Azure Translator API Integration Documentation

Welcome to the Azure Translator API Node.js Integration documentation. This setup guide will walk you through setting up, using, and troubleshooting the Azure Translator API in your Node.js application. With code snippets and clear explanations, you'll have everything you need to get started!

Overview

The Azure Translator API allows you to translate text into multiple languages with high accuracy. This guide includes:

- Setup Instructions: How to configure your Node.js app to interact with the API.
- Endpoints: Details of the endpoints and how to use them.
- Error Handling: Learn about common errors and how to resolve them.
- Examples: Code snippets for real-world scenarios.
- Best Practices: Tips to get the most out of the API.

Key Features

- Multi-Language Support: Translate to over 90 languages.
- Real-Time Processing: Get results almost instantly.
- Ease of Integration: Simple setup with Node.js.
- Scalable: Perfect for projects of any size.

Getting Started

1. Prerequisites

- An active Azure account.
- An Azure Cognitive Services Translator resource.
- Node.js installed on your machine.
- A tool like Postman or cURL for testing APIs.

2. Environment Setup

Install required dependencies:

```
> npm install express axios uuid dotenv
```

Create a .env file in your project directory:

```
AZURE_API_KEY=your-azure-api-key

AZURE_ENDPOINT=https://your-resource-name.cognitiveservices.azure.com

AZURE_LOCATION=your-resource-location
```

3. Directory Structure

Here's an example of how your project directory might look:

```
/project-directory
├──── index.js          # Main application file
├──── .env              # Environment variables
├──── package.json      # Node.js dependencies
└──── README.md         # Documentation
```

API Endpoint

POST /azure-api

Request

- URL: `http://104.248.227.148:3000/azure-api`
- Method: POST
- Content-Type: application/json

Request Body

Field	Type	Required	Description
text	string	Yes	Text to translate
from_language	string	No	Language of the input text (default: 'en')
to_translate	string	Yes	Comma-separated list of target languages (e.g., "fr,zu,hi")

Example Request Body:

```
{
  "text": "Hello, how are you?."
  "from_language": "en",
  "to_translate": "fr,zu,hi"
}
```

Response

- Status Code: 200 OK

Example Response:

```
[
  {
    "translations": [
      { "text": "Bonjour, comment ça va?", "to": "fr" },
      { "text": "Sawubona, unjani?", "to": "zu" },
      { "text": "नमस्ते, आप कैसे हैं?", "to": "hi" }
    ]
  }
]
```

Code Example

Node.js Implementation:

```
const express = require("express");
const axios = require("axios");
const { v4: uuidv4 } = require("uuid");
require("dotenv").config();

const app = express();
const PORT = 3000;

const API_KEY = process.env.AZURE_API_KEY;
const ENDPOINT = process.env.AZURE_ENDPOINT;
const LOCATION = process.env.AZURE_LOCATION;

if (!API_KEY || !ENDPOINT || !LOCATION) {
  console.error("Missing environment variables. Please check your .env file.");
  process.exit(1);
}

app.use(express.json());

app.post("/azure-api", async (req, res) => {
  try {
    const { text, from_language = "en", to_translate } = req.body;

    if (!text || !to_translate || !Array.isArray(to_translate)) {
      return res.status(400).json({ error: "Invalid request body." });
    }

    const params = {
      "api-version": "3.0",
      from: from_language,
      to: to_translate,
    };

    console.log("Final URL:", `${ENDPOINT}/translate`, params);
    const response = await axios({
      method: "post",
      baseURL: `${ENDPOINT}/translate`,
      headers: {
        "Ocp-Apim-Subscription-Key": API_KEY,
        "Ocp-Apim-Subscription-Region": LOCATION,
        "Content-type": "application/json",
      },
      params: params,
      data: [{ text }],
    });

    res.json(response.data);
  } catch (error) {
    console.error("Error translating text:", error.message);
    res.status(500).json({
      error: "Failed to fetch data from Azure API",
      message: error.response ? error.response.data : error.message,
    });
  }
});
app.listen(PORT, () => {
  console.log(`Server running at http:// 104.248.227.148:${PORT}`);
});
```

Error Handling

Common Errors

Status Code	Error	Cause	Solution
400	Bad Request	Missing or invalid input parameters	Ensure all required fields are present and valid.
401	Unauthorized	API key is missing or invalid	Verify API key and region in your '.env' file.
404	Resource Not Found	Incorrect endpoint or resource setup	Check endpoint URL and Azure setup
500	Internal Server Report	General server error	Check logs for more details

