MIRACLE
SOFTWARE SYSTEMS

Dev Workshop | IoT APIs 101 with StrongLoop and IBM Bluemix

API Developer Lab | Miracle Innovation Labs

Chanakya Lokam
Director – Marketing and Innovation
Miracle Software Systems, Inc.

February 14, 2016

Dev Workshop | IoT APIs 101 with StrongLoop and IBM Bluemix

Goal

In this lab the users will create a REST API for the IoT Data that is being stored in Cloudant. We will be using StrongLoop to create the REST API using the Loopback Framework and String Loop CLI Tools.

Pre-Requisites

The following are required to complete this lab,

- Web Browser for accessing Cloudant and Cloud9
- SOAP UI 5.0 (or) Curl for testing the API
- Existing Cloudant Account
- Completion of Part 1 of this lab(IoT Data Stored in Cloudant DB)

Technology Involved

Strong Loop
Node.js
REST/SOAP
Internet of Things
Cloudant and NoSQL

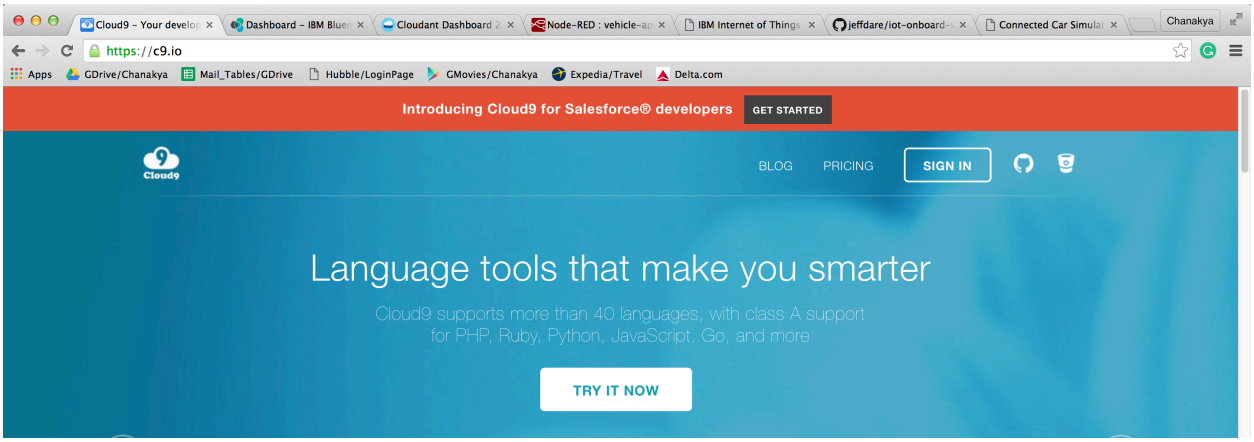
Lab Steps

So, let us get started with the lab!

