed8ff45c4dd68e6e2de447",
    "name": "cloudant",
    "connector": "cloudant"
  }
```

```
Compose your API, run, deploy, profile, and monitor it with Arc
  $ slc arc

Run the app
  $ node .

kri5hna:~/workspace $ cd Sample
kri5hna:~/workspace/Sample $ slc loopback:model
```

Go into your application directory "cd App name"

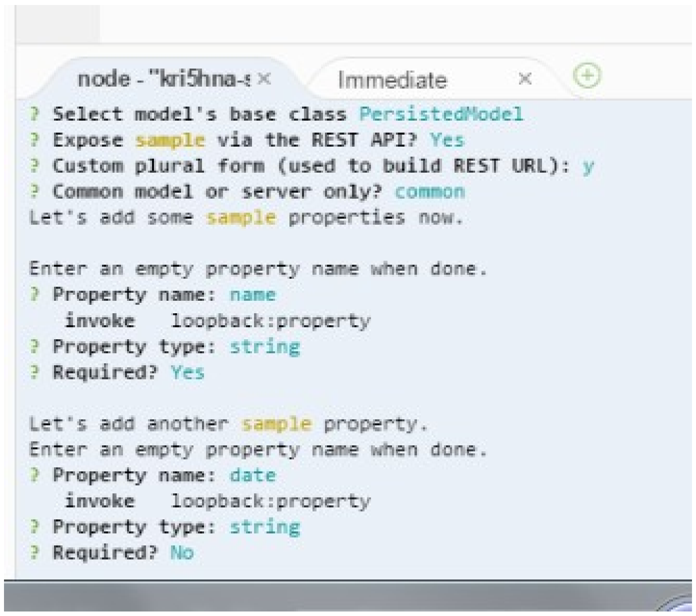
"slc loopback :model" and follow prompt

select the cloudant db



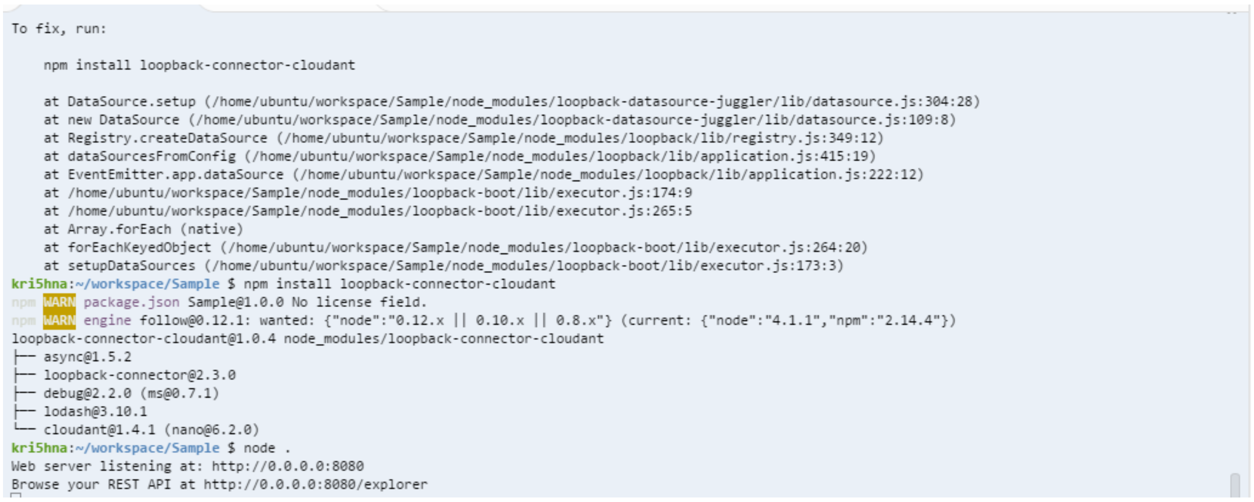
```
Run the app
  $ node .

kri5hna:~/workspace $ cd Sample
kri5hna:~/workspace/Sample $ slc loopback:model
? Enter the model name: Model1
? Select the data-source to attach Model1 to:
  (no data-source)
❯ db (memory)

kri5hna:~/workspace/Sample $ slc loopback

kri5hna:~/workspace/Sample $ slc loopback:model
? Enter the model name: sample
? Select the data-source to attach sample to:
  (no data-source)
❯ cloudant (cloudant)
  db (memory)
```

Follow the screen shot for further steps



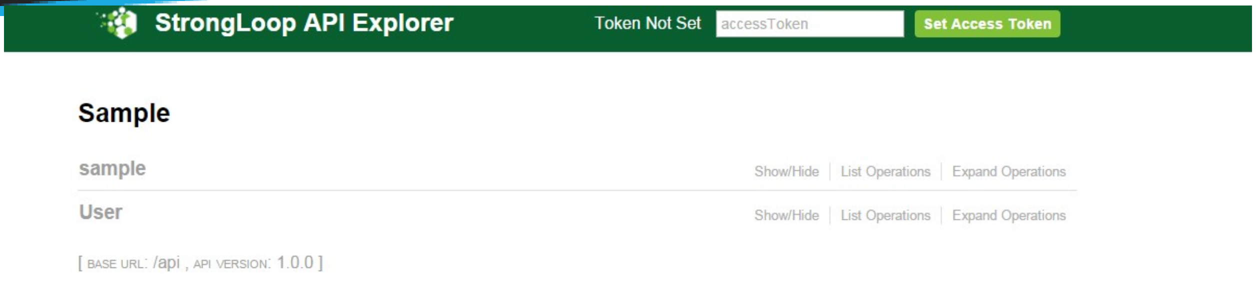
```
? Select model's base class PersistedModel
? Expose sample via the REST API? Yes
? Custom plural form (used to build REST URL): y
? Common model or server only? common
Let's add some sample properties now.

Enter an empty property name when done.
? Property name: name
   invoke   loopback:property
? Property type: string
? Required? Yes

Let's add another sample property.
Enter an empty property name when done.
? Property name: date
   invoke   loopback:property
? Property type: string
? Required? No
```

Now run the command npm install loopback-connector-cloudant to install cloudant DB connector



```
To fix, run:

    npm install loopback-connector-cloudant

    at DataSource.setup (/home/ubuntu/workspace/Sample/node_modules/loopback-datasource-juggler/lib/datasource.js:304:28)
    at new DataSource (/home/ubuntu/workspace/Sample/node_modules/loopback-datasource-juggler/lib/datasource.js:109:8)
    at Registry.createDataSource (/home/ubuntu/workspace/Sample/node_modules/loopback/lib/registry.js:349:12)
    at dataSourcesFromConfig (/home/ubuntu/workspace/Sample/node_modules/loopback/lib/application.js:415:19)
    at EventEmitter.app.dataSource (/home/ubuntu/workspace/Sample/node_modules/loopback/lib/application.js:222:12)
    at /home/ubuntu/workspace/Sample/node_modules/loopback-boot/lib/executor.js:174:9
    at /home/ubuntu/workspace/Sample/node_modules/loopback-boot/lib/executor.js:265:5
    at Array.forEach (native)
    at forEachKeyedObject (/home/ubuntu/workspace/Sample/node_modules/loopback-boot/lib/executor.js:264:20)
    at setupDataSources (/home/ubuntu/workspace/Sample/node_modules/loopback-boot/lib/executor.js:173:3)
kri5hna:~/workspace/Sample $ npm install loopback-connector-cloudant
npm WARN package.json Sample@1.0.0 No license field.
npm WARN engine follow@0.12.1: wanted: {"node":"0.12.x || 0.10.x || 0.8.x"} (current: {"node":"4.1.1","npm":"2.14.4"})
loopback-connector-cloudant@1.0.4 node_modules/loopback-connector-cloudant
├── async@1.5.2
├── loopback-connector@2.3.0
├── debug@2.2.0 (ms@0.7.1)
├── lodash@3.10.1
└── cloudant@1.4.1 (nano@6.2.0)
kri5hna:~/workspace/Sample $ node .
Web server listening at: http://0.0.0.0:8080
Browse your REST API at http://0.0.0.0:8080/explorer
```

Now run the URL http://0.0.0.0:8080/explorer *in the browser*

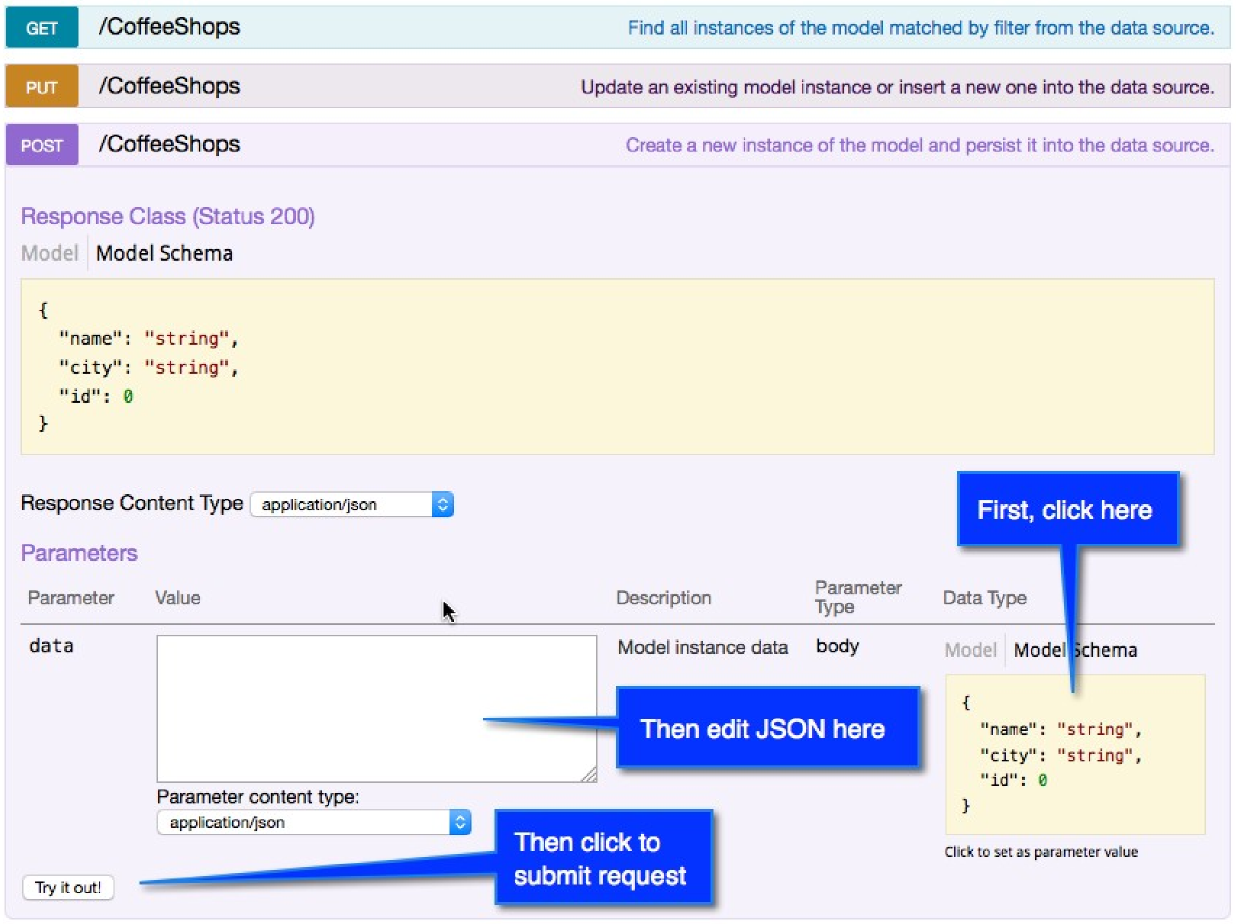


Sample

sample	Show/Hide | List Operations | Expand Operations
User	Show/Hide | List Operations | Expand Operations

[BASE URL: /api , API VERSION: 1.0.0]

Now select the post method as follow



GET /CoffeeShops Find all instances of the model matched by filter from the data source.

PUT /CoffeeShops Update an existing model instance or insert a new one into the data source.

POST /CoffeeShops Create a new instance of the model and persist it into the data source.

Response Class (Status 200)