Testing the API

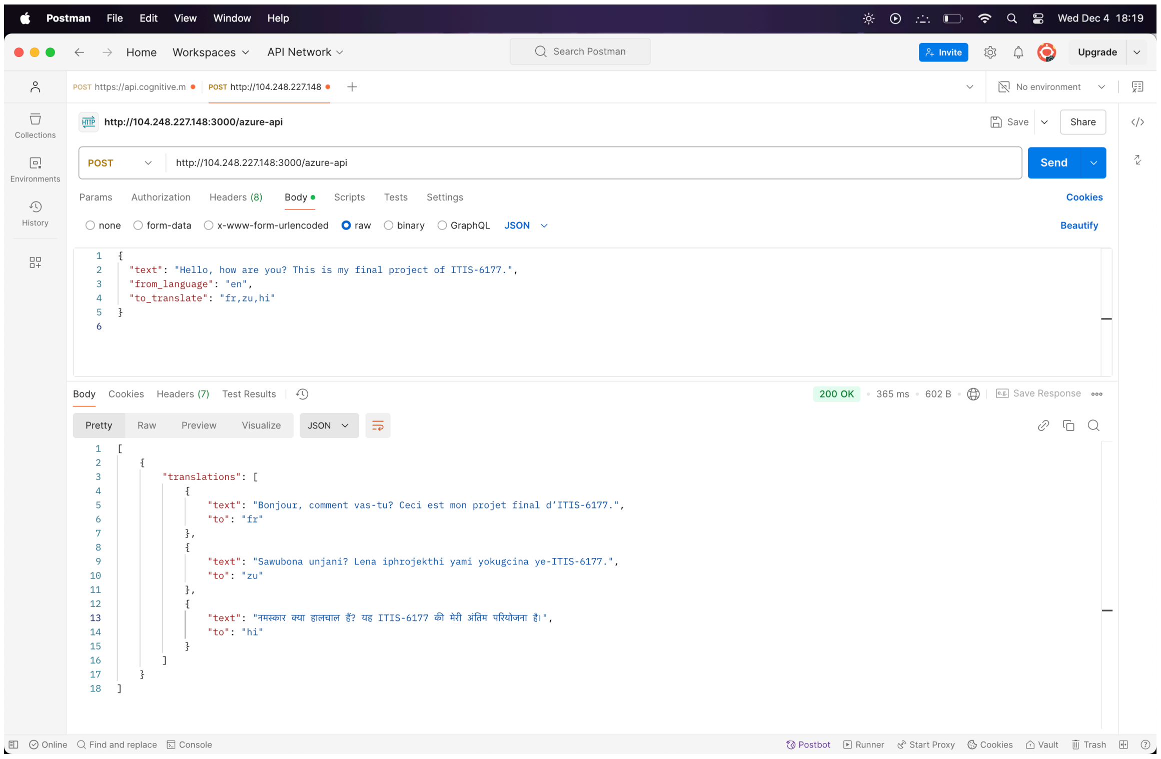
1. Open Postman and create a new POST request.

2. Set the URL to:
   `http://104.248.227.148:3000/azure-api`

3. Add the following JSON body:

```
{
  "text": "Hello, how are you? This is my final project of ITIS-6177.",
  "from_language": "en",
  "to_translate": "fr,zu,hi"
}
```

4. Hit Send and observe the translation results.



POST http://104.248.227.148:3000/azure-api

```
{
  "text": "Hello, how are you? This is my final project of ITIS-6177.",
  "from_language": "en",
  "to_translate": "fr,zu,hi"
}
```

200 OK · 365 ms · 602 B

```
[
    {
        "translations": [
            {
                "text": "Bonjour, comment vas-tu? Ceci est mon projet final d’ITIS-6177.",
                "to": "fr"
            },
            {
                "text": "Sawubona unjani? Lena iphrojekthi yami yokugcina ye-ITIS-6177.",
                "to": "zu"
            },
            {
                "text": "नमस्कार क्या हालचाल हैं? यह ITIS-6177 की मेरी अंतिम परियोजना है।",
                "to": "hi"
            }
        ]
    }
]
```

Best Practices

- Always validate user input to avoid unexpected API errors.
- Secure your API key using environment variables.
- Log API errors for easier debugging.
- Use caching for frequently used translations to reduce API calls.

FAQs

1. How do I find my Azure API Key and Endpoint?

- Go to the Azure Portal.
- Open your Translator resource.
- Navigate to the Keys and Endpoint section.
- Copy the API Key and Endpoint URL from there.

2. Why am I getting a 401 (Unauthorized) error?

This error usually happens due to:

- Missing or incorrect API Key in the .env file.
- Incorrect Endpoint URL.
- Using the wrong LOCATION value.

Double-check your .env file and Azure resource details.

3. How do I translate text into multiple languages?

You can provide an array of target languages in the to_translate parameter. For example:

```
{
  "text": "Hello, how are you?",
  "from_language": "en",
  "to_translate": "fr,zu,hi"
}
```

4. What should I do if I get a 400 (Bad Request) error?

This error indicates an issue with your request, such as:

- Missing text, from_language, or to_translate parameters.
- Invalid language codes or request format.
- Ensure the request body is structured correctly and uses valid ISO 639-1 language codes.

Resources

- Azure Translator API Documentation
- Node.js Axios Guide