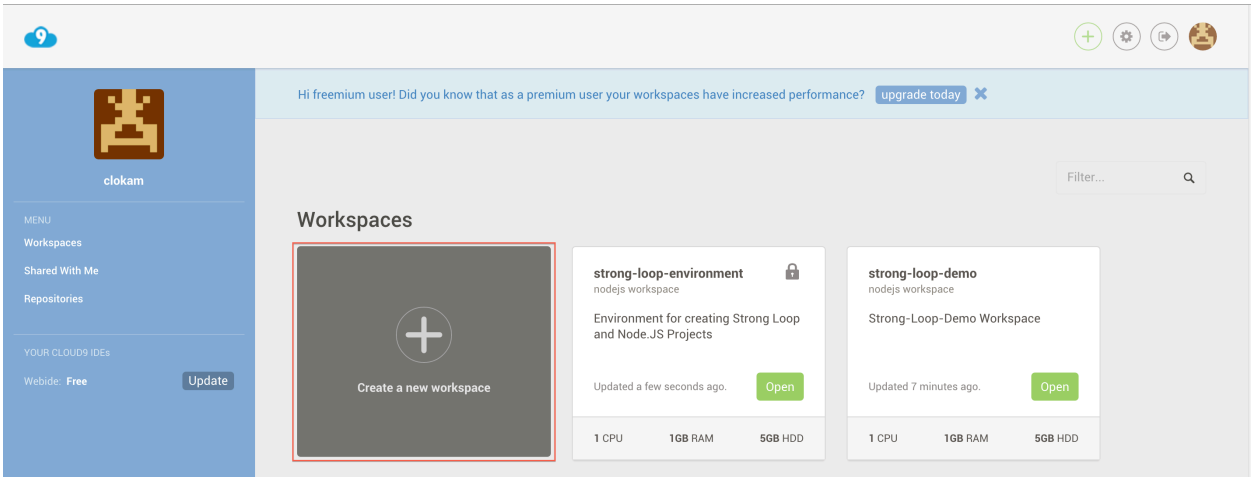
#1 | Get Access to Cloud9

We will be installing and using StrongLoop with Cloud9 which is an online IDE for developers. You are also free to install StrongLoop locally and follow the same steps with slight alterations.

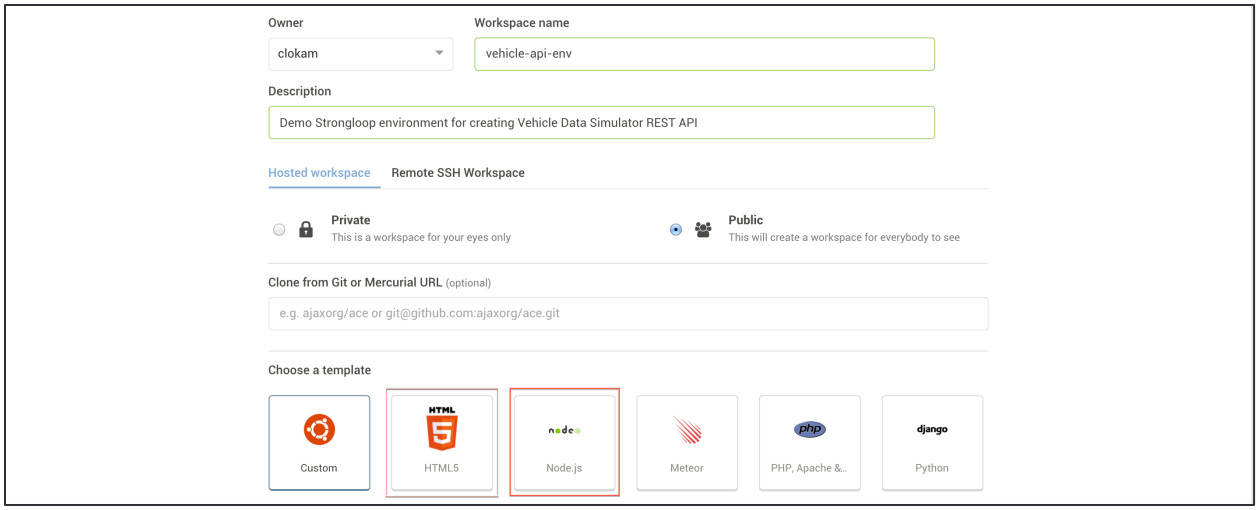
Go to **http://c9.io** and sign in if you already have any account (or) sign up if you are a new user.



Once you are signed in click on the **"Create a new workspace button"**.



Create the new workspace with the workspace name(For example, vehicle-api-env) and select the **node.js template** in the **"Choose a Template"** section.



#2 | Installing StrongLoop

Once you have created the workspace you will be directly taken to the new workspace screen and will have your cursor placed within a new terminal.

In the terminal type the following command to install Strong Loop. The installation may take a few moments.

npm install –g strongloop

After the installation is complete you should be able to see the following screen within your terminal.

