Exploring Azure AI:

A Tour of Azure AI Services for Modern Applications

by Marian Veteanu

Azure AI services

Azure OpenAI

Azure AI Search

Content Safety

Speech

Document Intelligence

Vision

Custom Vision

Face

Translator

Language

Video Indexer

Immersive Reader

Azure OpenAI

The Azure OpenAI Service is a platform that provides access to advanced generative AI models developed by OpenAI, such as GPT-4, Codex, and DALL-E, directly through the Azure ecosystem.

Key Models

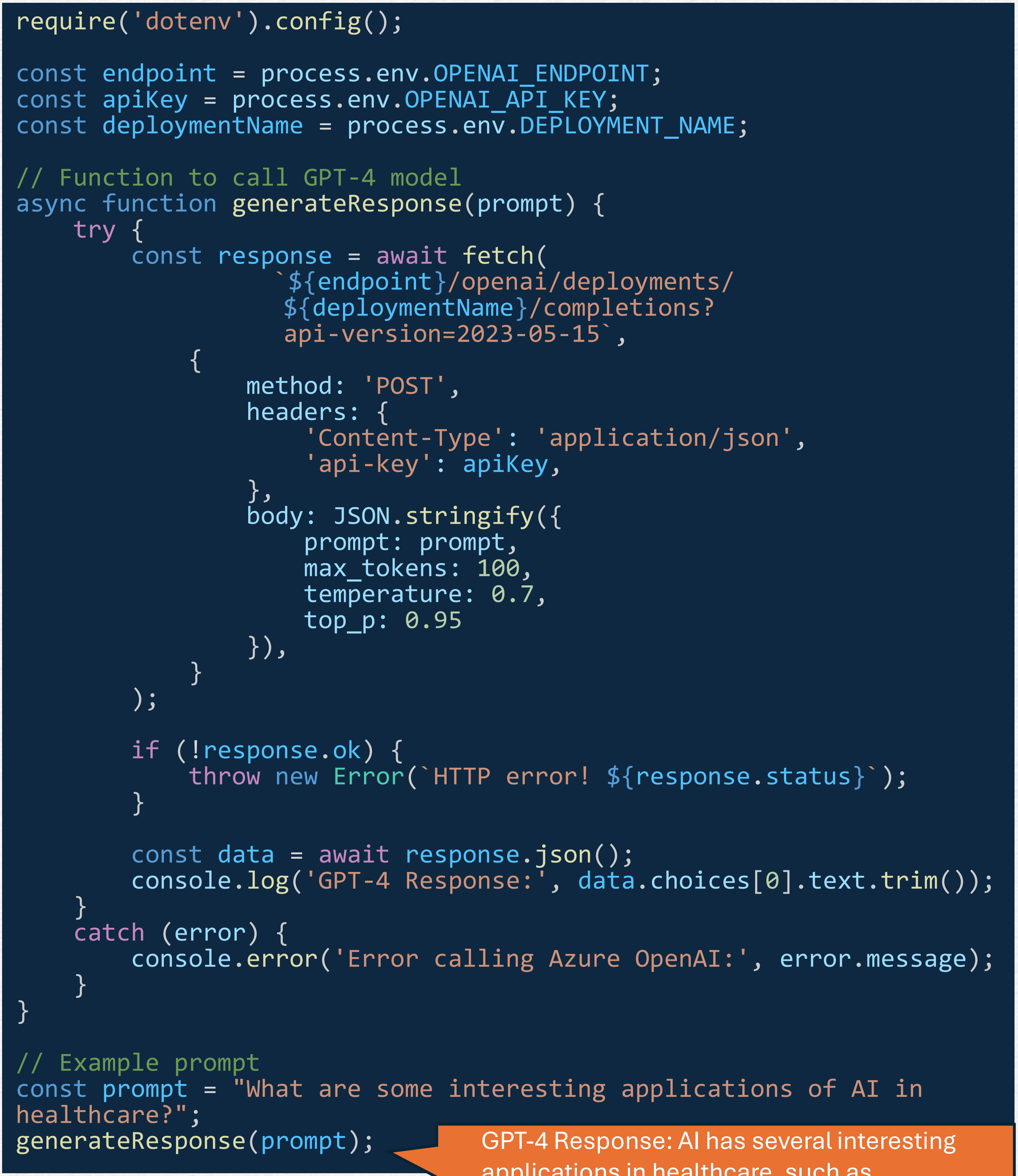
- **GPT-4 and GPT-3:** These language models generate human-like text, complete sentences, answer questions, summarize content, and provide conversational AI capabilities.

