

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.

Next steps:

Change directory to your app

```
$ cd restconn
```

Create a model in your app

```
$ slc loopback:model
```

Compose your API, run, deploy, profile, and monitor it with Arc

```
$ slc arc
```

Run the app

```
$ node .
```

```
C:\Users\miracle>cd restconn
```

```
C:\Users\miracle\restconn>slc loopback:datasource
```

```
> Enter the data-source name: restconn
```

```
> Enter the data-source name: restconn
```

```
> Select the connector for restconn: REST services (supported by StrongLoop)
```

```
Connector-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)
```

- 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.

```
sources.json x root.js x server.js x
{
  "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}",
        "query": {
          "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.

The screenshot shows the StrongLoop API Explorer interface. The top bar is green with the StrongLoop logo and the text "StrongLoop API Explorer". To the right of the bar, it says "Token Not Set" and "accessToken" with a "Set Access Token" button. Below the bar, there are two tabs: "GET /users/findOne" and "GET /users/geocode". The "GET /users/geocode" tab is selected. Below the tabs, there is a section for the selected endpoint. It shows "Response Class (Status 200)" and "Model | Model Schema". Below this, there is a yellow box containing the JSON response:

```
{}
```

. Below the yellow box, there is a "Response Content Type" dropdown menu set to "application/json". Below the dropdown menu, there is a "Parameters" section. It contains a table with the following columns: "Parameter", "Value", "Description", "Parameter Type", and "Data Type". The table has three rows: "street" with value "107 S B St", "city" with value "San Mateo", and "zipcode" with value "94401". Below the table, there is a "Try it out!" button. Below the "Parameters" section, there is a "POST /users/invoke" endpoint.

localhost:3000/explorer/#/ StrongLoop API Explorer Token Not Set accessToken Set Access Token

GET /users/findOne Find first instance of the model matched by filter from the data source.

GET /users/geocode

Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
street	107 S B St		query	string
city	San Mateo		query	string
zipcode	94401		query	string

Try it out!

POST /users/invoke

- 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:

The screenshot displays the StrongLoop API Explorer interface. At the top, there's a green header with the StrongLoop logo and the text 'StrongLoop API Explorer'. To the right of the header, it says 'Token Not Set' and 'accessToken' with a 'Set Access Token' button. Below the header, there are three input fields: 'street' with the value '107 S B St', 'city' with 'San Mateo', and 'zipcode' with '94401'. Each field has a 'query' label and a 'string' type indicator. Below these fields, there are two buttons: 'Try it out!' and 'Hide Response'. Under the 'Try it out!' button, there's a 'Curl' section showing a curl command: `curl -X GET --header "Accept: application/json" "http://localhost:3000/api/users/geocode?street=107%20S%20B%20St&city=San%20Mateo&zipcode=94401"`. Below the curl command is a 'Request URL' section showing the URL: `http://localhost:3000/api/users/geocode?street=107%20S%20B%20St&city=San%20Mateo&zipcode=94401`. Below the request URL is a 'Response Body' section showing a JSON array: `[{ "lat": 37.5669986, "lng": -122.3237495 }]`. Below the response body is a 'Response Code' section showing the status code: `200`.

You can save these details into another Datasource or You can provide RESTRICTED access to users(if you use built-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>