**Create a service to accept Meter Reads – Version 2.0**

Bruno Stuelp Filho – [bstuelp@gmail.com](mailto:bstuelp@gmail.com)

In this version, I used more Frameworks such as Hibernate to persistence, Jackson to parse json to Java Object and Maven to manager the dependencies.

In this project, I consider any Json as insert data. The second endpoint, select data in database by ID and show data.

I change Json to attend my implementation.

1. Technologies
   1. Java + JAX-RS + Jersey + Jackson
   2. Hibernate
   3. Gson
   4. Mysql/Workbench
   5. Tomcat 9
   6. Postman
2. Project Structure

package com.bluetel.entity;

Entities Customer and Read

package com.bluetel.rest;

Rest Services Methods and Interface

package com.bluetel.testJUnit;

JUnit tests

1. URLs

POST

http://localhost:8080/Bluetel/rest/meter-read/store

* Creates a new reader

Request Body: application/json

{

"serialNumber": "27263927192",

"mpxn": "14582749",

"read": [

{"type": "ANYTIME", "registerId": "387373", "value": "2729"},

{"type": "NIGHT", "registerId": "387373", "value": "2892"}

],

"readDate": "2017-11-20T16:19:48+00:00Z"

}

Responses

* 200 – the register was recorded

Response body: application/json: “Stored!”

* 400 – Json invalid or Parse error

GET

http://localhost:8080/Bluetel/rest/meter-read/present/{ID}

Show user meter stored

Parameters

* Request Body -> customerID

Responses

* 200 – registers

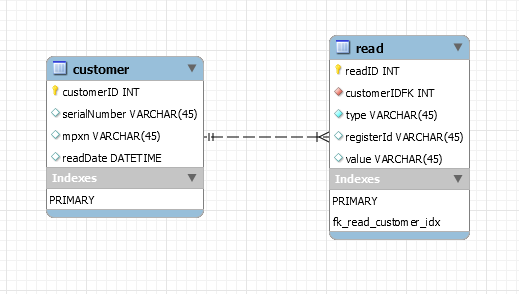
Response body: application/json: A reader

* 400 – ID not found

Examples

* http://localhost:8080/Bluetel/rest/meter-read/present/1 -> (200 – A reader)
* http://localhost:8080/Bluetel/rest/meter-read/present/12 -> (400 - ID not found)

1. Database Diagram **(Generated by Hibernate)**

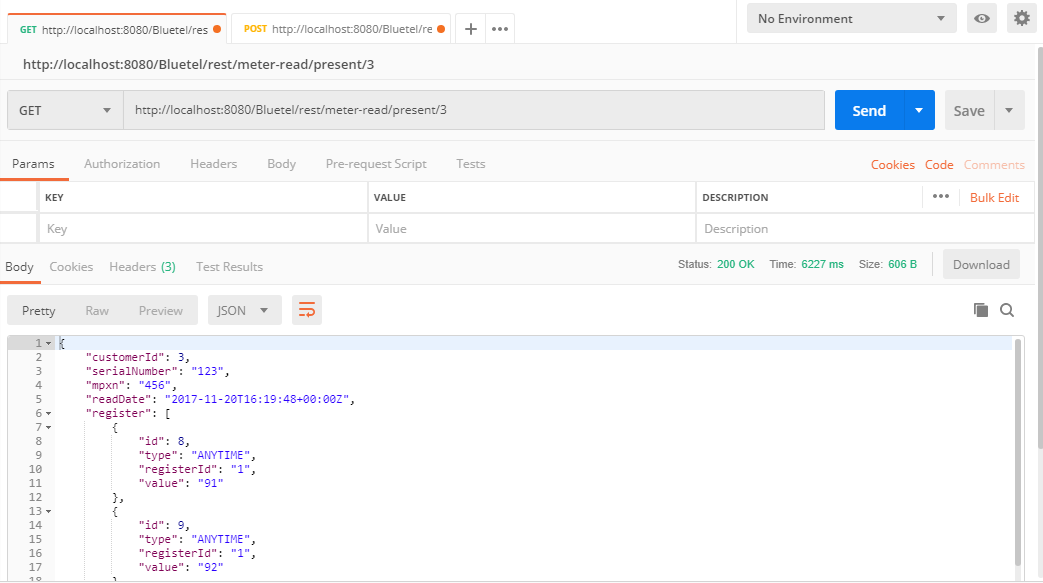


1. Project Decisions

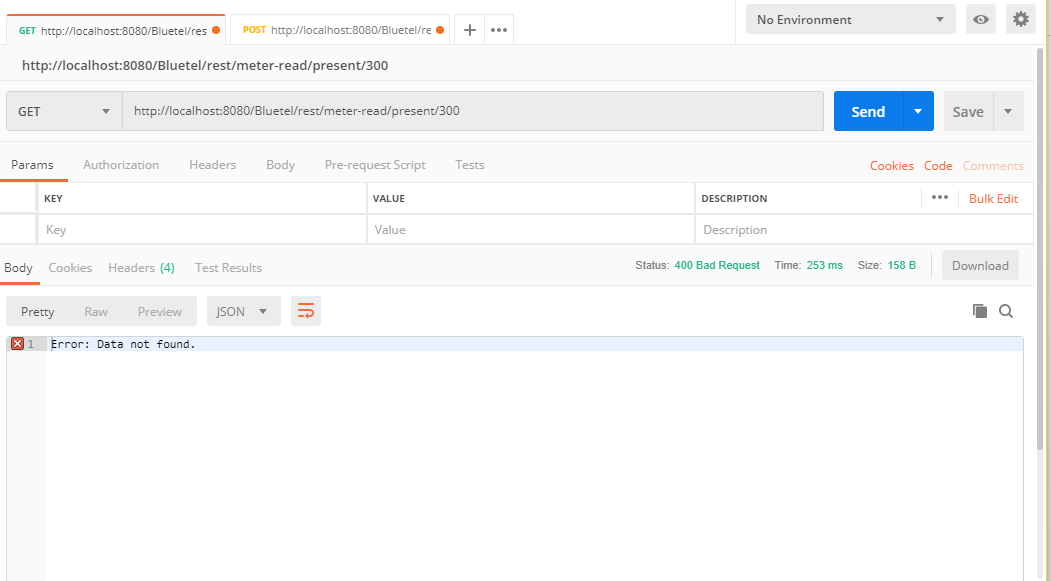
I had some questions about description project. To avoid many question about, I have decided to made some implementations follow my understand.