- **Codex:** Codex is designed to understand and generate code, making it a valuable tool for developers and software automation.

- **DALL-E:** DALL-E is a model that generates images based on text prompts, useful for creative and design-related tasks.

- **Whisper (Speech Recognition):** Whisper provides robust speech-to-text capabilities, enabling transcription and real-time voice applications.

Example using Azure OpenAI GPT-4



```
require('dotenv').config();

const endpoint = process.env.OPENAI_ENDPOINT;
const apiKey = process.env.OPENAI_API_KEY;
const deploymentName = process.env.DEPLOYMENT_NAME;

// Function to call GPT-4 model
async function generateResponse(prompt) {
    try {
        const response = await fetch(
                  `${endpoint}/openai/deployments/
                   ${deploymentName}/completions?
                   api-version=2023-05-15`,
            {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                    'api-key': apiKey,
                },
                body: JSON.stringify({
                    prompt: prompt,
                    max_tokens: 100,
                    temperature: 0.7,
                    top_p: 0.95
                }),
            }
        );

        if (!response.ok) {
            throw new Error(`HTTP error! ${response.status}`);
        }

        const data = await response.json();
        console.log('GPT-4 Response:', data.choices[0].text.trim());
    }
    catch (error) {
        console.error('Error calling Azure OpenAI:', error.message);
    }
}

// Example prompt
const prompt = "What are some interesting applications of AI in
healthcare?";
generateResponse(prompt);
```

GPT-4 Response: AI has several interesting applications in healthcare, such as predictive analytics, personalized medicine, and medical image analysis...

Example using Azure OpenAI Whisper

```
dotenv.config();

const endpoint = process.env.OPENAI_ENDPOINT;
const apiKey = process.env.OPENAI_API_KEY;
const deploymentName = process.env.DEPLOYMENT_NAME;

// Function to call Whisper for transcription
async function transcribeAudio(filePath) {
    try {
        // Read the audio file into a buffer
        const audioBuffer = fs.readFileSync(filePath);

        const response = await fetch(
            `${endpoint}/openai/deployments/
           ${deploymentName}/transcriptions?api-version=2023-05-15`,
            {
                method: 'POST',
                headers: {
                    'Content-Type': 'audio/wav',
                    'api-key': apiKey,
                    'model': 'whisper-1'
                },
                body: audioBuffer
            }
        );

        if (!response.ok)
            throw new Error(`HTTP error! Status:
${response.status}`);

        const data = await response.json();
        console.log('Transcription:', data.text);
    } catch (error) {
        console.error('Error transcribing audio:', error.message);
    }
}

// Example usage
const audioFilePath = 'path/to/your/audio-file.wav';
transcribeAudio(audioFilePath);
```

You should see the transcribed text printed to the console, like:

```
Transcription: "This is a sample audio transcription text."
```

Azure AI Search

Azure AI Search is a powerful, AI-enhanced search service from Microsoft Azure that allows developers to integrate search functionality into web and mobile applications.

Azure AI Search indexes and searches through data sources connected to it, allowing it to search various types of content:

- **Structured and Unstructured Data in Databases.** It can index structured data from databases like Azure SQL Database, Cosmos DB, or other SQL and NoSQL sources.
- **Documents and Files.** Azure AI Search can index content from document storage like Azure Blob Storage, including files such as PDFs, Word documents, Excel spreadsheets, images, and text files.
- **Text from Images and Scanned Documents.** Through cognitive skills, Azure AI Search can apply Optical Character Recognition (OCR) to extract text from images and scanned documents, making the text searchable.
- **Metadata from Audio and Video Files.** Azure AI Search can use Azure Video Indexer to extract and index metadata, like keywords, speakers, and key phrases, from audio and video content.
- **Web Content and JSON Data.** Azure AI Search can index JSON files and structured web content to enable searching through structured, semi-structured, or hierarchical data.
- **Application-Specific Data Models.** Azure AI Search works well with custom data models, where data is organized to fit the needs of specific applications or industries.

