Spring 2023-ITIS-6177 System Integration

Azure Computer Vision API

By

Tejas Devendra Patil(801273490)

In this project we are designing a simple OCR Application using Computer Vision API.

OCR (Optical Character Recognition) is a process of converting images with printed or handwritten text into machine-encoded text. The OCR technology uses computer vision algorithms to analyze and recognize the text in an image. This documentation will explain the OCR code written in Node.js using the Express.js framework and Computer Vision API.

Prerequisites:

- Node.js installed on your computer
- Microsoft Azure account with access to the Computer Vision API
- I have deployed the code on Digital Ocean Server

Working of the Application:

1. **Upload image**:-
    - It is a Web-Application in which the user has to provide with the image **url.**
2. **Text Recognition**:-
    - The server analyzes the uploaded image using the Microsoft Cognitive Services Vision API for text recognition. The API returns a response object that contains the operation ID for the recognition process.
    - **Read API** is used of Computer Vision API, it performs the Read Operation. It is a POST call and provides with response header which has **Operation ID.**
    - **Get Read Result** operation of the Computer Vision API takes the Operation ID as parameter and retrieves the recognized text from the image.
3. **Error Handling**:-
    - The server handles errors that may occur during the text recognition process. The error message is displayed on the user's screen if an error occurs.
    - 400 : Bad or unrecognizable request JSON or binary file/Image format unsupported. Supported formats include JPEG, PNG, BMP, PDF and TIFF.
    - 415: Unsupported media type. 'Content-Type' does not match the content of the POST request.
    - 404: Operation ID is invalid or expired.

API Endpoints

Create a **.env** and add your Azure Subscription key there:

“API_KEY=<your-subscription-key>”

Testing the Application on Postman and UI:-

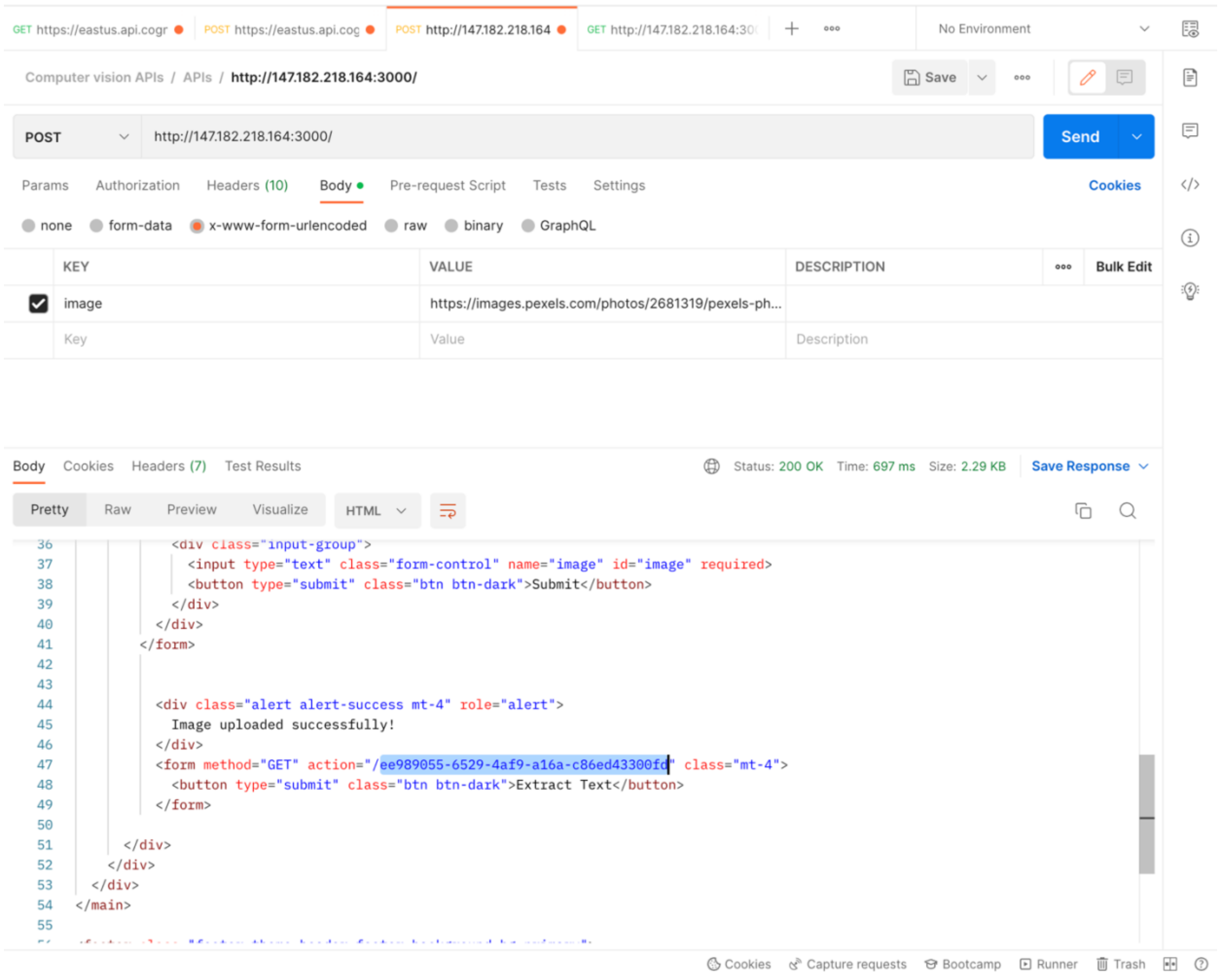
Post Method :- POST "/":

