



REST Connector in Strongloop

GeetaKrishna.Adhikari

Miracle Software Systems, Inc.

Date:6/29/2016



REST Connector

Procedure

RESULT



REST Connector: One of the Connectors provided by Strongloop. It is meant to connect a Model and a REST API.

Procedure:

- Create a loopback application using slc loopback
- You can access your REST Connector using the command slc loopback:datasource
- Among the available Datasource Connectors, select the REST Services.

```
lext steps:
Change directory to your app
Create a model in your app
Compose your API, run, deploy, profile, and monitor it with Arc
Run the app
:\Users\miracle>cd restconn
:\Users\miracle\restconn>slc loopback:datasource
Enter the data-source name: restconn
Enter the data-source name: restco
Select the connector for restconn: REST services (supported by StrongLoop)
onnector-specific configuration:
Base URL for the REST service: http://maps.googleapis.com/maps/api/geocode/{format=json}
Default options for the request:
An array of operation templates:
Use default CRUD mapping: Yes
Install loopback-connector-rest@^2.0 (Y/n)
                                                  DIN_
```

- You'll be asked to provide a BASE URL for your REST Service.
- You'll be asked for Configuring Options.
- The REST connector uses the request() module as the HTTP client.
- Here you can configure the Content Type and type of data.
- Next step is to provide the Operation templates.



- These are the details of HTTP method(here GET) and Query parameters you want to use for the REST service.
- But these details should be **Stringified**.
- If you want to perform CRUD operations on the details of REST service using a Model created by you Type **yes** for CRUD operations.
- And Install loopback connector for REST service.
- Now create a model so the CRUD operations of this Model can access the REST Service.

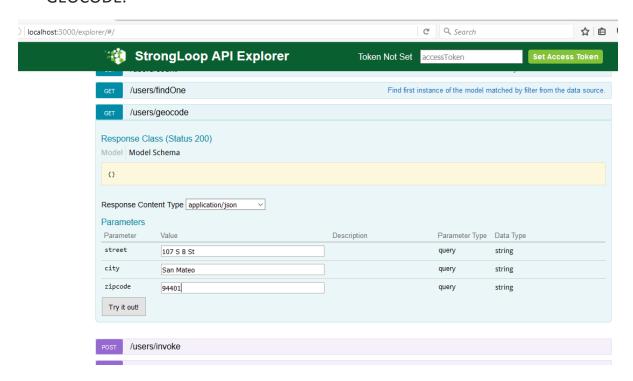
If there is a problem you can start coding Manually:

Go to **server/datasource.json** file of your Lopback and edit(add content) as below.

```
ources.json 🔀 📙 root.js 🖾 📙 server.js 🖾
    "baseURL": "http://maps.googleapis.com/maps/api/geocode/{format=json}",
    "crud": true,
    "connector": "rest",
     "debug": false,
  "options": {
    "headers": {
     "accept": "application/json",
      "content-type": "application/json"
    "strictSSL": false
  "operations": [
      "template": {
        "method": "GET",
        "url": "http://maps.googleapis.com/maps/api/geocode/{format=json}",
          "address": "{street}, {city}, {zipcode}",
          "sensor": "{sensor=false}"
        "options": {
          "strictSSL": true,
         "useQuerystring": true
        "responsePath": "$.results[0].geometry.location"
      "functions": {
        "geocode": ["street", "city", "zipcode"]
```



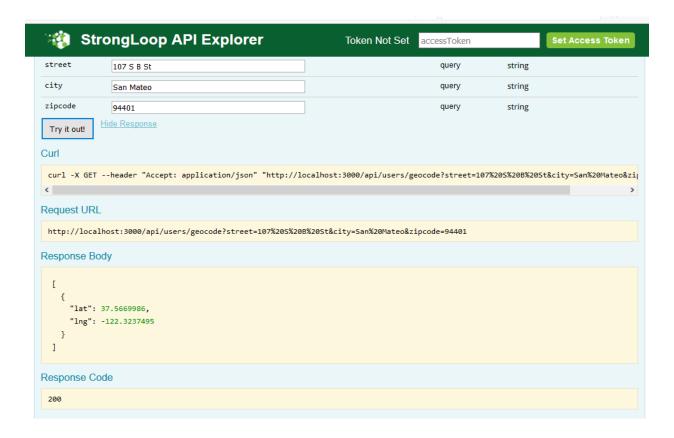
- And setting strictSSL to false allows the application to accept self signed certificates.
- Save the changes.
- Now run the Application using **node**.
- Access the URL at localhost.
- You'll find the Operations of Model in Strongloop Explorer.
- Amon them the Operation that you can really use are the INVOKE and GEOCODE.



- These 2 operations belongs to REST service that you've provided(Base url).
- And the other Operations belongs to Model.
- Provide the Data required into the Query columns and Click on TRY IT OUT.
- You'll get details of Latitude and Longitude.



RESULT:



You can save these details into another Datasource or You can provide RESTRICTED access to users(if you use buit-in USER model).

Coming to INVOKE you need to provide the Data in the form data.

You can find Advanced Documentation at Strongloop:

https://docs.strongloop.com/display/public/LB/REST+connector