Example Basic Full-Text Search in PDFs

Perform a Basic Full-Text Search in PDFs stored in Azure Blob Storage using Azure AI Search.

1. Store PDFs in Azure Blob Storage
Upload your PDF files to a container in Azure Blob Storage.

2. Create an Azure AI Search Resource
Create an Azure AI Search resource if you don't already have one. Get the endpoint URL and API key for your search service.

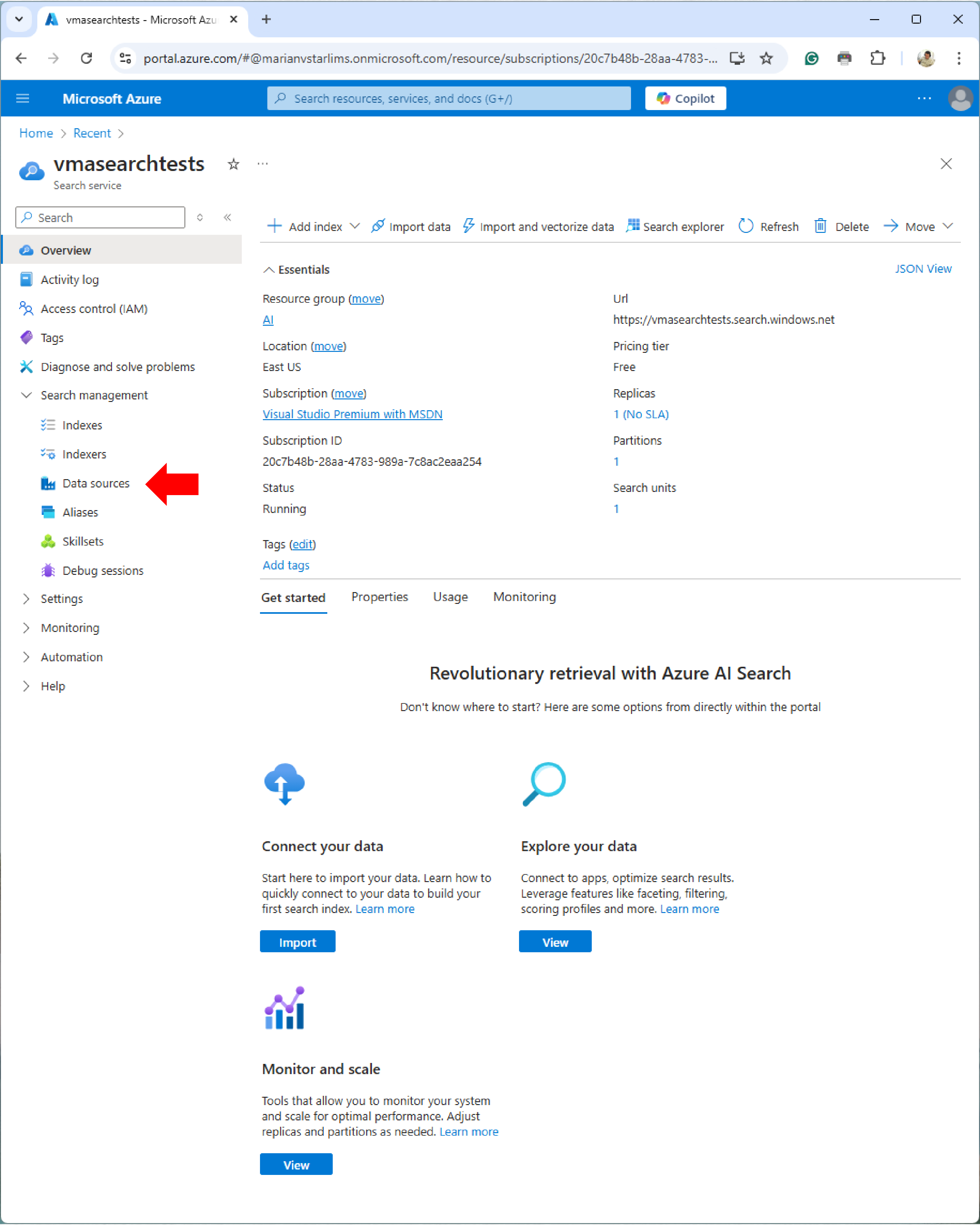
3. Create a Data Source in Azure AI Search
Go to your Azure AI Search resource in the Azure portal. Create a new data source with the following configuration:
Name: Give your data source a name (e.g., pdf-datasource).
Type: Choose Azure Blob Storage.
Connection String: Use your blob storage connection string.
Container: Enter the name of the blob container where your PDFs are stored.
Parsing Mode: Set to delimited if your PDFs contain structured data, or leave as default.