```
{
  "name": "string",
  "city": "string",
  "id": 0
}
```

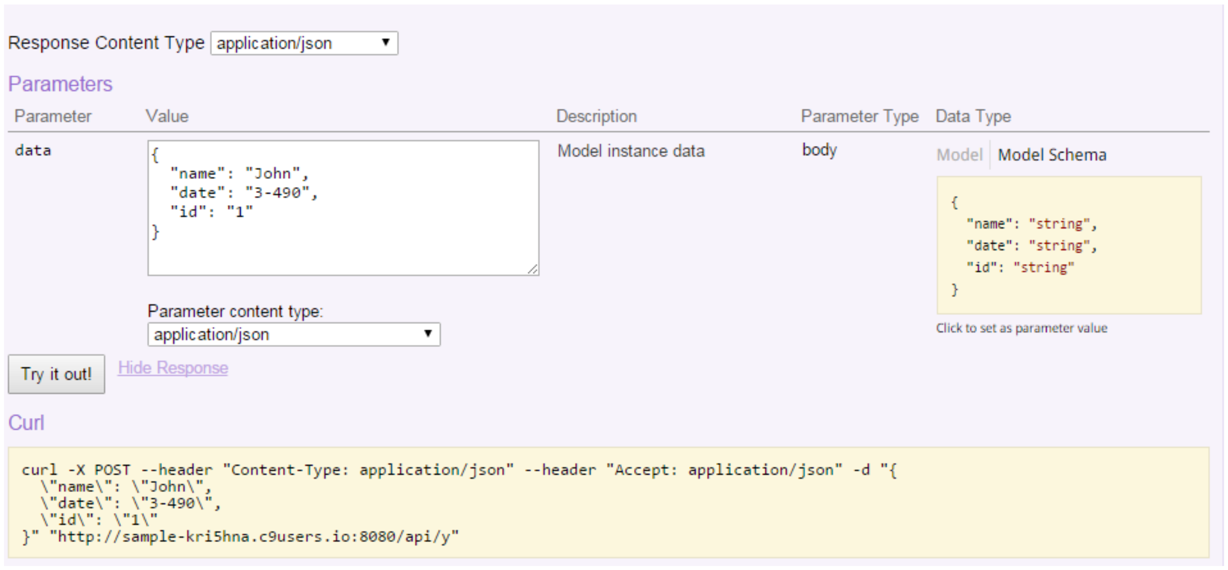
Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	Parameter content type: application/json	Model instance data	body	Model Model Schema

```
{
  "name": "string",
  "city": "string",
  "id": 0
}
```

Insert the values in the post method and try it!



Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	{ "name": "John", "date": "3-490", "id": "1" }	Model instance data	body	Model Model Schema

```
{
  "name": "string",
  "date": "string",
  "id": "string"
}
```

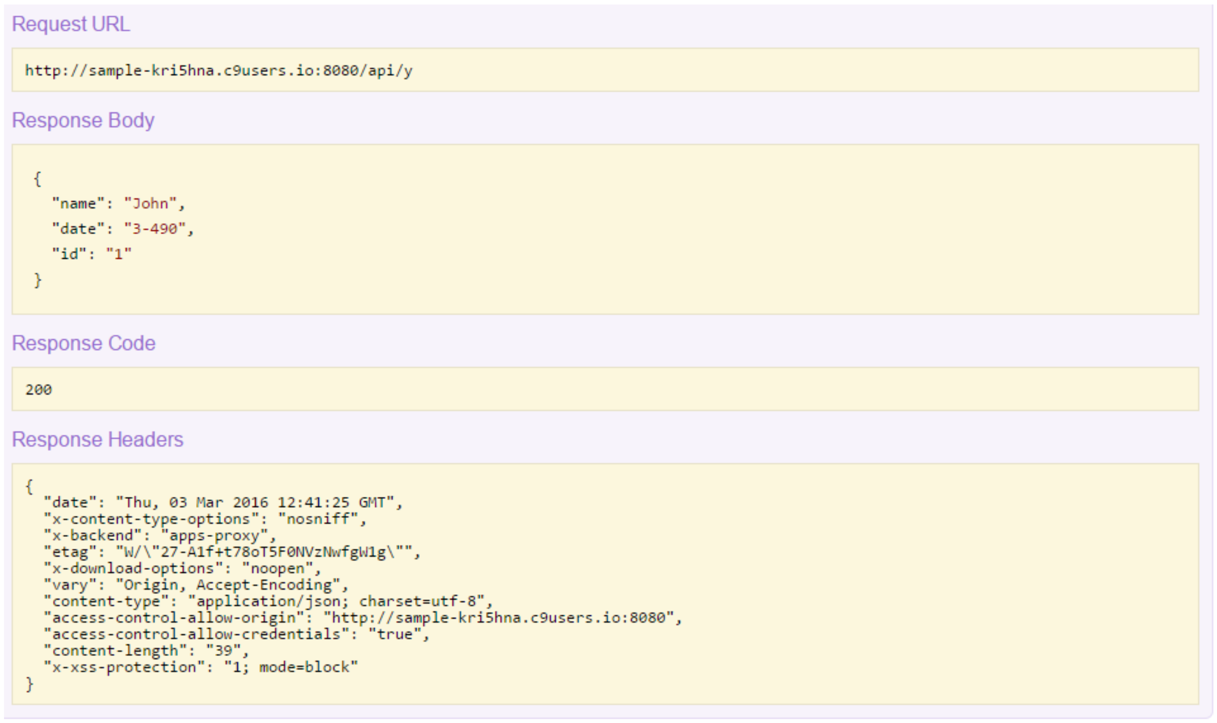
Click to set as parameter value

Parameter content type: application/json

Curl

```
curl -X POST --header "Content-Type: application/json" --header "Accept: application/json" -d "{
  \"name\": \"John\",
  \"date\": \"3-490\",
  \"id\": \"1\"
}" "http://sample-kri5hna.c9users.io:8080/api/y"
```

Then you will get response as shown below:



Request URL

