Azure Translation API Documentation

Created by: Shreyash Thakre

Contents

1 Introduction

Welcome to the Translation API documentation. It enables you to translate text into several languages using the Azure Cognitive Services API. It also supports single or multiple translations, language detection, and the list of supported languages. You will find all details on how to effectively make use of the API as you go through the requests examples, response formats, error handling, among others.
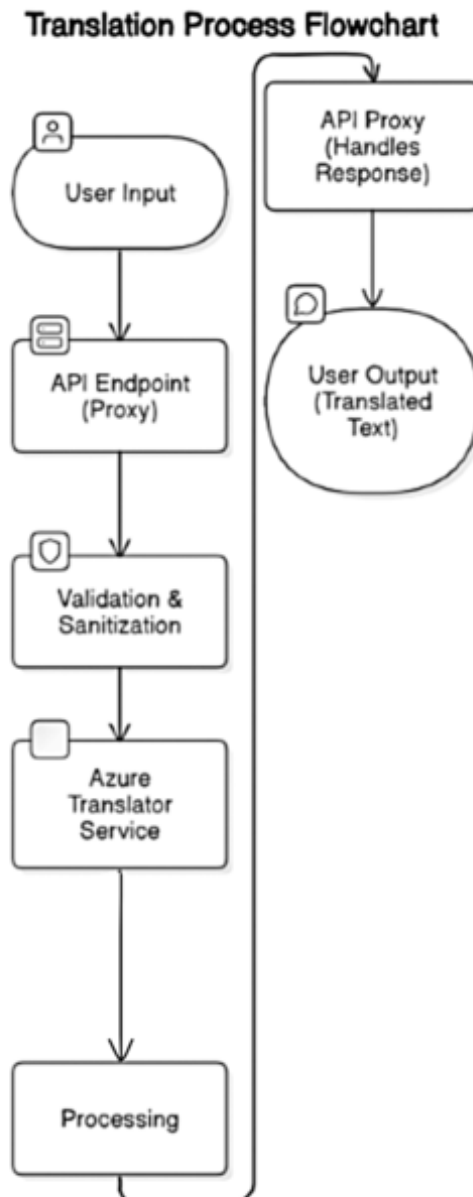

Figure 1: Overview of the Translation API Features

2   Table of Contents

3   API Overview

The Translation API helps to detect and translate text between languages.

- Translate a single text.

- Translate multiple texts at once.

- Detect the language of the provided text.

- Get a list of supported languages.

4   Authentication

This API uses API keys for authentication. You need an Azure API key to use the service.
You should pass the API key in the request headers, as shown:

```
Ocp-Apim-Subscription-Key: <your-api-key>
```

```
Ocp-Apim-Subscription-Region: <your-region>
```

```
Content-Type: <set-content-type>
```

```
try {
  const response = await axios.post(
    `${process.env.AZURE_TRANSLATOR_ENDPOINT}/translate?api-version=3.0&to=${toLanguage}`,
    [{ Text: text }],
    {
      headers: {
        'Ocp-Apim-Subscription-Key': process.env.AZURE_TRANSLATOR_KEY,
        'Ocp-Apim-Subscription-Region': process.env.AZURE_TRANSLATOR_REGION,
        'Content-Type': 'application/json',
      }
    }
  );
```

Figure 2: Example of Including API Key in Headers

5   Base URL

All API requests are made to the following base URL:

```
http://<your-server-ip>:<port>/api/v1
```

For example:

```
http://104.131.126.122:3000/api/v1
```

6 Endpoints

6.1 POST /api/v1/translate/translate-single

Description: Translates a single text from one language to another.

Request Body:

```
{
  "text": "Hello",
  "toLanguage": "fr"
}
```

Response Body:

```
{
  "translations": [
    {
      "text": "Bonjour",
      "to": "fr"
    }
  ]
}
```

6.2 POST /api/v1/translate/translate-multiple

Description: Translates multiple texts at once.