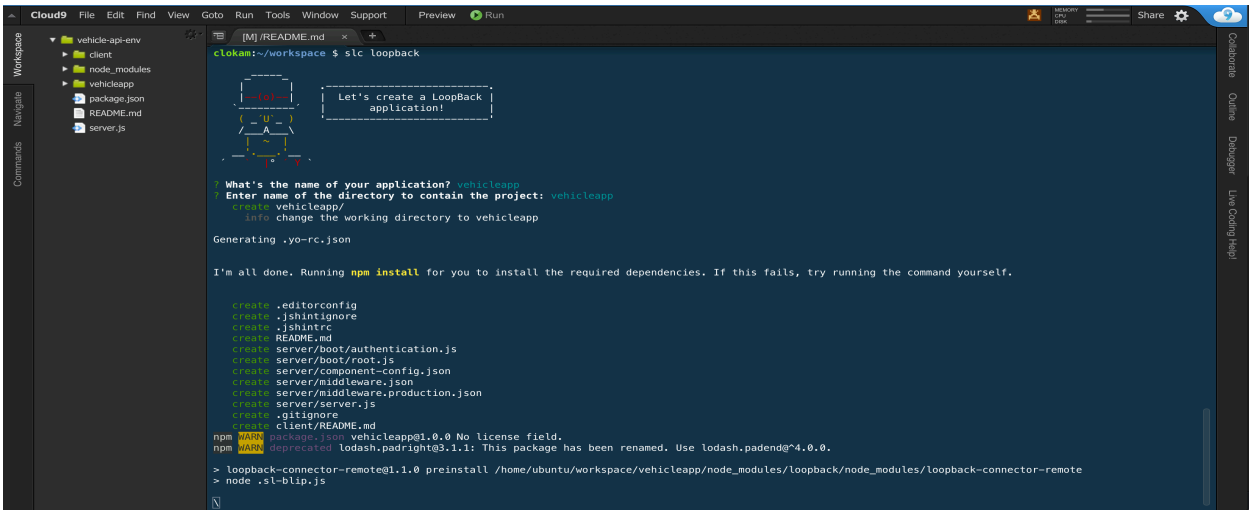
Note : The installation process is dependent on the network and is likely to take between 3-5 minutes to complete

#3 | Create Loopback Application and Model

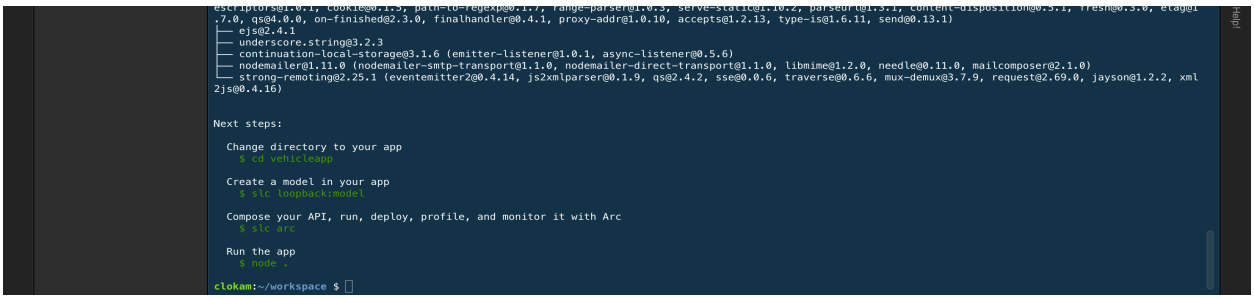
Once StrongLoop is installed we will now need to create the Application, Data Source and the Application Model. This will define how the REST API will be shaped out and what it will connect to.

Use the **slc loopback** command to create the new application. When prompted give the **Application Name**(For example vehicleapp) and the **Application Directory**(For example vehicleapp). This is based on Yeoman and scaffolds your entire application structure to ensure that you can get things running quickly!

NPM will ensure that all the dependencies of the application are installed along with the entire application structure.