```
http://sample-kri5hna.c9users.io:8080/api/y
```

Response Body

```
{
  "name": "John",
  "date": "3-490",
  "id": "1"
}
```

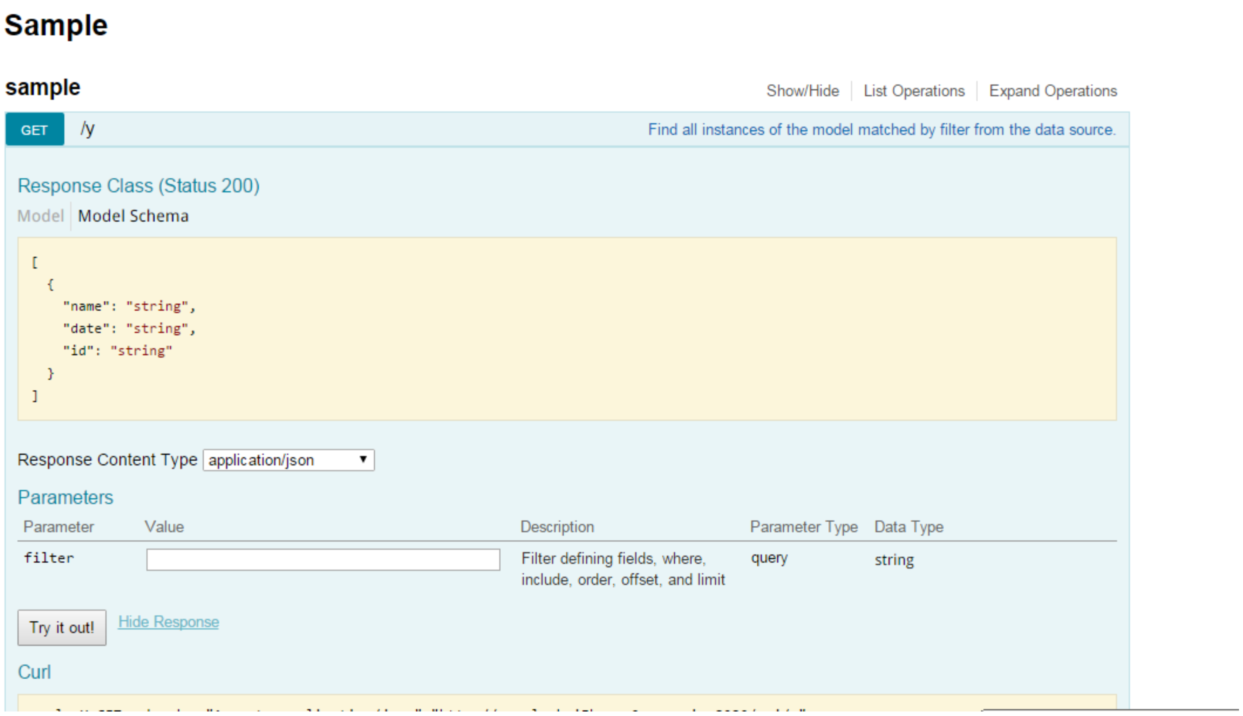
Response Code

```
200
```

Response Headers

```
{
  "date": "Thu, 03 Mar 2016 12:41:25 GMT",
  "x-content-type-options": "nosniff",
  "x-backend": "apps-proxy",
  "etag": "W/\"27-A1f+t78oT5F0NVzNwfgW1g\"",
  "x-download-options": "noopen",
  "vary": "Origin, Accept-Encoding",
  "content-type": "application/json; charset=utf-8",
  "access-control-allow-origin": "http://sample-kri5hna.c9users.io:8080",
  "access-control-allow-credentials": "true",
  "content-length": "39",
  "x-xss-protection": "1; mode=block"
}
```

Expand GET method to get the request URL by clicking on "Try it out".


Sample

sample

Show/Hide | List Operations | Expand Operations

GET /y Find all instances of the model matched by filter from the data source.

Response Class (Status 200)

Model | Model Schema

```
[
  {
    "name": "string",
    "date": "string",
    "id": "string"
  }
]
```

Response Content Type application/json

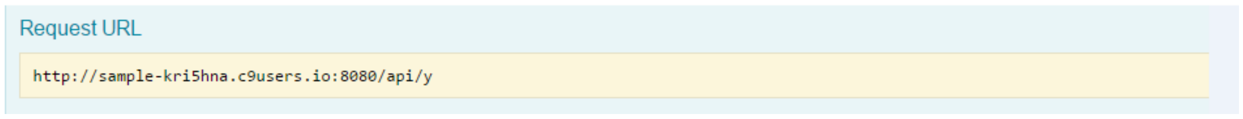
Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter		Filter defining fields, where, include, order, offset, and limit	query	string

Try it out! Hide Response

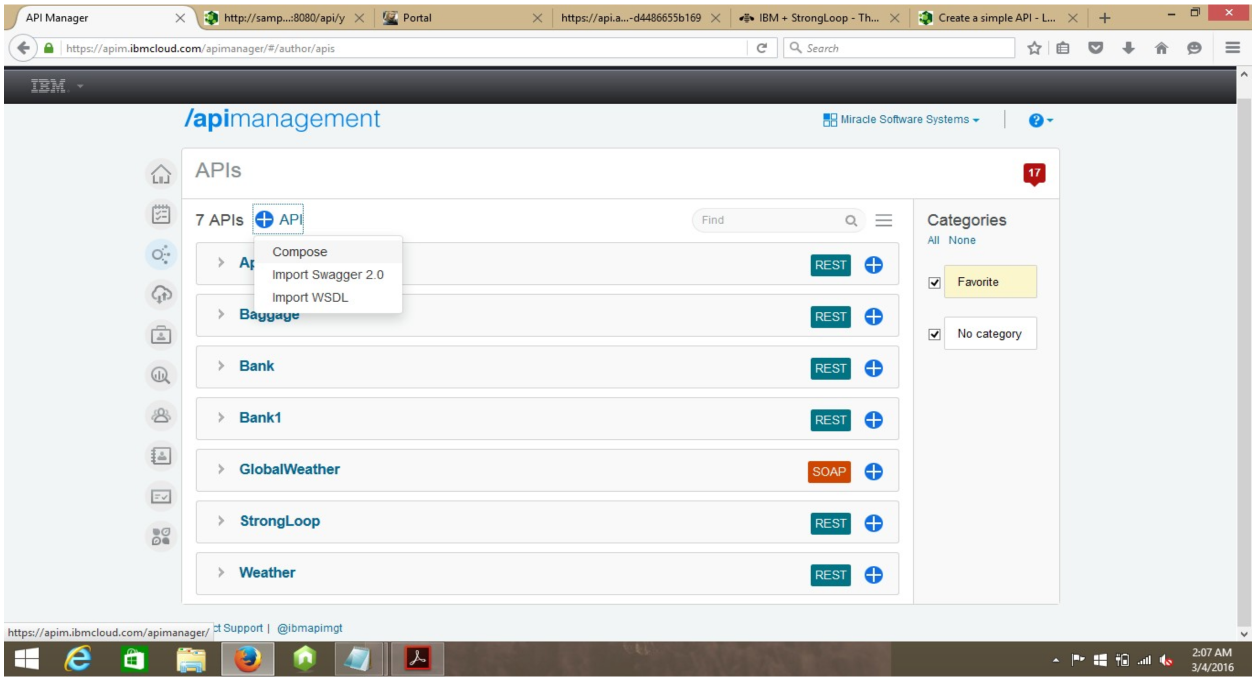
Curl

Copy the request URL:

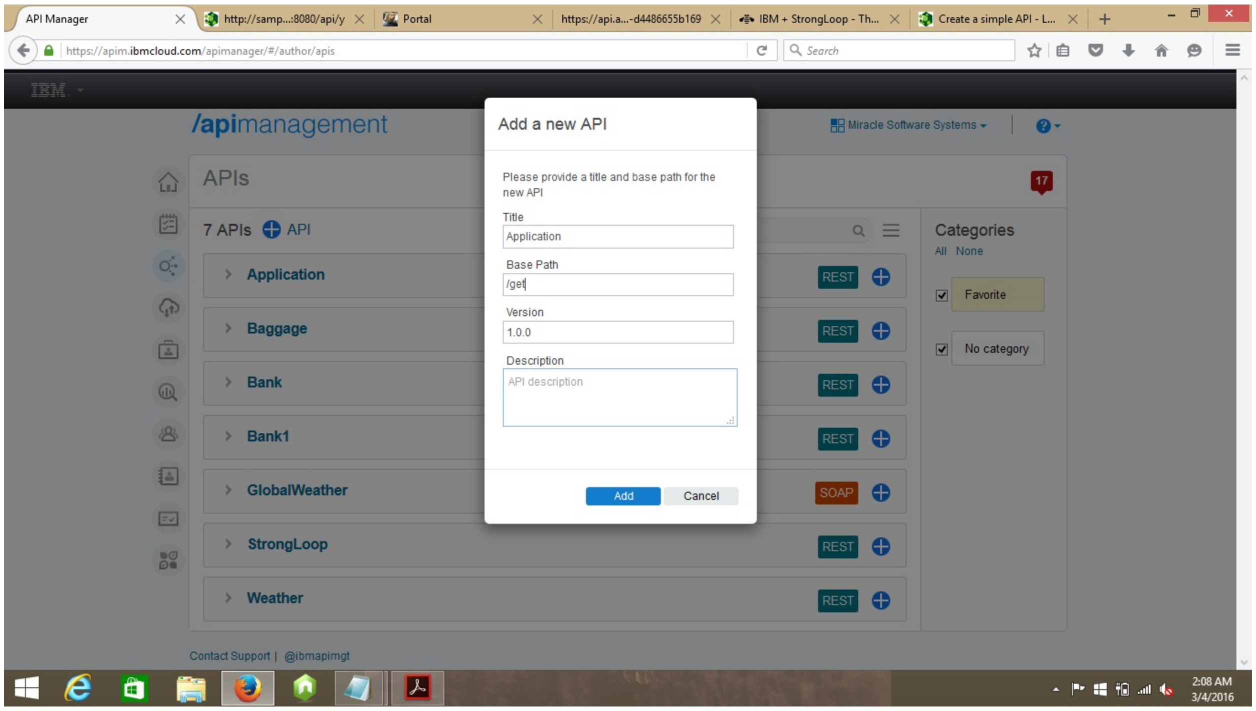

Request URL