4. Create a Skillset (Optional)
This is optional if you only need basic text extraction, as Azure AI Search includes a built-in OCR capability.

5. Create an Index
In Azure AI Search, create a new index that defines the fields you want to make searchable. Define fields like content, metadata_title, and metadata_author to store the extracted content and metadata from PDFs.

6. Create an Indexer
Create an indexer to pull data from your blob storage and populate your search index:
Name: Give your indexer a name (e.g., pdf-indexer).
Data source: Select the data source you created.
Index: Select the index you created.
Schedule: Configure the indexer to run on a schedule or on-demand.

7. Perform a Basic Full-Text Search (see next page)



vmasearchtests - Microsoft Azu ×

portal.azure.com/#@marianvstarlims.onmicrosoft.com/resource/subscriptions/20c7b48b-28aa-4783-...

Microsoft Azure

Search resources, services, and docs (G+/)

Copilot

Home > Recent >

vmasearchtests ☆ ···

Search service

Search

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Search management
  - Indexes
  - Indexers
  - Data sources
  - Aliases
  - Skillsets
  - Debug sessions
- Settings
- Monitoring
- Automation
- Help

+ Add index | Import data | Import and vectorize data | Search explorer | Refresh | Delete | Move

^ Essentials

JSON View

Resource group (move) AI	Url https://vmasearchtests.search.windows.net
Location (move) East US	Pricing tier Free
Subscription (move) Visual Studio Premium with MSDN	Replicas 1 (No SLA)
Subscription ID 20c7b48b-28aa-4783-989a-7c8ac2eaa254	Partitions 1
Status Running	Search units 1

Tags (edit)

Add tags

Get started | Properties | Usage | Monitoring

Revolutionary retrieval with Azure AI Search

Don't know where to start? Here are some options from directly within the portal

Connect your data

Start here to import your data. Learn how to quickly connect to your data to build your first search index. Learn more

Import

Explore your data

Connect to apps, optimize search results. Leverage features like faceting, filtering, scoring profiles and more. Learn more