Once the installation is complete you should see the next steps as shown below.

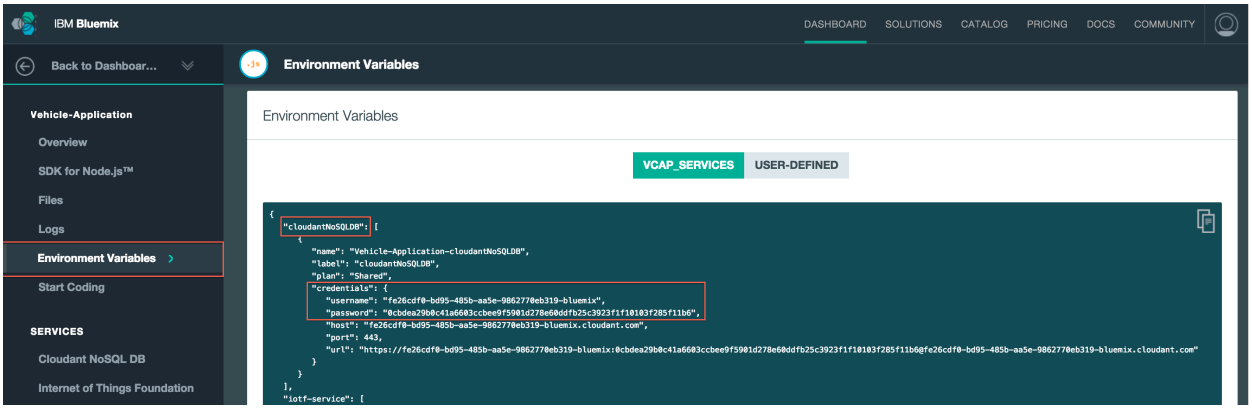


Now we must configure the Data Source and install the connector for using Cloudant with our StrongLoop API.

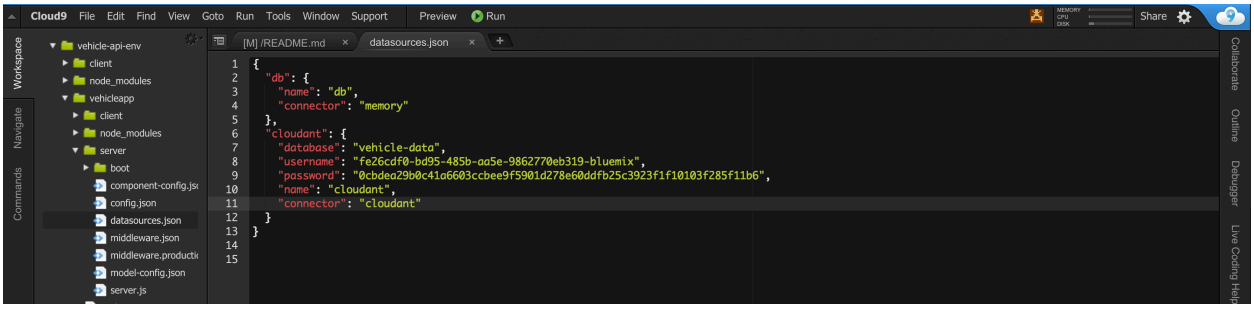
Navigate to the **<directory-name>/server/datasources.json** file using the left side workspace explorer menu. **Remove the existing code and paste the below code.**

```
{
 "db": {
  "name": "db",
  "connector": "memory"
 },
 "cloudant": {
  "database": "vehicle-data",
  "username": "fe26cdf0-bd95-485b-aa5e-9862770eb319-bluemix",
  "password":
"0cbdea29b0c41a6603ccbee9f5901d278e60ddfb25c3923f1f10103f285f11b6",
  "name": "cloudant",
  "connector": "cloudant"
 }
}
```

For getting the **Cloudant User Name** and **Password** you will have to go back to your Bluemix Application and click on the **Environment Variables** option. Here you can copy and paste the credentials.