1. Any Json is inserted;
2. Parse Json to Java Object: **public** Customer parseJsonJavaObject(String strJson);
3. Json validation is done at **public** **boolean** validJson(String json)
4. Tests API

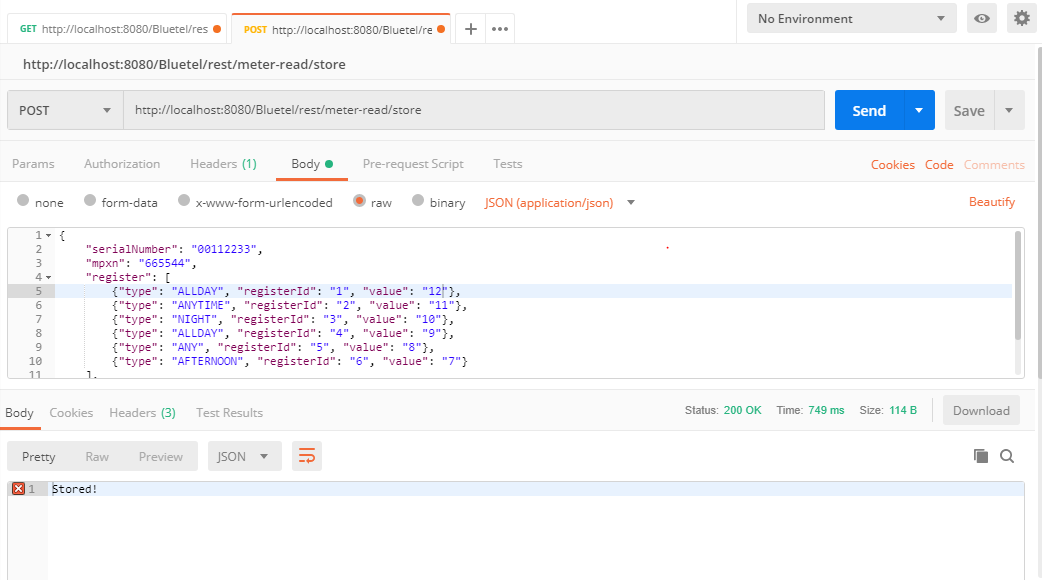
GET - http://localhost:8080/Bluetel/rest/meter-read/present/3



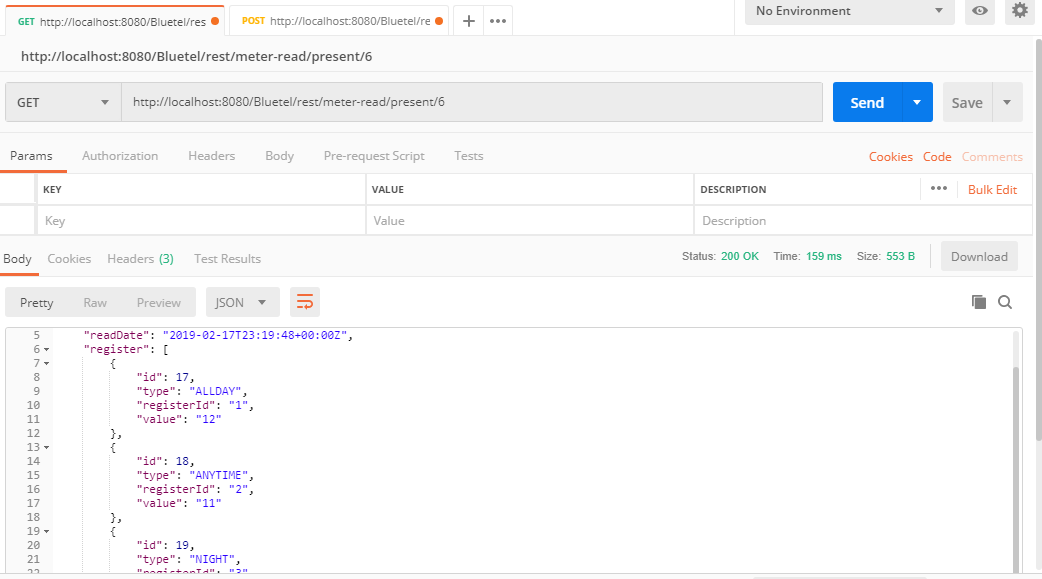
GET - <http://localhost:8080/Bluetel/rest/meter-read/present/300> - ERROR



POST - http://localhost:8080/Bluetel/rest/meter-read/store



http://localhost:8080/Bluetel/rest/meter-read/present/6



1. UNIT TEST