View

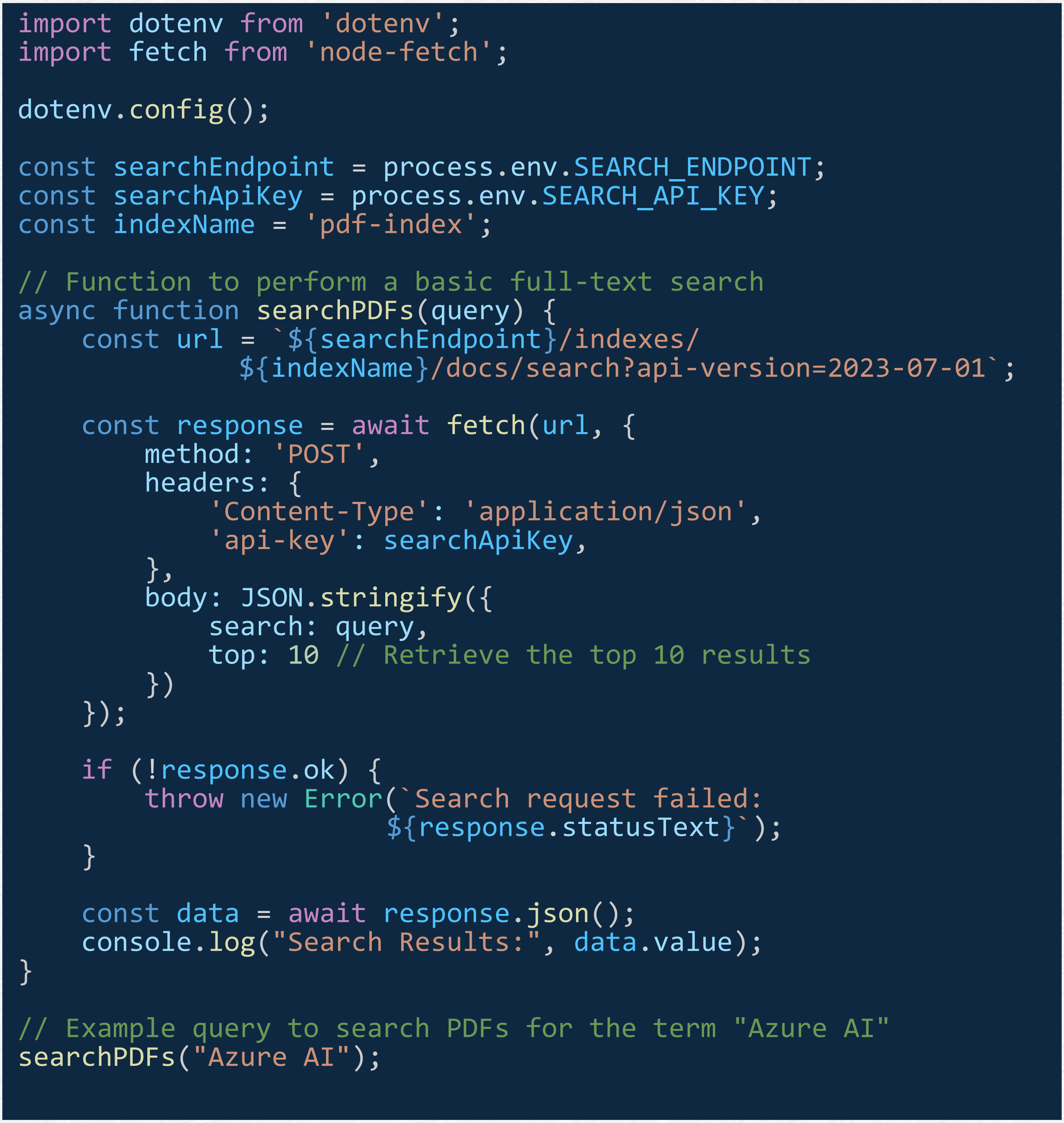
Monitor and scale

Tools that allow you to monitor your system and scale for optimal performance. Adjust replicas and partitions as needed. Learn more

View

Example Basic Full-Text Search in PDFs – part 2

Once indexing is complete, you can perform a full-text search on your PDFs. Here's an example for querying the indexed PDFs:



```
import dotenv from 'dotenv';
import fetch from 'node-fetch';

dotenv.config();

const searchEndpoint = process.env.SEARCH_ENDPOINT;
const searchApiKey = process.env.SEARCH_API_KEY;
const indexName = 'pdf-index';

// Function to perform a basic full-text search
async function searchPDFs(query) {
    const url = `${searchEndpoint}/indexes/
              ${indexName}/docs/search?api-version=2023-07-01`;

    const response = await fetch(url, {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            'api-key': searchApiKey,
        },
        body: JSON.stringify({
            search: query,
            top: 10 // Retrieve the top 10 results
        })
    });

    if (!response.ok) {
        throw new Error(`Search request failed:
                       ${response.statusText}`);
    }

    const data = await response.json();
    console.log("Search Results:", data.value);
}

// Example query to search PDFs for the term "Azure AI"
searchPDFs("Azure AI");
```