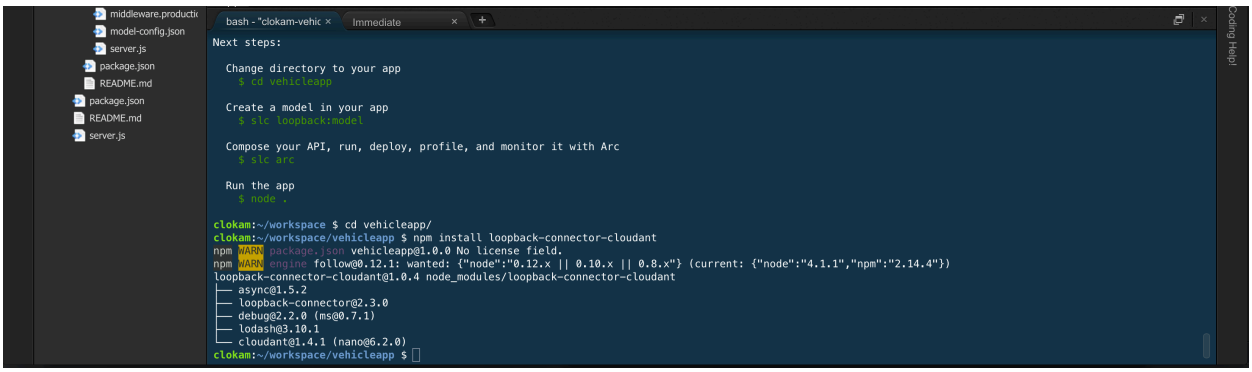


Save the datasources.json file.



Now go back to the terminal, navigate to the application directory by using the command **cd <directory-name>**. Then execute the following command to install the Cloudant DB Connector.

npm install loopback-connector-cloudant

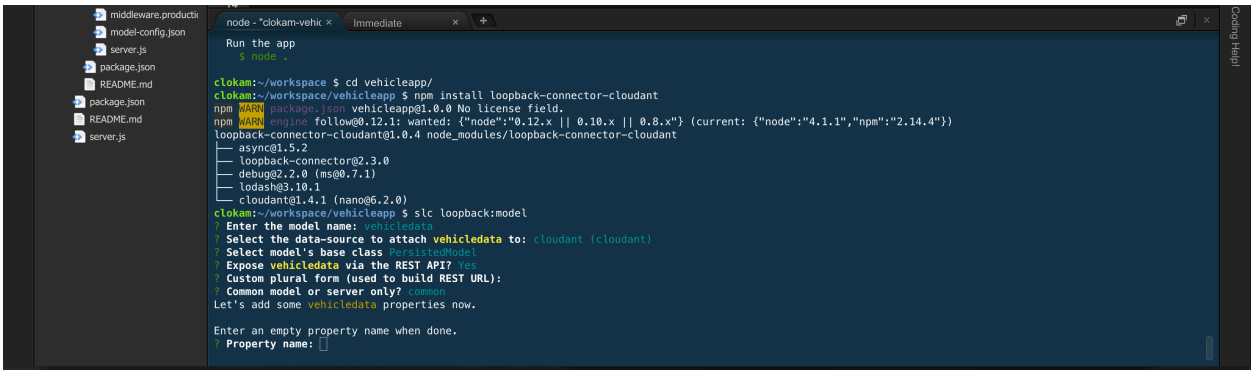


The next step is to create the data model within the application. Stay in the same directory and use the following command.

