Documented by Rujwi Deshmukh

Text Translation API Documentation

Prerequisites

The developer must have:

- Node.js installed on your computer.
- MySql Database is required to connect to database on local computer.
- A basic understanding of Node.js, javascript, and MySql.
- A code editor. I have used Visual studio code.
- A browser to test the application routes.

Initialize the application

We will initialize the application by running the following command:

>npm init

Installing the required dependencies

Next, we will install the required dependencies.

We need to install the google-translate-api API dependency for translating text supplied. Run the following command:

- **npm install google-translate –save**
- **npm install**

Import the dependencies

In the root folder of the application, create a new file named index.jsand adding following sentences means we are importing dependencies :

const translate = require("translate-google");// import google translator

const sequelize = require("../config/Dbconnections");// sequelize the database means the table is created automatically as per model into the database

Routes:

- **It is the post request made to the server .**

route.post("/translatetext", getTranslationResponse);

Translating the fetched text:

If the text is found in database :

```
sequelize
    .query(
      "CALL getTranslatedResponse(:fromLanguage ,:toLanguage, :text) ",
      {
        replacements: { fromLanguage: from, toLanguage: to, text: text },
      }
    )
```

here, getTranslatedResponse is a MySql stored procedure fetches the result of the translated text.

If the text is new then store it into the database:(using MySql stored procedure

```
 ) DBconnections.query(
          "CALL addTranslatedResponse(:fromLanguage, :toLanguage, :text )",
          {
            replacements: {
              fromLanguage: from,
              toLanguage: to,
              text: text,
              translatedText: data,
            },
          }
        )
```

here, addTranslatedResponse MSSql stored procedure is used to store the data.

The procedure scripts used in this project:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `getTranslatedResponse`(
IN fromLanguage char(200),
IN toLanguage char(200),
IN text TEXT
)
BEGIN
SELECT *
FROM translation.translations
WHERE fromLanguage = fromLanguage
AND toLanguage = toLanguage
AND textContent = textContent;
END
----------------------------------------------.
CREATE DEFINER=`root`@`localhost` PROCEDURE `addTranslatedResponse`(
IN fromLanguage int,
 IN toLanguage varchar(20),
 IN text Varchar(20),
 IN translatedText Varchar(20)
)
BEGIN
        insert into Translators(fromLanguage, toLanguage, text, translatedText) values
(fromLanguage,toLanguage, text,translatedText);
END
-------------------------------------
```

Running Project:

To run this backend use following command:

>npm run start

Testing the API:-

I have used **Postman**(API Tester) for testing the created the API for text translation.