```
http://sample-kri5hna.c9users.io:8080/api/y
```

Create a new API in IBM API Manager



Add a new API and specify the base path


Add a new API

Please provide a title and base path for the new API

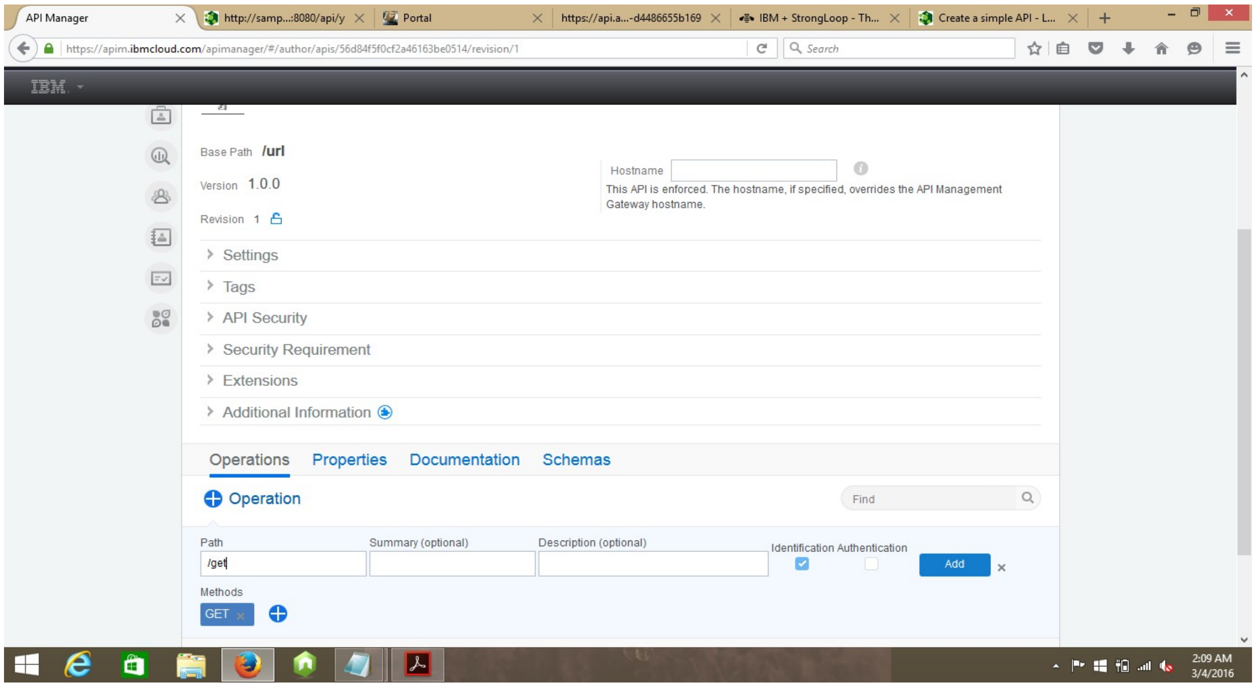
Title: Application

Base Path: /get

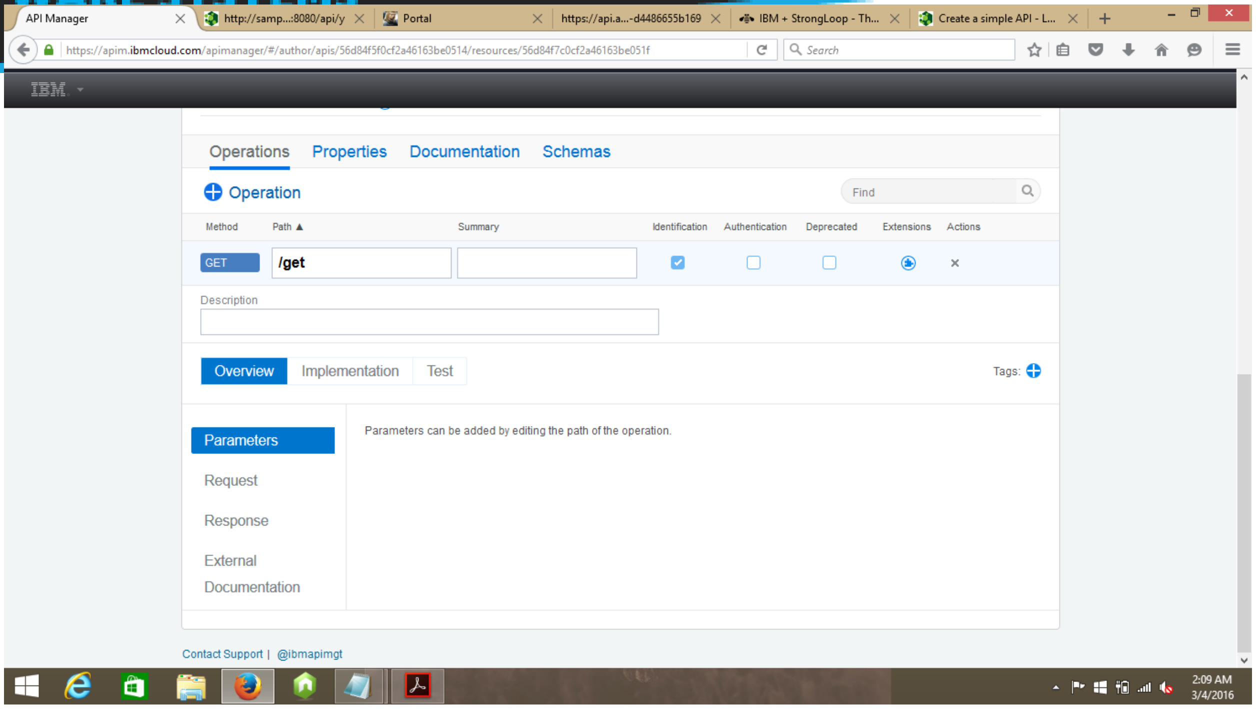
Version: 1.0.0

Description: API description

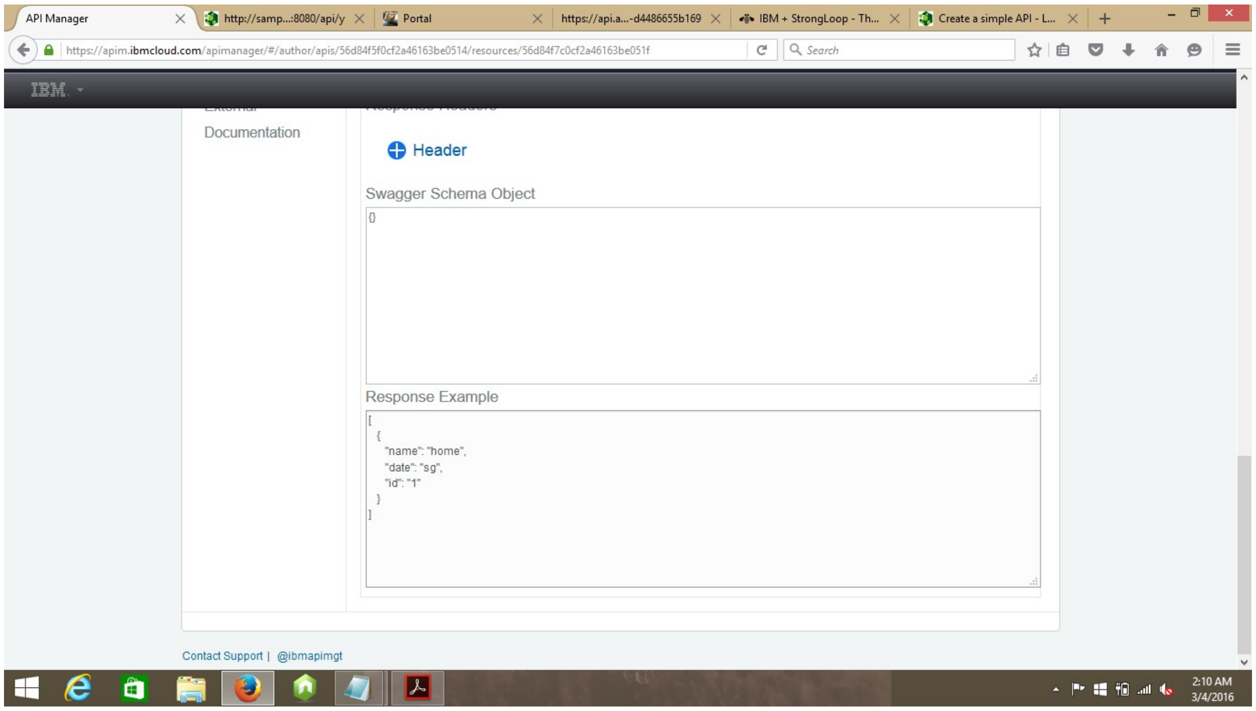
Add a new operation and specify path and select method you require:



Copy the response which you get while using request URL



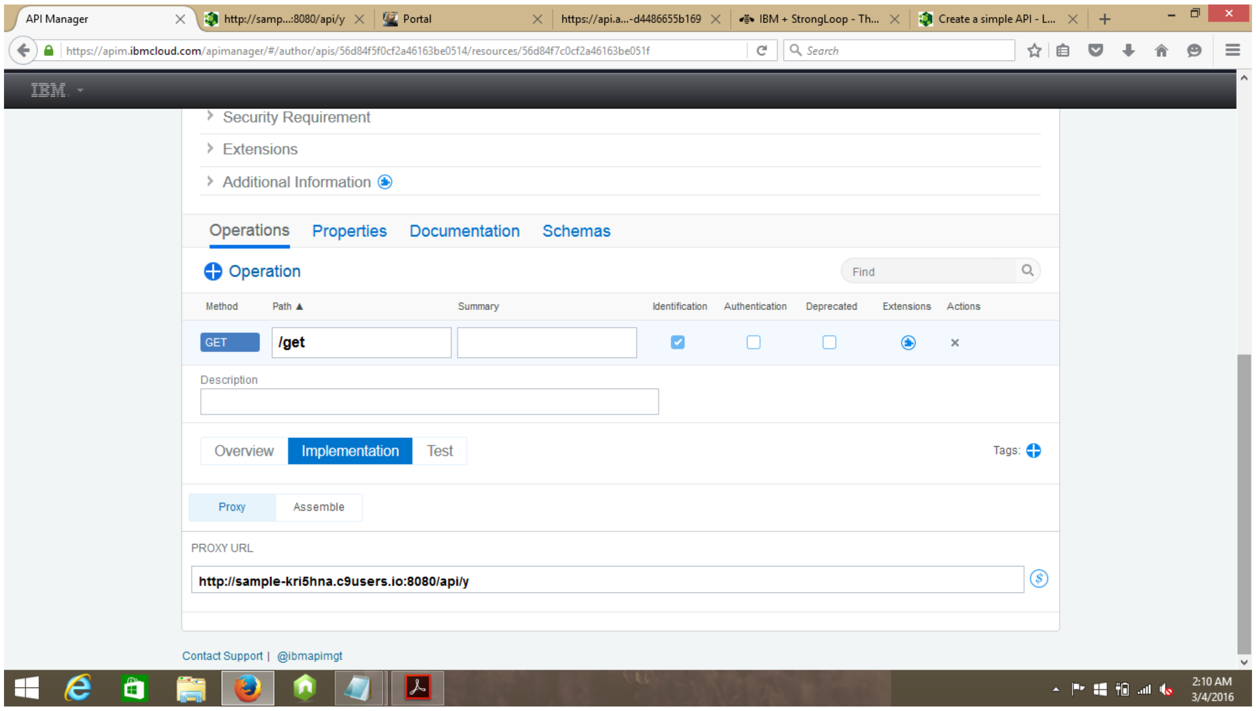
Paste it in the response body and check whether the given response is in validate form


API Manager
https://apim.ibmcloud.com/apimanager/#/author/apis/56d84f5f0cf2a46163be0514/resources/56d84f7c0cf2a46163be051f
IBM
Documentation
Header
Swagger Schema Object
Response Example
Contact Support | @ibmapimgt