Example: Searching documents in Azure Cosmos DB

Step 1: Store Documents in Azure Cosmos DB. Example:

```
{
  "id": "1",
  "title": "Introduction to Cloud Computing",
  "content": "Cloud computing is a way to store and access
data over the internet...",
  "category": "Technology"
}
```

Step 2: Create an Azure AI Search Resource

Step 3: Go to Data Sources and create a new data source:

Name: Give your data source a name (e.g., cosmosdb-datasource).
Type: Select Azure Cosmos DB.
Connection String: Provide the Cosmos DB connection string.
Container: Specify the Cosmos DB container that holds your documents.
Query: Optionally, provide a query to filter documents if needed.

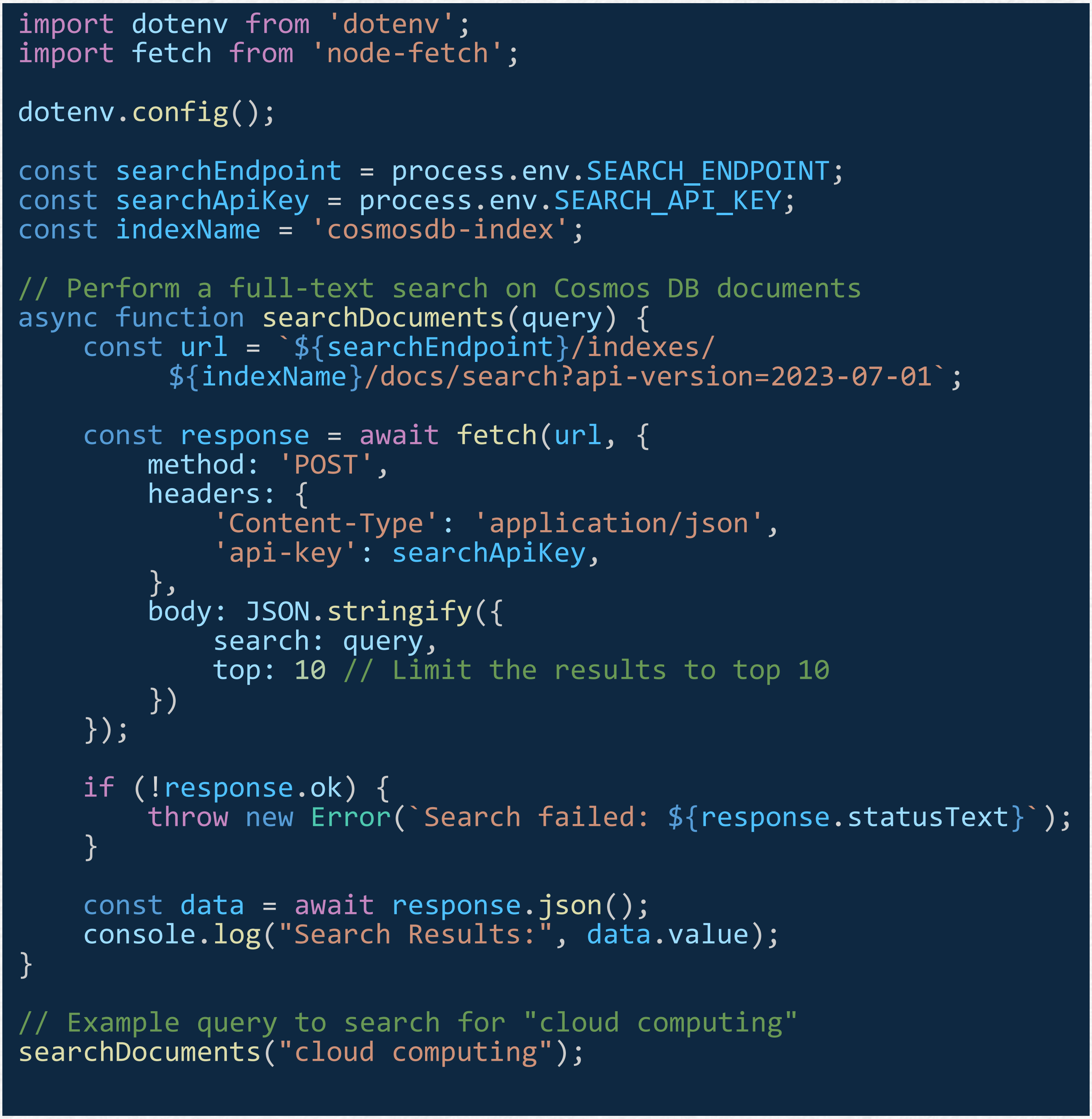
Step 4: Go to Indexes and create a new index:

id: Set as key.
title: Set as searchable.
content: Set as searchable.
category: Set as filterable.

Step 5: Create an Indexer

Step 6: Query the Indexed Documents

Example: Searching documents in Azure Cosmos DB – part 2



```
import dotenv from 'dotenv';
import fetch from 'node-fetch';

dotenv.config();

const searchEndpoint = process.env.SEARCH_ENDPOINT;
const searchApiKey = process.env.SEARCH_API_KEY;
const indexName = 'cosmosdb-index';

// Perform a full-text search on Cosmos DB documents
async function searchDocuments(query) {
    const url = `${searchEndpoint}/indexes/
         ${indexName}/docs/search?api-version=2023-07-01`;

    const response = await fetch(url, {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            'api-key': searchApiKey,
        },
        body: JSON.stringify({
            search: query,
            top: 10 // Limit the results to top 10
        })
    });

    if (!response.ok) {
        throw new Error(`Search failed: ${response.statusText}`);
    }

    const data = await response.json();
    console.log("Search Results:", data.value);
}

// Example query to search for "cloud computing"
searchDocuments("cloud computing");
```

The console output should show the top 10 documents containing the term "cloud computing," including their titles, content, and other fields defined in your index.

Content Safety

Azure AI Content Safety is a service within Microsoft's Azure ecosystem designed to help organizations detect and filter harmful, offensive, or inappropriate content. This service enables developers to protect users from exposure to unsafe content by flagging or blocking it, which is critical for applications in social media, customer support, gaming, and educational platforms.

Key Features of Azure AI Content Safety

Text Moderation. Detects inappropriate or harmful text content, including profanity, hate speech, bullying, and sexually explicit content.