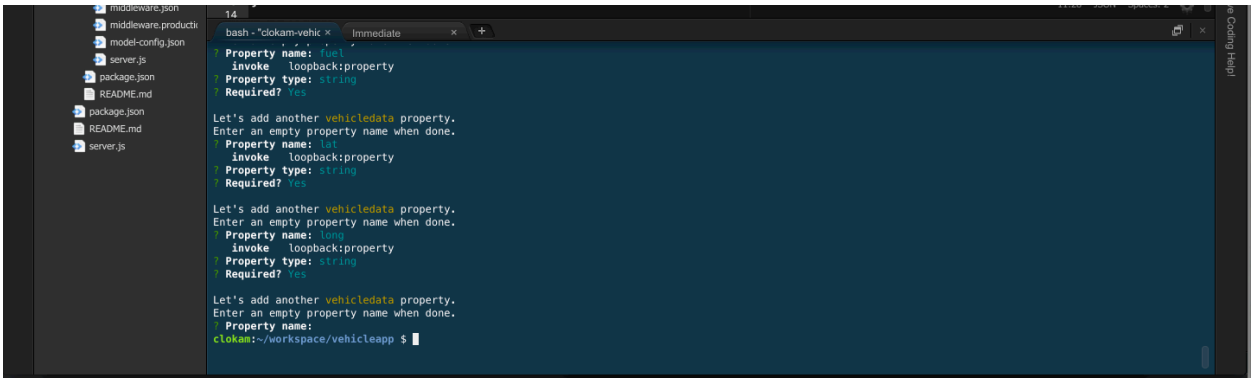
slc looback:model

When prompted give the **model name**(For example vehicledata). Select the data-source as **cloudant** and the model's base class as **PersistedModel** using the arrows keys.

When asked **“Expose vehicledata via REST API?”** answer with **“Y”**. Click enter when asked about the **“Custom Plural Form”**. Select Common Model when asked and then we can go about creating the property names.



Give the first property name as **“fuel”** and select it as **String Type**. Also add two more **String** properties **“lat”** and **“long”**. Select required as **“y”** for all 3 of the properties. Once done click on Enter to finish the Loopback Model creation.



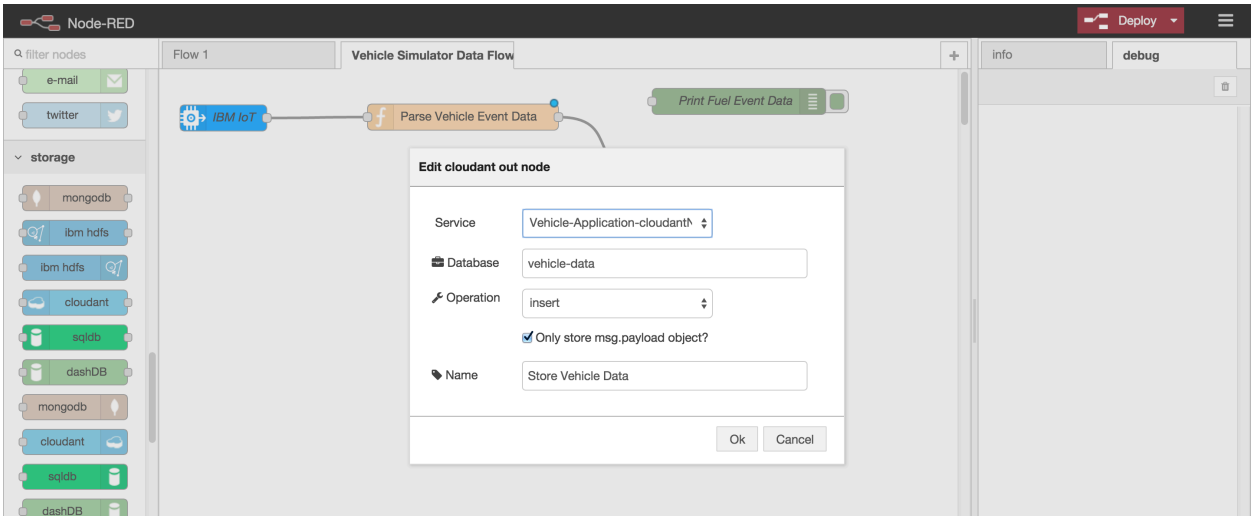
#4 | Editing the Node Red Flow

For the looback model to be able to gather data from the Cloudant DB we have to make sure that we are storing data from Node Red in the same format.

