**REST / JSON web service in Java using SPARK framework**

An API represents a contract between server (you) and those who are clients (Consumer) of your data. Breaking this contract will result big issues, and a slew of sad users with mobile apps which no longer work. Documentation is half the battle, and it is very difficult to find programmer who also likes to write.

The project is intended to give brief overview on REST/JSON web service application and its implementation. The project contains implementation of industries related use-cases at smaller scale.

The project solves complex scenarios related REST APIs and data handling without using database complexities. Data will be saved in a global java object and it persists till the server is running. It is easily integrable with database of your choice.

It is an educational tutorial for creating and managing REST/JSON web services in JAVA using SPARK.

Request-methods (GET, POST, PUT etc.) and Response-Code are as per HTTP standards.

Feel free to connect with me for any query/suggestion/improvements.

Navneet Singh : [navneet.nks@gmail.com](mailto:navneet.nks@gmail.com)

Configuration :

Default server URL is: <http://localhost:4567>

Default authentication key is **PASSWORD**

Server URL can be changed by setting port and ip-address. Sample code for the same is commented in Controller class.

* Create new company

It is for creating a new company using Rest API.

*URL* : <http://localhost:4567/create_company>

*Method*: POST

*Headers*: **Authentication: PASSWORD**

**Content-Type: application/json**

*Payload/data/body* : It contains the data in JSON format that supposed to be sent to server.

*Sample data*:

{    
   **"id"**:"101",  
   **"name"**:"MyRewardz.com",  
   **"address"**:"Sector 11",  
   **"city"**:"Noida",  
   **"country"**:"India",  
   **"email"**:"navneet.nks@gmail.com",  
   **"phoneNumber"**:"+911234567890",  
   **"owners"**:[    
      {    
         **"name"**:"Navneet",  
         **"email"**:"navneet@diro.io",  
         **"contactNum"**:"+919876543210"  
      }  
   ]  
}

* Get a list of all companies

It fetches list of all company from server and sends the result back to the client.

*URL* : <http://localhost:4567/get_all_companies>

*Method*: GET

*Headers*: **Authentication: PASSWORD**

**Content-Type: application/json**

* Get details about a company

It fetches list of all company from server and sends the result back to the client.

*URL* : <http://localhost:4567/company_detail/101>

Note: at the end of the url, `101` indicates the id of the company.

*Method*: GET

*Headers*: **Authentication: PASSWORD**

**Content-Type: application/json**

* update a company

*URL* : <http://localhost:4567/update_company/101>

Note: at the end of the url, `101` indicates the id of the company.

*Method*: PUT

*Headers*: **Authentication: PASSWORD**

**Content-Type: application/json**

*Payload/data/body* : It contains the data in JSON format that supposed to be sent to server.

*Sample data*:

{    
   **"id"**:"101",  
   **"name"**:"MyRewardz.com",  
   **"address"**:"Sector 11",  
   **"city"**:"Noida",  
   **"country"**:"India",  
   **"email"**:"navneet.nks@gmail.com",  
   **"phoneNumber"**:"+911234567890",  
   **"owners"**:[    
      {    
         **"name"**:"Navneet",  
         **"email"**:"navneet@diro.io",  
         **"contactNum"**:"+919876543210"  
      }  
   ]  
}

* add beneficial owner(s) of the company

*URL* : <http://localhost:4567/add_owners/101>

Note: at the end of the url, `101` indicates the id of the company.

*Method*: PATCH

*Headers*: **Authentication: PASSWORD**

**Content-Type: application/json**

*Payload/data/body* : It contains the data in JSON format that supposed to be sent to server.

*Sample data*:

[    
   {    
      **"name"**:"Navneet Singh",  
      **"email"**:"navneet@abc.io",  
      **"contactNum"**:"+919876543210"  
   },  
   {    
      **"name"**:"Gabbar Singh",  
      **"email"**:"navneet@abc.com",  
      **"contactNum"**:"+919876543777"  
   }  
]