```
{}
```

```
[
  {
    "name": "home",
    "date": "sg",
    "id": "1"
  }
]
```

Now move to Implementation and paste the request URL in Proxy.



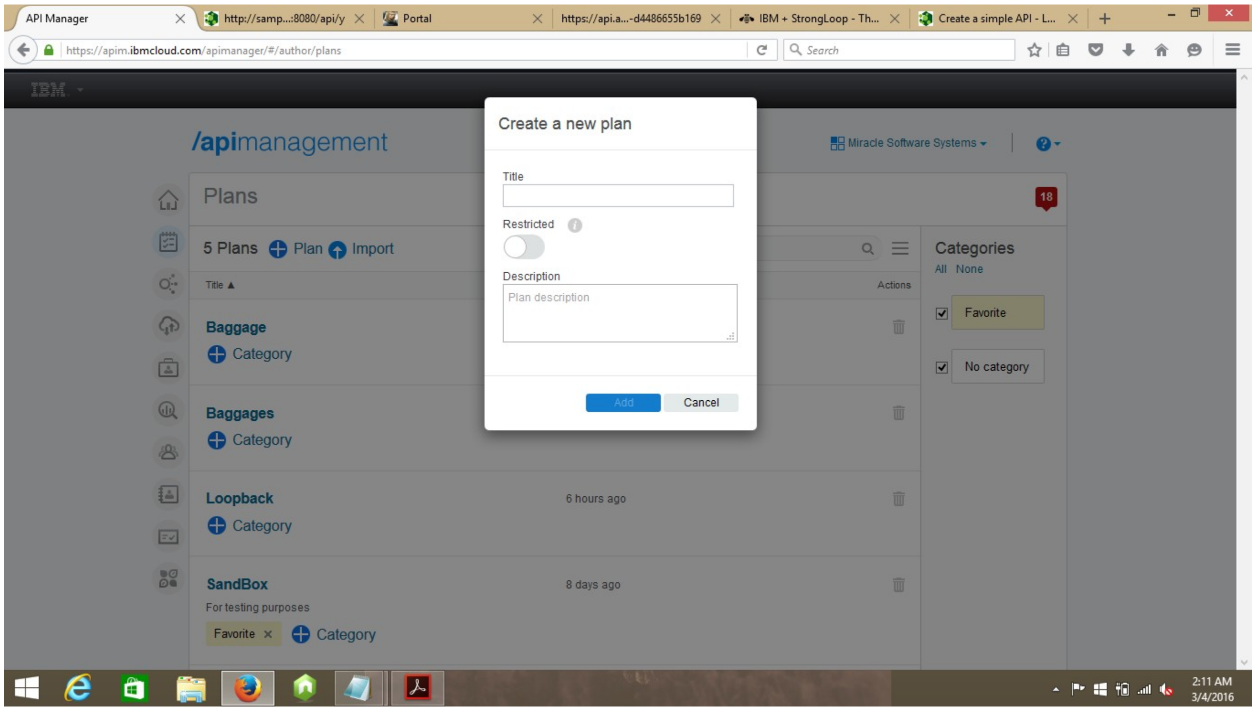
https://apim.ibmcloud.com/apimanager/#/author/apis/56d84f5f0cf2a46163be0514/resources/56d84f7c0cf2a46163be051f

/get

PROXY URL

http://sample-kri5hna.c9users.io:8080/api/y

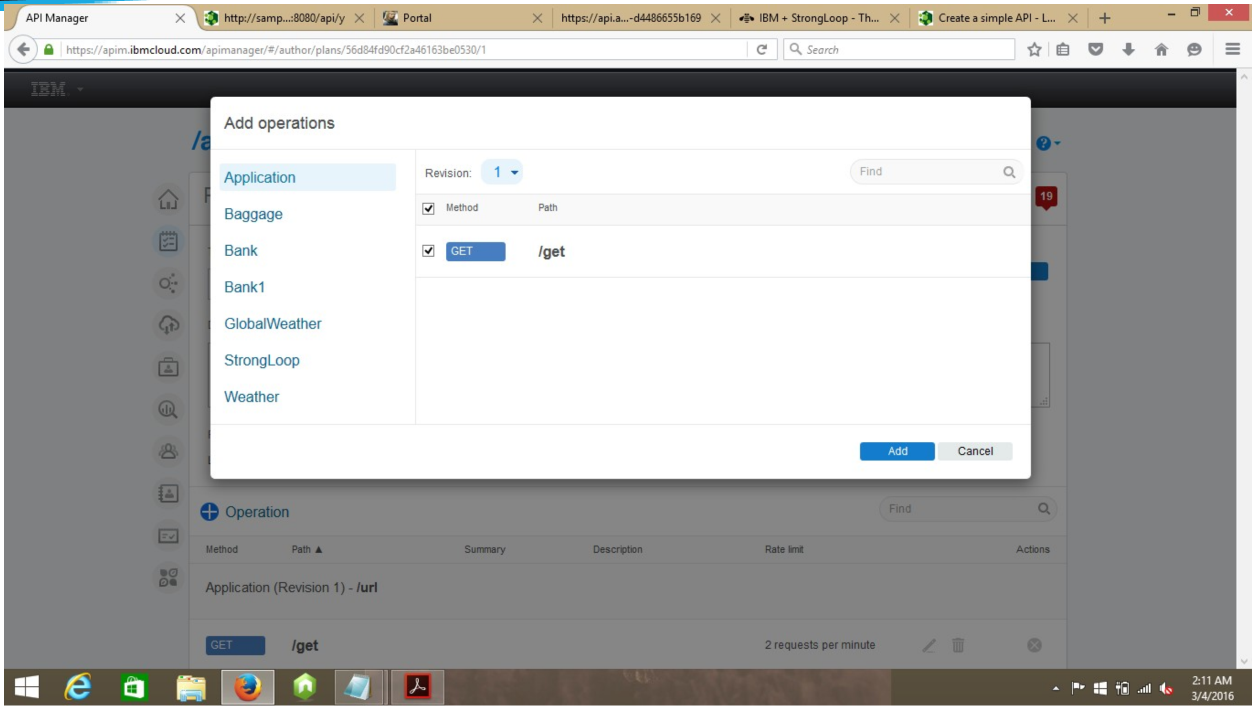
Create a new plan



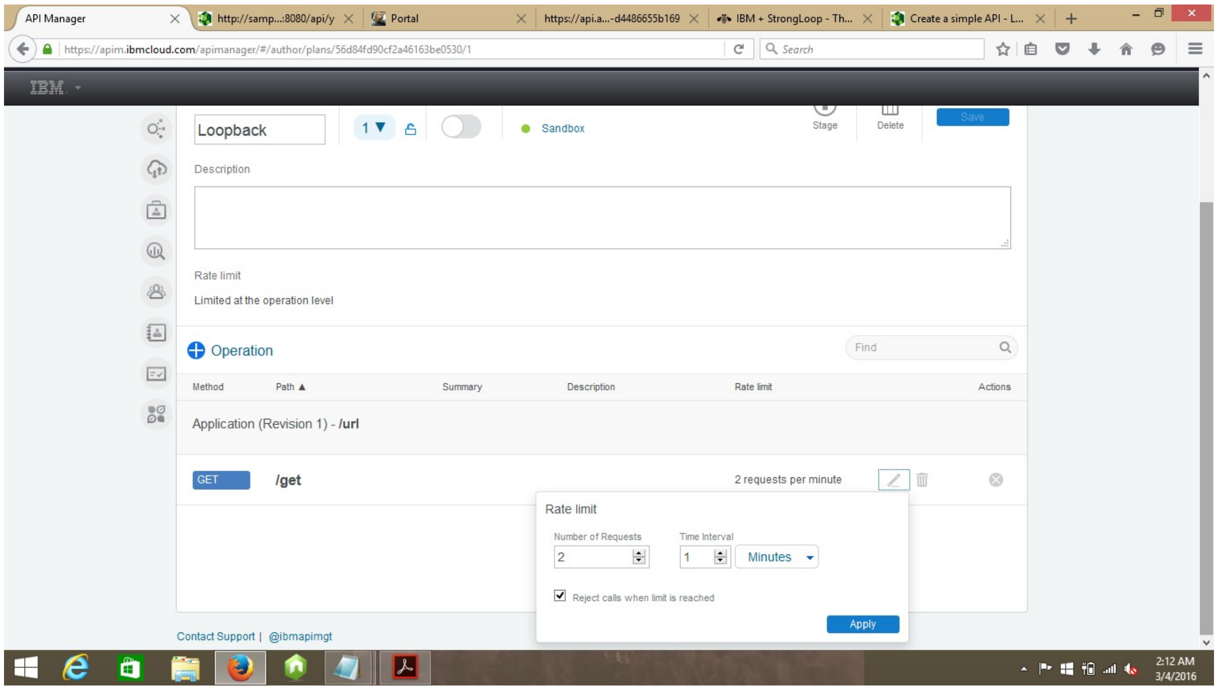
https://apim.ibmcloud.com/apimanager/#/author/plans

Baggage	
Baggages	
Loopback	6 hours ago
SandBox For testing purposes	8 days ago

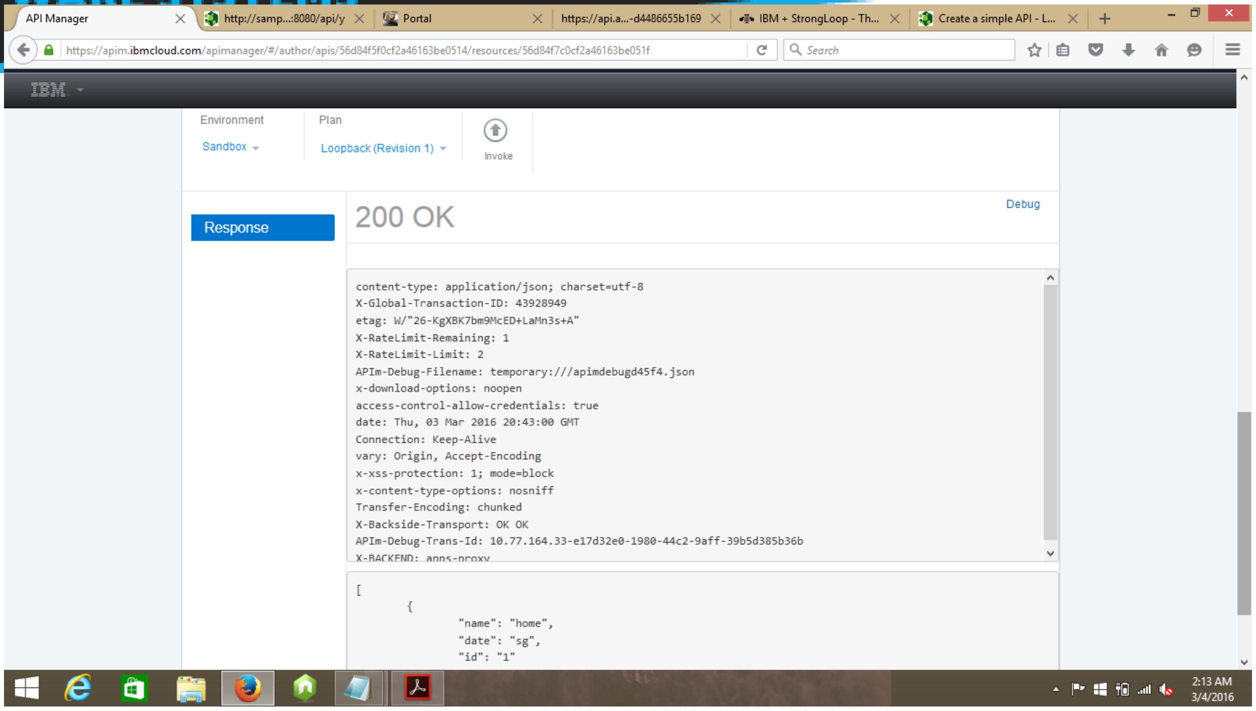
Add a new operation and select the required API you require



You can specify the rate limit



Invoke the operation and check response


API Manager
http://samp...:8080/api/y
Portal
https://api.a...-d4486655b169
IBM + StrongLoop - Th...
Create a simple API - L...
https://apim.ibmcloud.com/apimanager/#/author/apis/56d84f5f0cf2a46163be0514/resources/56d84f7c0cf2a46163be051f
Search
IBM
Environment
Sandbox
Plan
Loopback (Revision 1)
Invoke
Response
Debug
2:13 AM
3/4/2016

200 OK

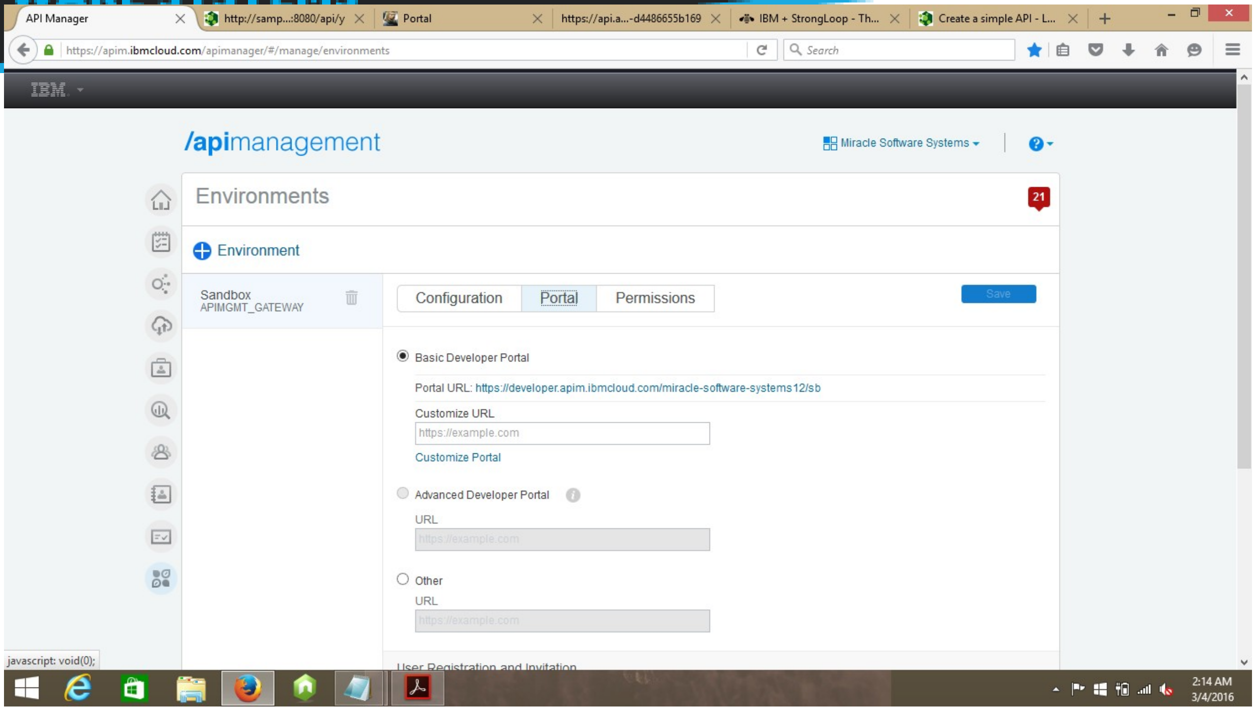
```
content-type: application/json; charset=utf-8
X-Global-Transaction-ID: 43928949
etag: W/"26-KgXBK7bm9McED+LaMn3s+A"
X-RateLimit-Remaining: 1
X-RateLimit-Limit: 2
APIm-Debug-Filename: temporary:///apimdebugd45f4.json
x-download-options: noopen
access-control-allow-credentials: true
date: Thu, 03 Mar 2016 20:43:00 GMT
Connection: Keep-Alive
vary: Origin, Accept-Encoding
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
Transfer-Encoding: chunked
X-Backside-Transport: OK OK
APIm-Debug-Trans-Id: 10.77.164.33-e17d32e0-1980-44c2-9aff-39b5d385b36b
X-BACKEND: apps-proxy
```