For this we must edit our Node Red Flow. Go back to Node Red and change the code within our **“Parse Vehicle Event Data”** function node to show the below.

msg.fuel = msg.payload.d.fuel
msg.lat = msg.payload.d.lat
msg.long = msg.payload.d.long
msg.loopback_model_name = "**<Your Loopback Model Name>**"
msg.payload = {"fuel":msg.payload.d.fuel,"loopback__model__name":"vehicle-model"}
return msg;

Also go to the Cloudant node and check the **"Only store msg.payload object?"** as well to ensure that we are stroing only our msg.payload object in to Cloudant.



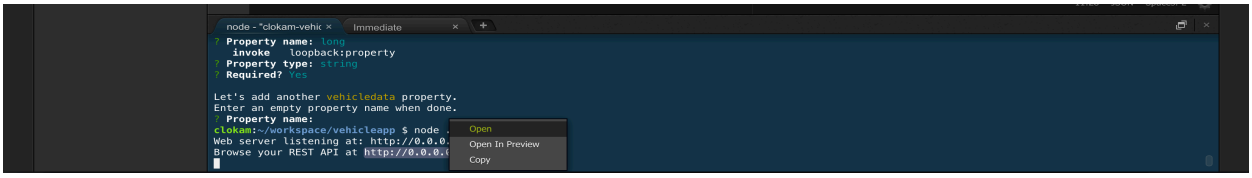
Redeploy the Node Red flow, go back to the Vehicle Simulator and send some more Fuel Events to see if the data gets stored in the Cloudant DB.

#5 | Testing the StrongLoop API

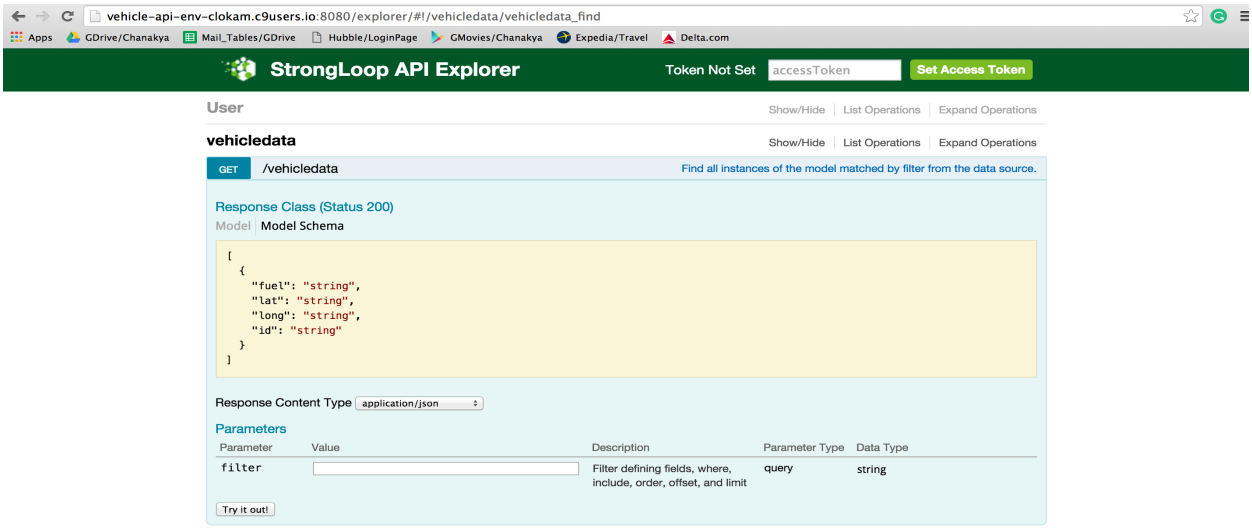
Now that we have everything set, go back to the Cloud9 Terminal within the application directory and run the following command.

node .