This endpoint handles the form submission from the index page. The image is sent as a request body parameter to this endpoint. The server analyzes the image and returns the operation ID for the text recognition process.

Request URL: ‘https://eastus.api.cognitive.microsoft.com/vision/v3.2/read/analyze’

API:- http://147.182.218.164:3000/

```
Headers:- {
        Ocp-Apim-Subscription-Key : “your-subscription-key”
        Content-Type: “Application/json”
        }
Body:- {
        “image”: https://images.pexels.com/photos/2681319/pexels-photo-2681319.jpeg?cs=srgb&dl=pexels-ivan-bertolazzi-2681319.jpg&fm=jpg
        }
```


GET https://eastus.api.cogr | POST https://eastus.api.cog | POST http://147.182.218.164 | GET http://147.182.218.164:30

No Environment

Computer vision APIs / APIs / **http://147.182.218.164:3000/**

POST http://147.182.218.164:3000/ Send

Params | Authorization | Headers (10) | **Body** | Pre-request Script | Tests | Settings | Cookies

none | form-data | [✓ x-www-form-urlencoded] | raw | binary | GraphQL

	KEY	VALUE	DESCRIPTION
✓	image	https://images.pexels.com/photos/2681319/pexels-ph...	
	Key	Value	Description

Body | Cookies | Headers (7) | Test Results

Status: 200 OK Time: 697 ms Size: 2.29 KB Save Response

Pretty | Raw | Preview | Visualize | HTML

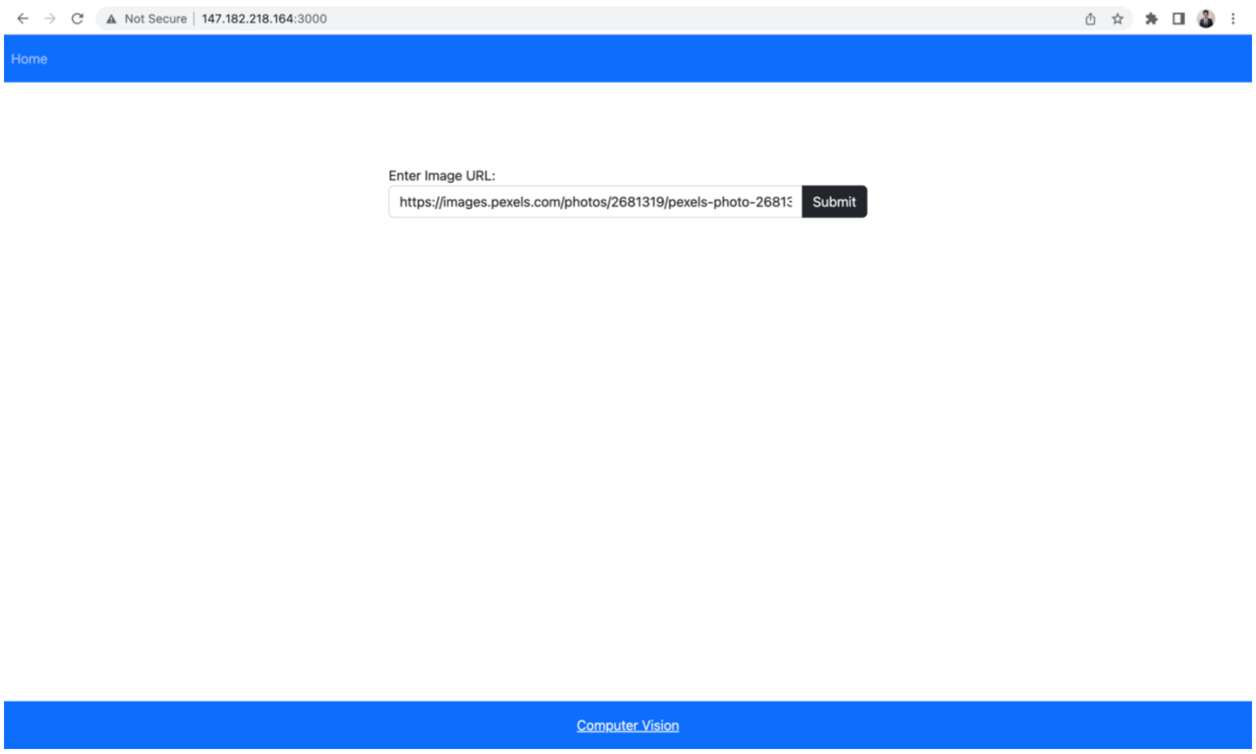
```
            <div class="input-group">
              <input type="text" class="form-control" name="image" id="image" required>
              <button type="submit" class="btn btn-dark">Submit</button>
            </div>
          </div>
        </form>


          <div class="alert alert-success mt-4" role="alert">
            Image uploaded successfully!
          </div>
          <form method="GET" action="/ee989055-6529-4af9-a16a-c86ed43300fd" class="mt-4">
            <button type="submit" class="btn btn-dark">Extract Text</button>
          </form>

        </div>
      </div>
    </div>
  </main>