Request Body:

```
{
  "texts": ["Hello", "Good morning"],
  "toLanguage": "fr"
}
```

Response Body:

```
{
  "translations": [
    {
      "text": "Bonjour",
      "to": "fr"
    },
    {
      "text": "Bonjour le matin",
      "to": "fr"
    }
  ]
}
```

6.3   POST /api/v1/translate/detect-language

Description: Detects the language of the provided text.
   Request Body:

```
{
  "text": "Hola"
}
```

   Response Body:

```
{
  "language": "es"
}
```

6.4   GET /api/v1/translate/supported-languages

Description: Gets a list of all supported languages.
   Response Body:

```
{
  "languages": [
    "en", "fr", "es", "de", "it", "pt", "nl", "ar", "zh", "ja"
  ]
}
```

7   Request & Response Examples

7.1   Example 1: Translate Single Text

Request:

```
POST /api/v1/translate/translate-single
{
  "text": "Hello",
  "toLanguage": "fr"
}
```

   Response:

```
{
  "translations": [
    {
      "text": "Bonjour",
      "to": "fr"
    }
  ]
}
```
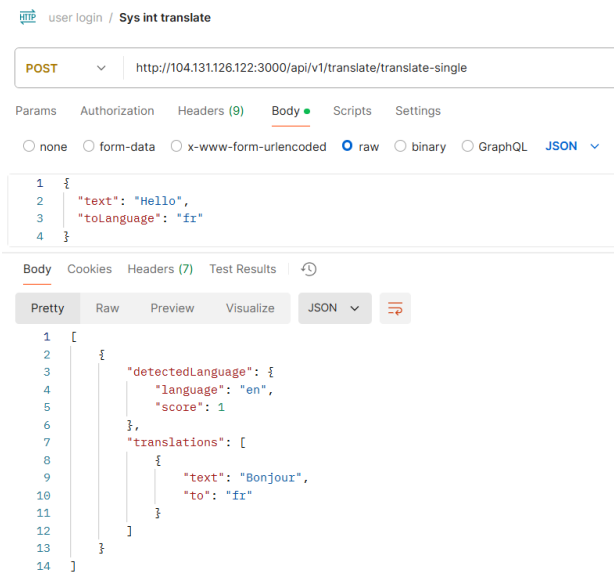
Figure 3: Translate Single Text

7.2 Example 2: Translate Multiple Texts

Request:

```
POST /api/v1/translate/translate-multiple
{
  "texts": ["Hello", "Good morning"],
  "toLanguage": "fr"
}
```

Response:

```
{
  "translations": [
    {
      "text": "Bonjour",
      "to": "fr"
    },
    {
      "text": "Bonjour le matin",
      "to": "fr"
    }
  ]
}
```
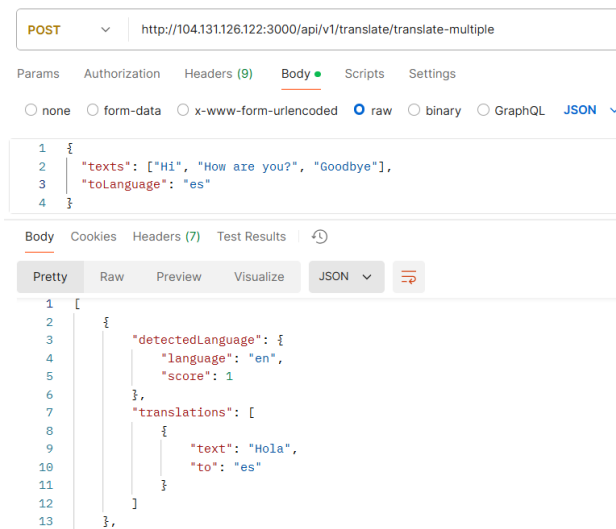
Figure 4: Translate Multiple Texts