```
[
        {
                "name": "home",
                "date": "sg",
                "id": "1"
```

Testing:

Now go to Environments􀃆Portal

Use the Portal URL


Environments

Sandbox
APIMGMT_GATEWAY

Configuration | Portal | Permissions

● Basic Developer Portal

Portal URL: https://developer.apim.ibmcloud.com/miracle-software-systems12/sb

Customize URL

https://example.com

Customize Portal

Advanced Developer Portal

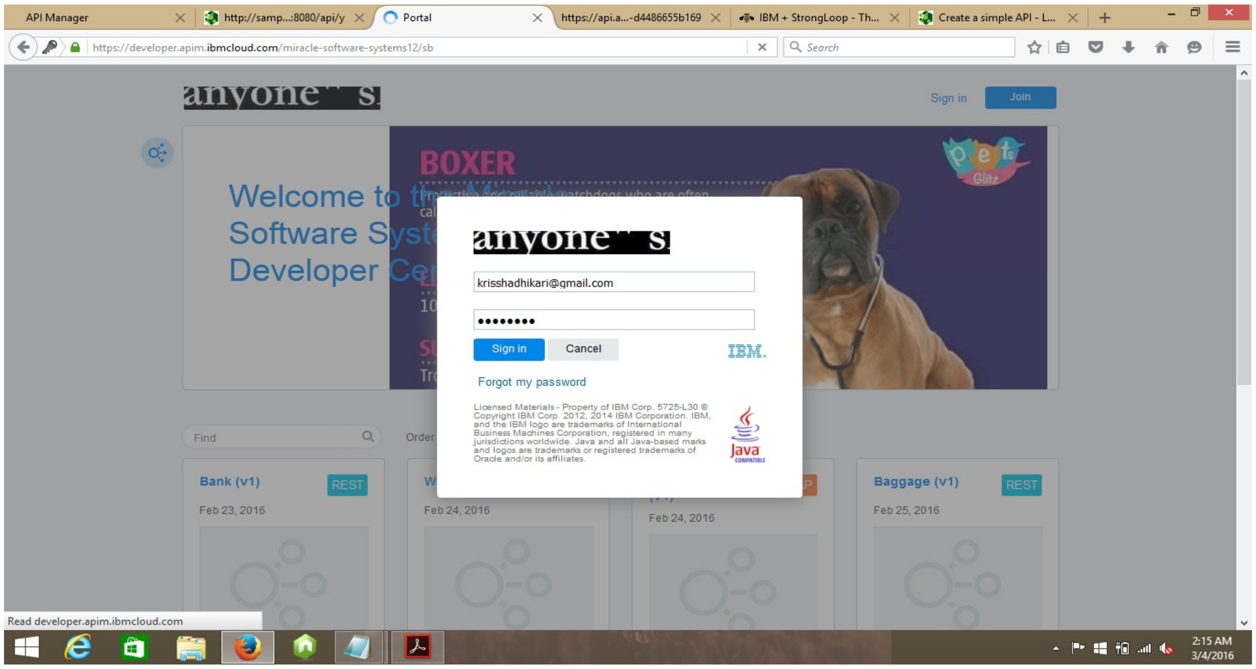
URL

Other

URL

User Registration and Invitation

“Sign in” to the developer portal


krisshadhikari@gmail.com

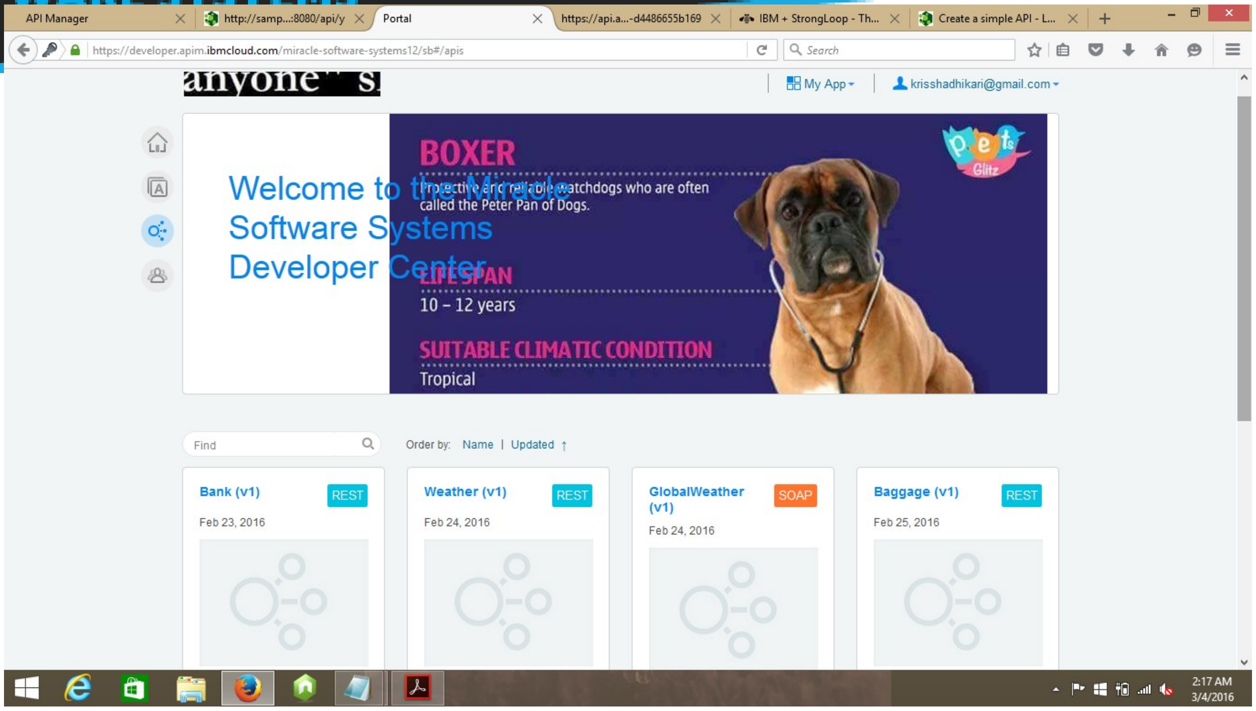
••••••••

Sign in | Cancel

Forgot my password

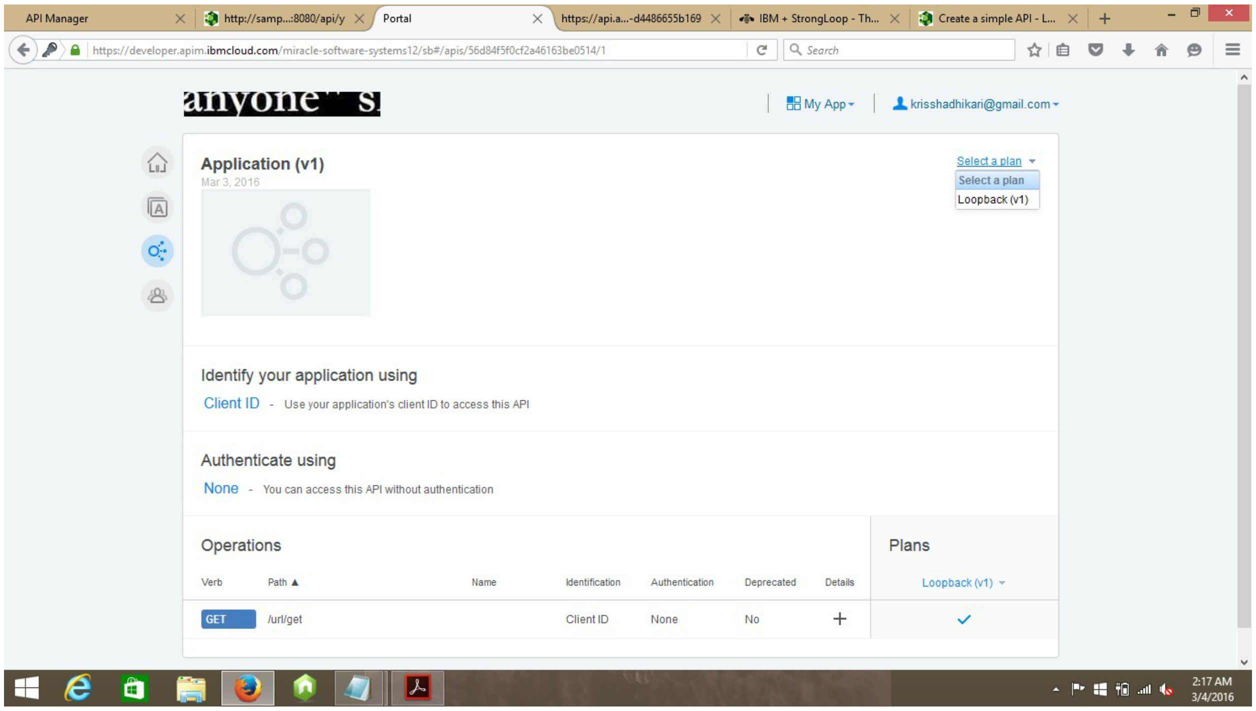
Licensed Materials - Property of IBM Corp. 5725-L30 © Copyright IBM Corp. 2012, 2014 IBM Corporation. IBM, and the IBM logo are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Java and all Java-based marks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Select the application that you have created


API Manager
Portal
https://developer.apim.ibmcloud.com/miracle-software-systems12/sb#/apis
My App
krisshadhikari@gmail.com
Welcome to the Miracle Software Systems Developer Center
Find
Order by: Name | Updated