```

Cookies | Capture requests | Bootcamp | Runner | Trash


Not Secure | 147.182.218.164:3000

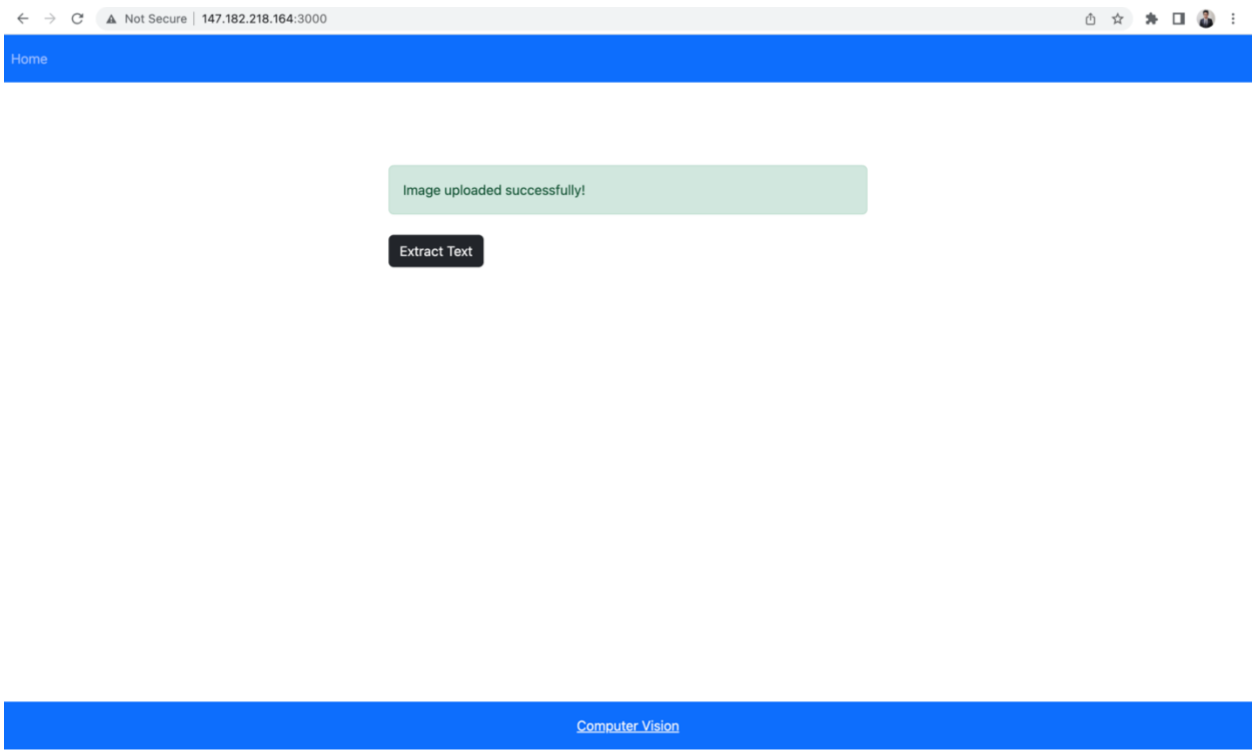
Home

Enter Image URL:

https://images.pexels.com/photos/2681319/pexels-photo-26813

Submit

Computer Vision


Not Secure | 147.182.218.164:3000

Home

Image uploaded successfully!

