Translator API Documentation  
ITIS 6177: Systems Integration  
Jason Ellerbeck  
15 December 2022

1. **Use Case**

This API is an application that can translate any English text to a desired language. As it stands, this application can only translate from English to a select number of languages that Azure Microsoft Services has implemented. A list of languages that are accepted for this application is one of the main endpoints to show the user what information is needed for a desired language. Created using Node.js, this application leverages the Express and Axios packages to create a functioning translator website.

To start, I created an account with Azure Microsoft Services to access their cognitive services. Within these services I was able to select a Translator resource to configure. This configuration resulted it supplying me with endpoints requests should be made on, parameters of the requests, and most importantly, an API key. With this information I was able to create a JavaScript program and make requests using Axios. After setting up a front-end and successfully translating English strings into a desired language, I was able to launch the application using Digital Ocean. The following sections provide documentation on the endpoints of this applications and provides examples.

1. **Endpoints**

Base URL: <https://translator-final-958cj.ondigitalocean.app/>

GET - /text

Displays a textbox and drop-down menu for the user to input the text they would like to translate and select a language to translate to.

POST - /text

This post request uses the user information from the GET request on the /text to call the Azure Translator API.

Parameters:

resText: (String)

* The input string the user entered of the English text that they would like to translate

language: (String)

* The language code of the language that the result will be translated to.

Response:

Sends a JSON object containing a result string of the translated input string. This object also contains the language code, ‘to’, to display which language it is translated to.

Errors:

400: Bad Request

* This error is displayed when the request to the Azure Translator API fails. This most commonly happens when the to variable of a request is not a valid code for translations.

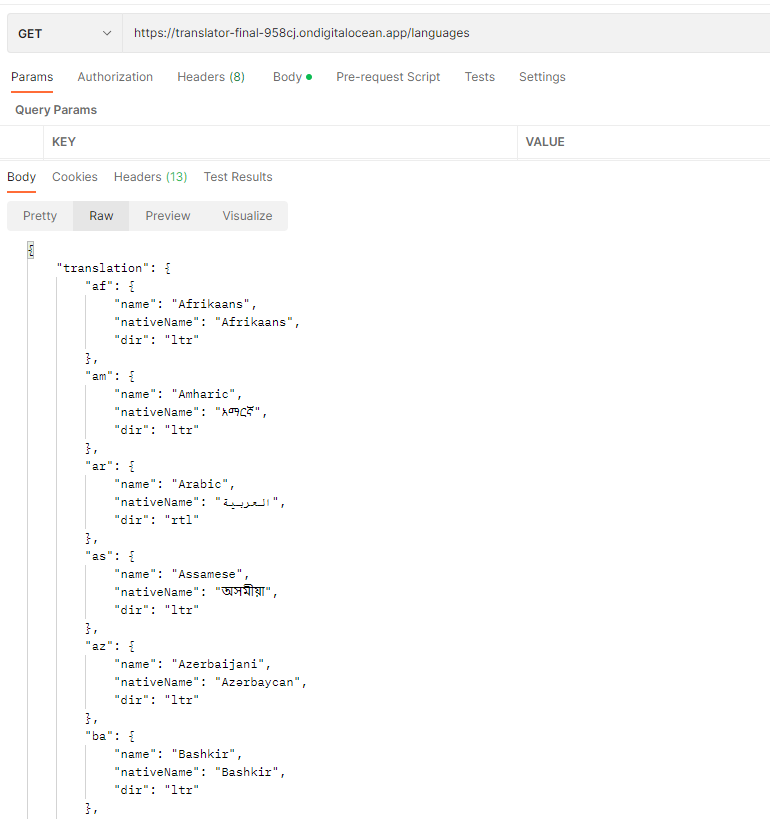
GET - /languages

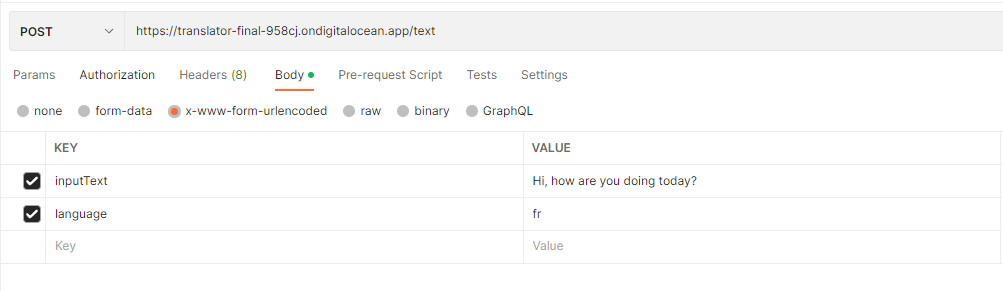
This gets request to the /languages endpoint sends a JSON data structure to the user containing available language names and language codes for the text-translator call to use.

Response:

JSON object containing language names and language codes.

1. **Examples using Postman**
   1. Obtaining language codes



* 1. Translating English to French
     1. Setting request body
     2. Response