Bank (v1) REST Feb 23, 2016
Weather (v1) REST Feb 24, 2016
GlobalWeather (v1) SOAP Feb 24, 2016
Baggage (v1) REST Feb 25, 2016

Click on Application icon

Select your application Select plan


https://developer.apim.ibmcloud.com/miracle-software-systems12/sb#/apis/56d84f5f0cf2a46163be0514/1
My App
krisshadhikari@gmail.com
Select a plan
Loopback (v1)

Application (v1)

Mar 3, 2016

Identify your application using

Client ID - Use your application's client ID to access this API

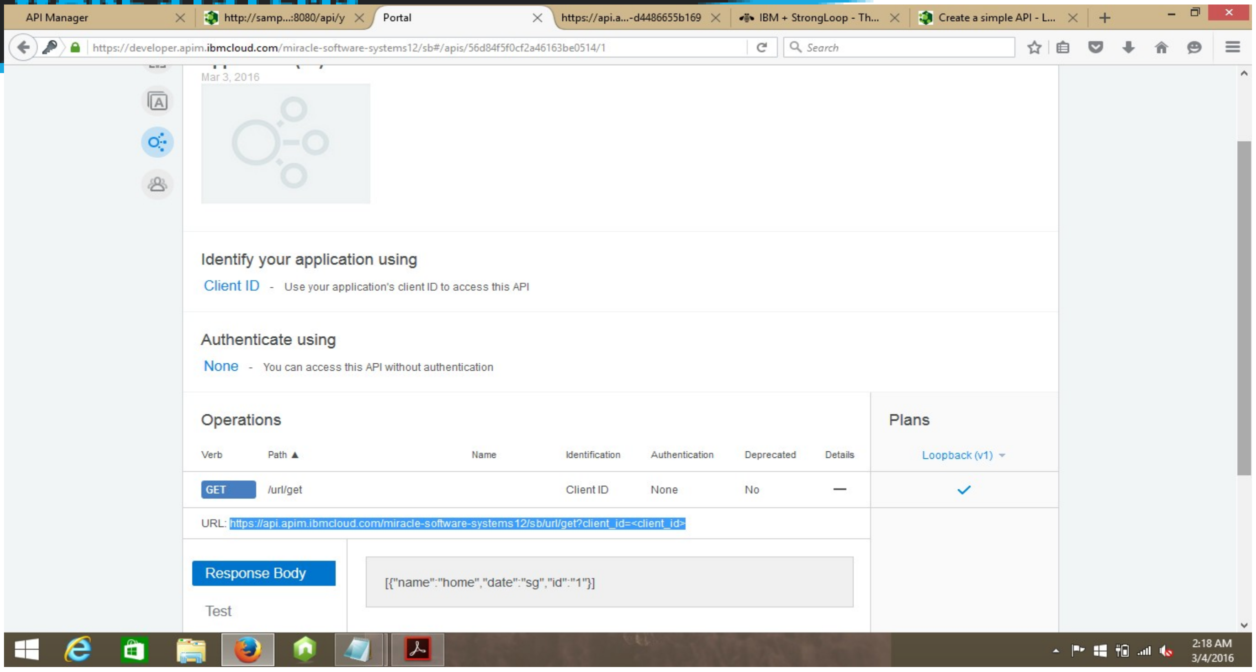
Authenticate using

None - You can access this API without authentication

Operations

Verb	Path ▲	Name	Identification	Authentication	Deprecated	Details	Loopback (v1)
GET	/url/get		Client ID	None	No	+	✓

Click on "+" symbol of details and copy URL provided


Identify your application using

Client ID - Use your application's client ID to access this API

Authenticate using

None - You can access this API without authentication

Operations

Verb	Path	Name	Identification	Authentication	Deprecated	Details	Loopback (v1)
GET	/url/get		Client ID	None	No	—	✓

URL: https://api.apim.ibmcloud.com/miracle-software-systems12/sb/url/get?client_id=<client_id>

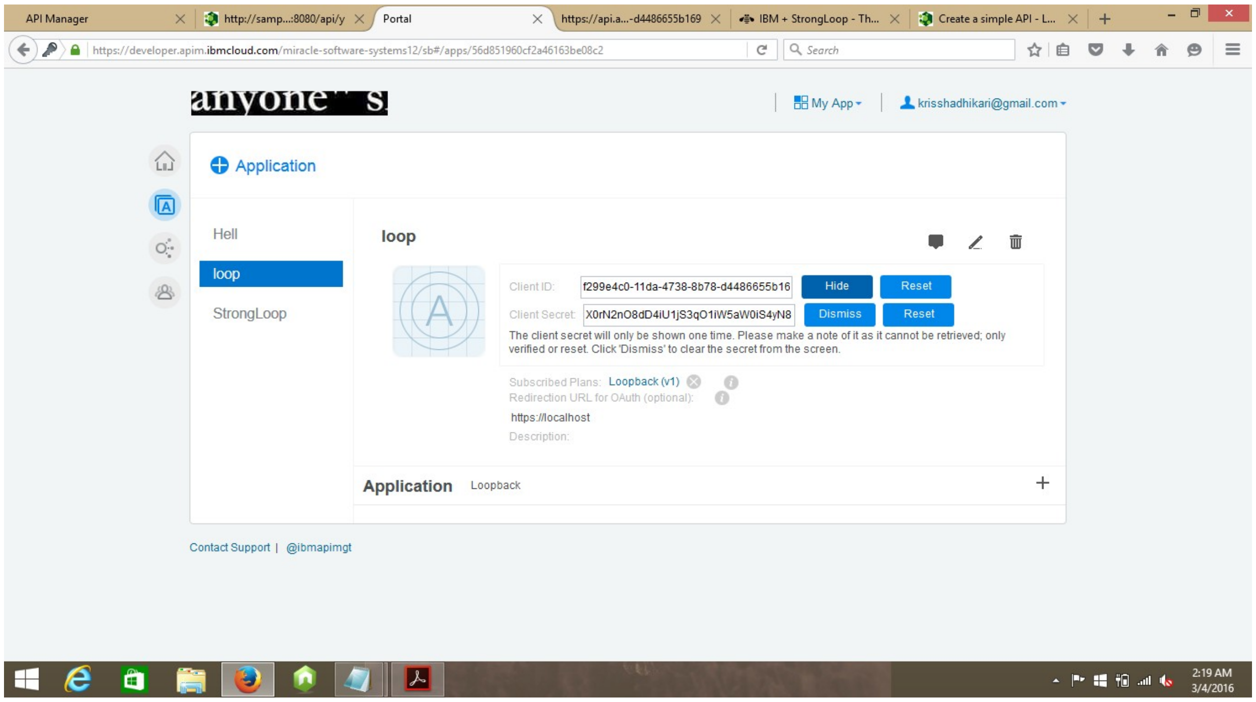
Response Body