Extract Text

Computer Vision

GET Method:- GET "/:id":

This endpoint handles the polling for the text recognition process. The server sends requests to the Microsoft Cognitive Services Vision API with the operation ID until the process is complete. Once the process is complete, the recognized text is displayed on the user's screen.

Request URL: 'https://eastus.api.cognitive.microsoft.com/vision/v3.2/read/analyzeResults/{Operation ID}'

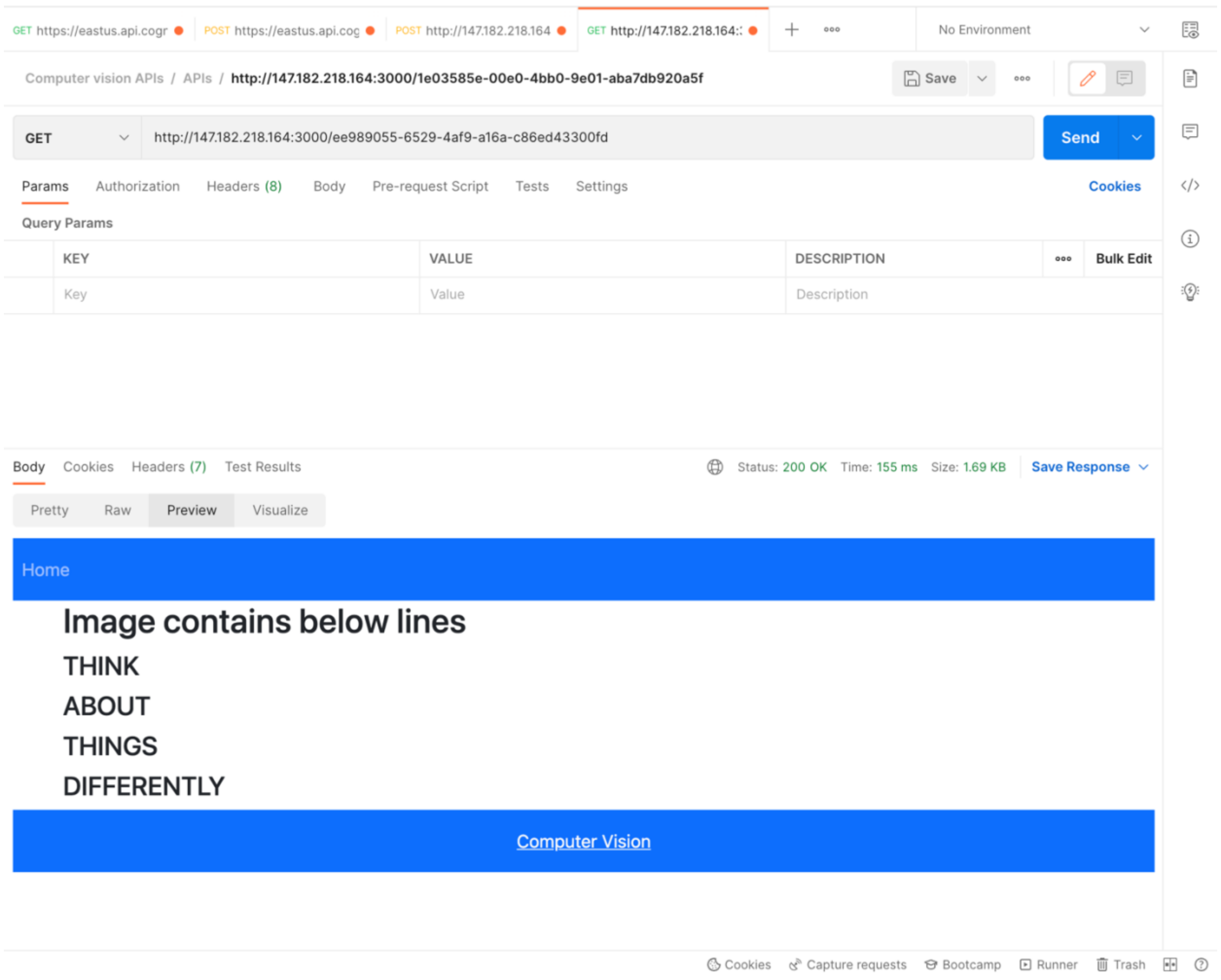
API:- http://147.182.218.164:3000/ee989055-6529-4af9-a16a-c86ed43300fd