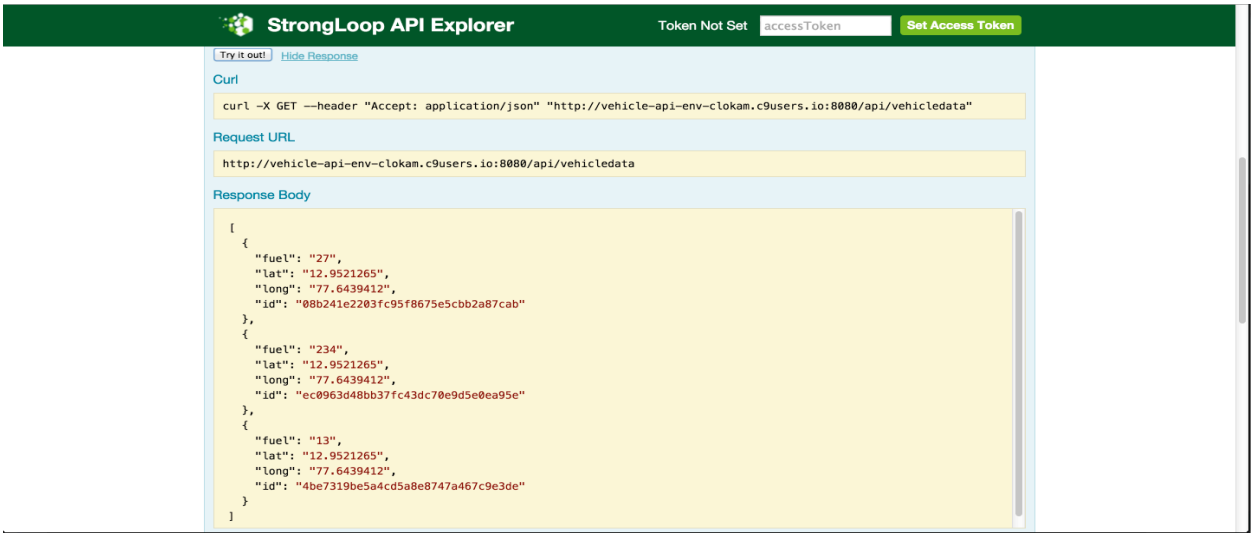
This will start the **explorer** on **localhost(0.0.0.0)** on port **8080**. Click on the link in the terminal and select "Open" to open the explore in a new tab.



In the explorer, click on your loopback model and then on the GET Method to open up the following screen,

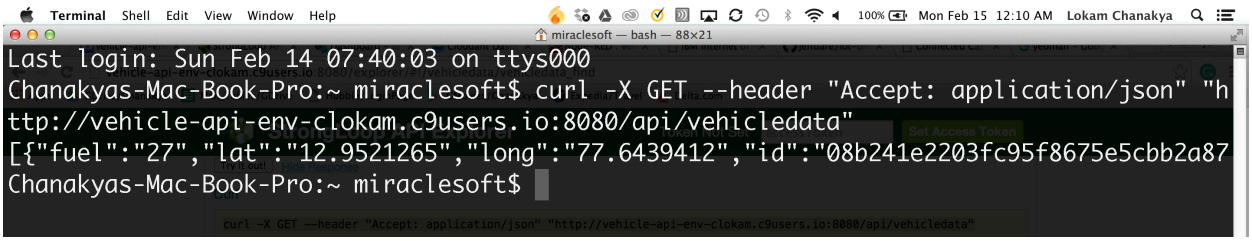


Click on “Try it out!” and you should be able to see the data coming in from the API with the Vehicle Data.



You can also try the same using the **CURL** Option given in the explorer (or) try and send the request using an API testing tool such as **SOAP UI** from SmartBear.

Below is a screenshot of a CURL Test from a local machine terminal.



Now that you have your devices connected, data stored and api exposed you are ready to tackle the world of IoT. Also try out managing your APIs so that you can control which operations you want to expose, who gets access and how much privileges they can extend on the data.