```
[{"name":"home","date":"sg","id":"1"}]
```

But for the client Id there, follow the steps below:

Go to the Applications, select the required application.

Click show button of client ID, copy it


loop

Client ID: f299e4c0-11da-4738-8b78-d4486655b16

Client Secret: X0rN2nO8dD4iU1jS3qO1iW5aW0iS4yN8

The client secret will only be shown one time. Please make a note of it as it cannot be retrieved; only verified or reset. Click 'Dismiss' to clear the secret from the screen.

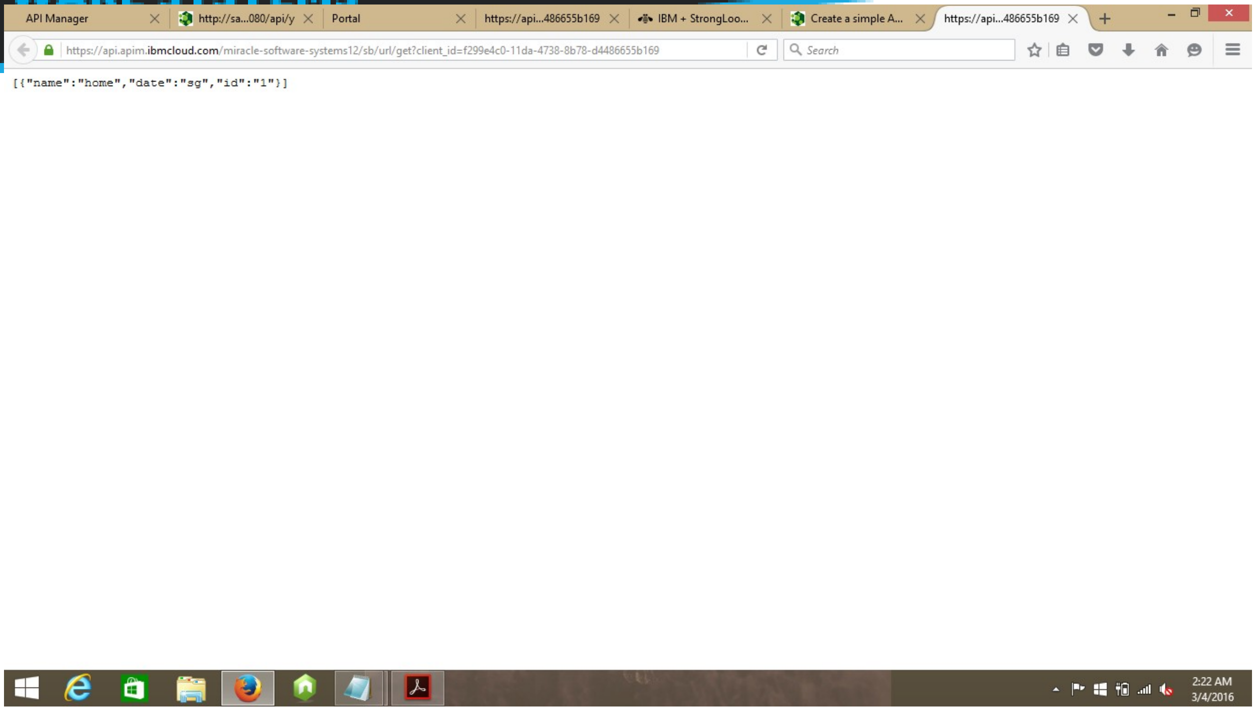
Subscribed Plans: Loopback (v1)

Redirection URL for OAuth (optional):

https://localhost

Description:

Paste the Client ID in the URL and run it.



https://api.apim.ibmcloud.com/miracle-software-systems12/sb/url/get?client_id=f299e4c0-11da-4738-8b78-d4486655b169

```
[{"name":"home","date":"sg","id":"1"}]
```

Reference Links:

Cloud9: **https://c9.io**

StrongLoop:
https://docs.strongloop.com/display/public/LB/Create+a+simple+API