Headers:- {

Ocp-Apim-Subscription-Key : "your-subscription-key"

Content-Type: "Application/json"

}



```
GET http://147.182.218.164:3000/ee989055-6529-4af9-a16a-c86ed43300fd
Status: 200 OK   Time: 155 ms   Size: 1.69 KB
```

Home

Image contains below lines

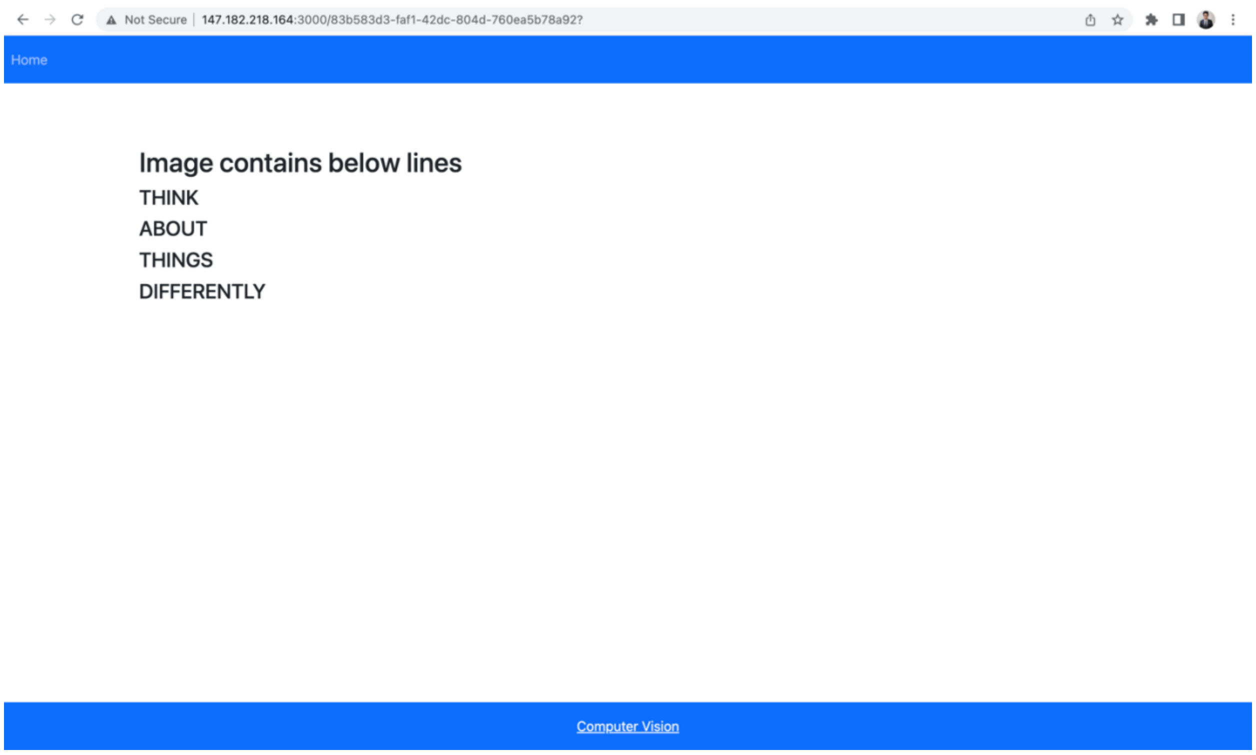
THINK

ABOUT

THINGS

DIFFERENTLY

Computer Vision


Not Secure | 147.182.218.164:3000/83b583d3-faf1-42dc-804d-760ea5b78a92?

Home

Image contains below lines

THINK

ABOUT

THINGS

DIFFERENTLY

Computer Vision

Future Scopes:-

1) UI can be optimized.
2) Other Computer Vision Endpoints can also be leveraged and create a bigger application.
3) For example, Detect Objects and Describe Objects endpoints can be used to determine if the image contains mature/graphic content.