Spring Boot REST Translator

Users Guide

The *REST Translator* is a tool that can be used as a lookup table. Any string can be used as an input value, and when it is presented as a URL path parameter, a JSON payload will be returned with its assigned translation value. The translation table can be modified using the REST API as well.

Dependencies:

- Java 17 or later

- Maven 3.3 or later

- MySQL 8.0 or later

- IntelliJ is optional, but recommended

- Postman or any REST client application

Prerequisites:

1. Download and install Java. Follow the instructions to setup environment variables and verify your installation.

2. Download and install Maven. As with Java, follow instructions and verify that it works.

3. Download and install MySQL. Create an account and use it to login to the command line or Workspace client. Run the command, "Create database translator" to make the db.

4. In the src/main/resources directory, find and update the *application.properties* file to include your database login credentials.

5. At the top directory, type "mvn clean install" to build the jarfile. If successful, it will create the file translator-RELEASE-3.2.jar in the target directory.

6. From the target directory, run the application with the command "java –jar translator-RELEASE-3.2.jar." Using your REST client, perform any of the commands below.

RestAPI definition:

GET   View Lookup Table   {hostname}:8080/translator   returns JSON:

```json
[
    {
        "id": 1,
        "input": "0001",
        "output": "LOOKUP_CODE_0001"
    },
    {
        "id": 2,
        "input": "0002",
        "output": "LOOKUP_CODE_0002"
    },
    {
        "id": 3,
        "input": "0003",
        "output": "LOOKUP_CODE_0003"
    }
]
```

GET   View Lookup Table (by ID)   {hostname}:8080/translator/id/{id}   Returns JSON:

```json
{
    "id": 1,
    "input": "0001",
    "output": "LOOKUP_CODE_0001"
}
```

GET   View Lookup Table (by input value)   {hostname}:8080/translator/{value}   Returns JSON:

```json
{
    "id": 1,
    "input": "0001",
    "output": "LOOKUP_CODE_0001"
}
```

DEL   Delete Translation Entry   {hostname}:8080/translator/delete/{id}   Returns JSON:

```json
{}
```

RestAPI definition (continued):

POST   Create Translation Entry   {hostname}:8080/translator   JSON body:

```
{
    "input": "0004",
    "output": "LOOKUP_CODE_0004"
}
```

Returns JSON:

```
{
    "id": 4,
    "input": "0004",
    "output": "LOOKUP_CODE_0004"
}
```

PUT   Update Translation Entry   {hostname}:8080/translator   JSON body:

```
{
    "id": 4,
    "input": "0004",
    "output": "NEW_CODE_4"
}
```

Returns JSON:

```
{
    "id":"4",
    "input":"0004",
    "output":"NEW_CODE_4"
}
```