7.3   Example 3: Detect Language

Request:

```
POST /api/v1/translate/detect-language
{
  "text": "Bonjour"
}
```

Response:
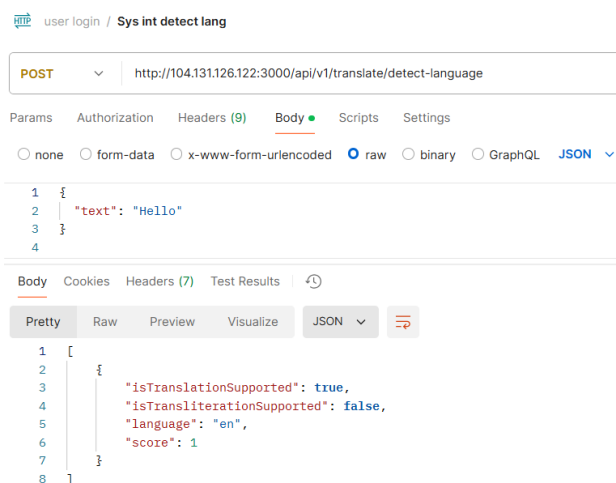
```
{
  "language": "fr"
}
```


Figure 5: Detect Language

7.4 Example 4: Get Supported Languages

Request:

```
GET /api/v1/translate/supported-languages
```

Response:

```json
{
  "languages": [
    "en", "fr", "es", "de", "it", "pt", "nl", "ar", "zh", "ja"
  ]
}
```
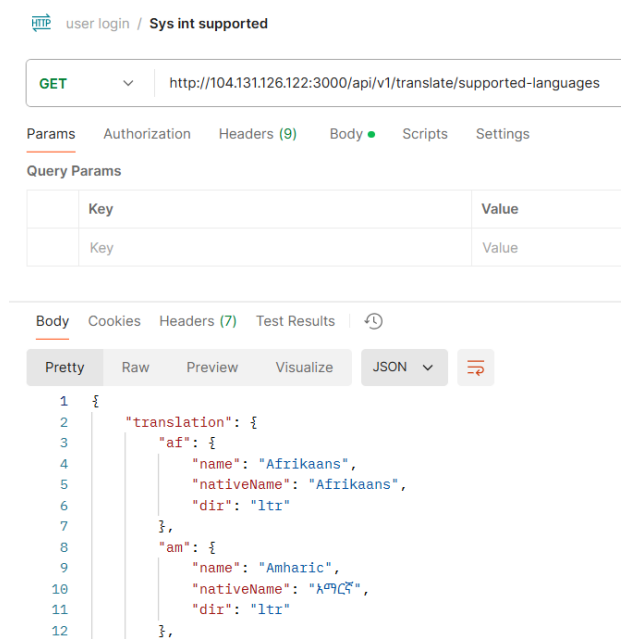

Figure 6: Get Supported Languages

8   Error Handling

This section explains how errors are handled in the API.

8.1   Common Error Responses

- **400 Bad Request:** Invalid or missing parameters.

- **401 Unauthorized:** Invalid API key.

- **404 Not Found:** Endpoint not found.

- **500 Internal Server Error:** Unexpected server error.

8.2   Example Error Response

```
{
  "error": {
    "code": "BadRequest",
    "message": "The 'text' parameter is missing."
  }
}
```

9   Interactive API Testing (Swagger UI)

The API documentation includes an interactive Swagger UI that allows you to test the API endpoints directly in your browser. Swagger UI provides a user-friendly interface where you can send requests, view responses, and quickly try out different API functionalities.

You can access the Swagger UI for your API documentation at the following link: http://104.131.126.122:3000/api-docs/.

Once you navigate to the Swagger UI, you will see a list of all available endpoints with detailed information. You can click on any endpoint to expand its details, send requests with specific parameters, and view the response directly in the interface.

This allows you to test various functionalities of the API in real time and troubleshoot any issues.

Example of Using Swagger UI:

- Select the `POST /api/v1/translate/translate-single` endpoint.

- Enter the text `"Hello"` and select the target language `"fr"` (French).

- Click `Execute` to send the request. The response will show the translated text `"Bonjour"`.

Using Swagger UI makes it easier to understand how each API endpoint works, test your API with real-time data, and troubleshoot any issues directly from the documentation interface.

10   Error Codes

- **400:** Bad Request

- **401:** Unauthorized

- **500:** Internal Server Error

11   FAQ

- **Q: How do I obtain my API key?**
  A: You can obtain your API key by signing up for Azure Cognitive Services and creating a Translation resource.

- **Q: Do I Need a Microsoft Account to sign up for Azure?**
  A: You can sign up with either a Microsoft account or a GitHub account.