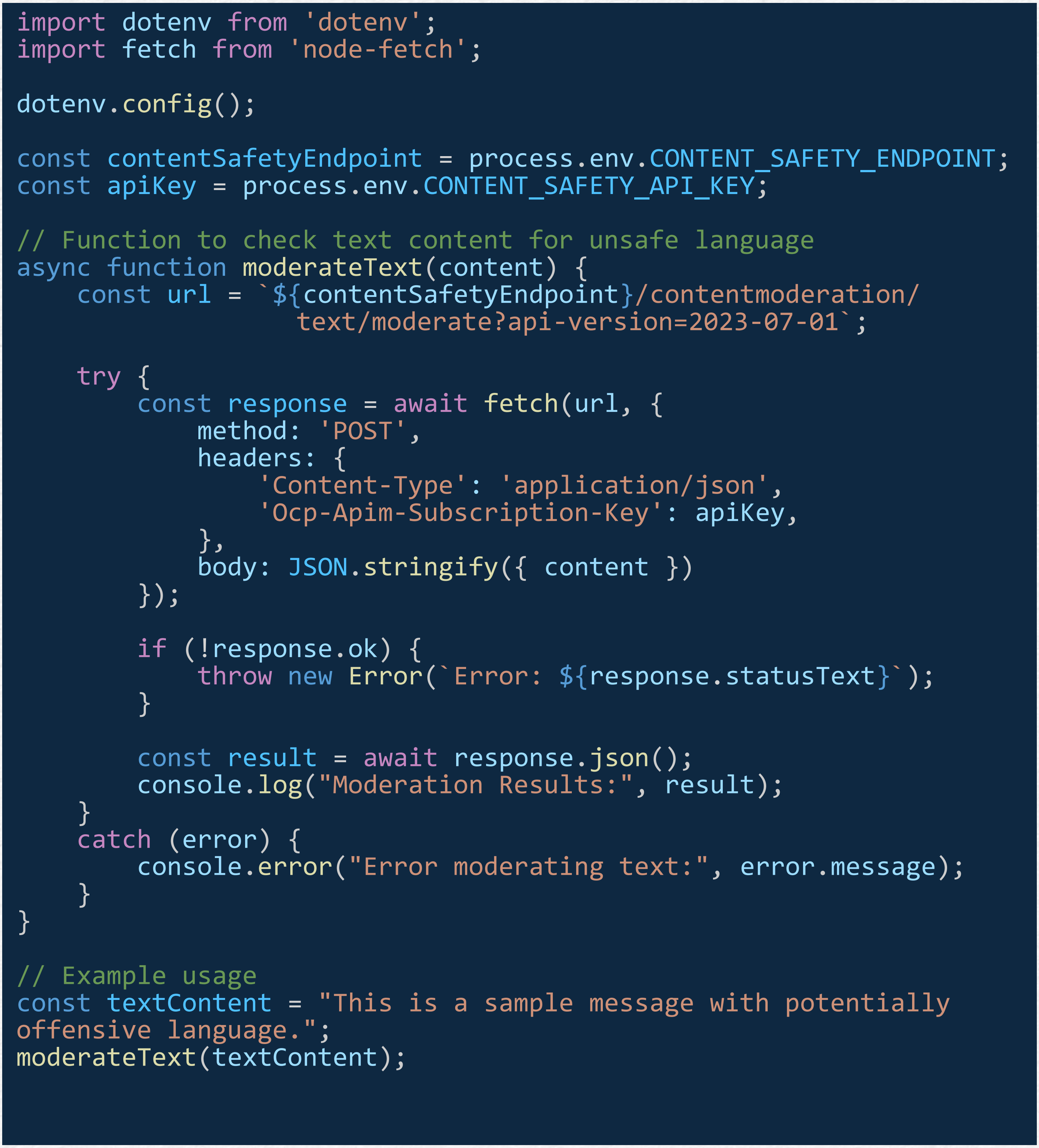
Image Moderation. Scans images to detect inappropriate visual content, such as nudity, violent imagery, or other unsafe elements.

Customizable Moderation Policies. Allows for custom rules based on the organization’s needs, supporting unique definitions of "unsafe" content based on industry or community standards.

Real-Time and Batch Processing. Supports both real-time moderation for quick responses and batch processing for large volumes of content.

Integration with Other Azure Services. Can be integrated with other Azure services to create automated moderation workflows.

Example of Text Moderation



```
import dotenv from 'dotenv';
import fetch from 'node-fetch';

dotenv.config();

const contentSafetyEndpoint = process.env.CONTENT_SAFETY_ENDPOINT;
const apiKey = process.env.CONTENT_SAFETY_API_KEY;

// Function to check text content for unsafe language
async function moderateText(content) {
    const url = `${contentSafetyEndpoint}/contentmoderation/
                  text/moderate?api-version=2023-07-01`;

    try {
        const response = await fetch(url, {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
                'Ocp-Apim-Subscription-Key': apiKey,
            },
            body: JSON.stringify({ content })
        });

        if (!response.ok) {
            throw new Error(`Error: ${response.statusText}`);
        }

        const result = await response.json();
        console.log("Moderation Results:", result);
    }
    catch (error) {
        console.error("Error moderating text:", error.message);
    }
}

// Example usage
const textContent = "This is a sample message with potentially
offensive language.";
moderateText(textContent);
```

Speech

The **Azure AI Speech Service** is a cloud-based service that provides powerful speech capabilities, including speech-to-text (STT), text-to-speech (TTS), speech translation, and speaker recognition. These capabilities are great for applications in customer support, accessibility, virtual assistants, and more.

Key Capabilities of Azure Speech Service

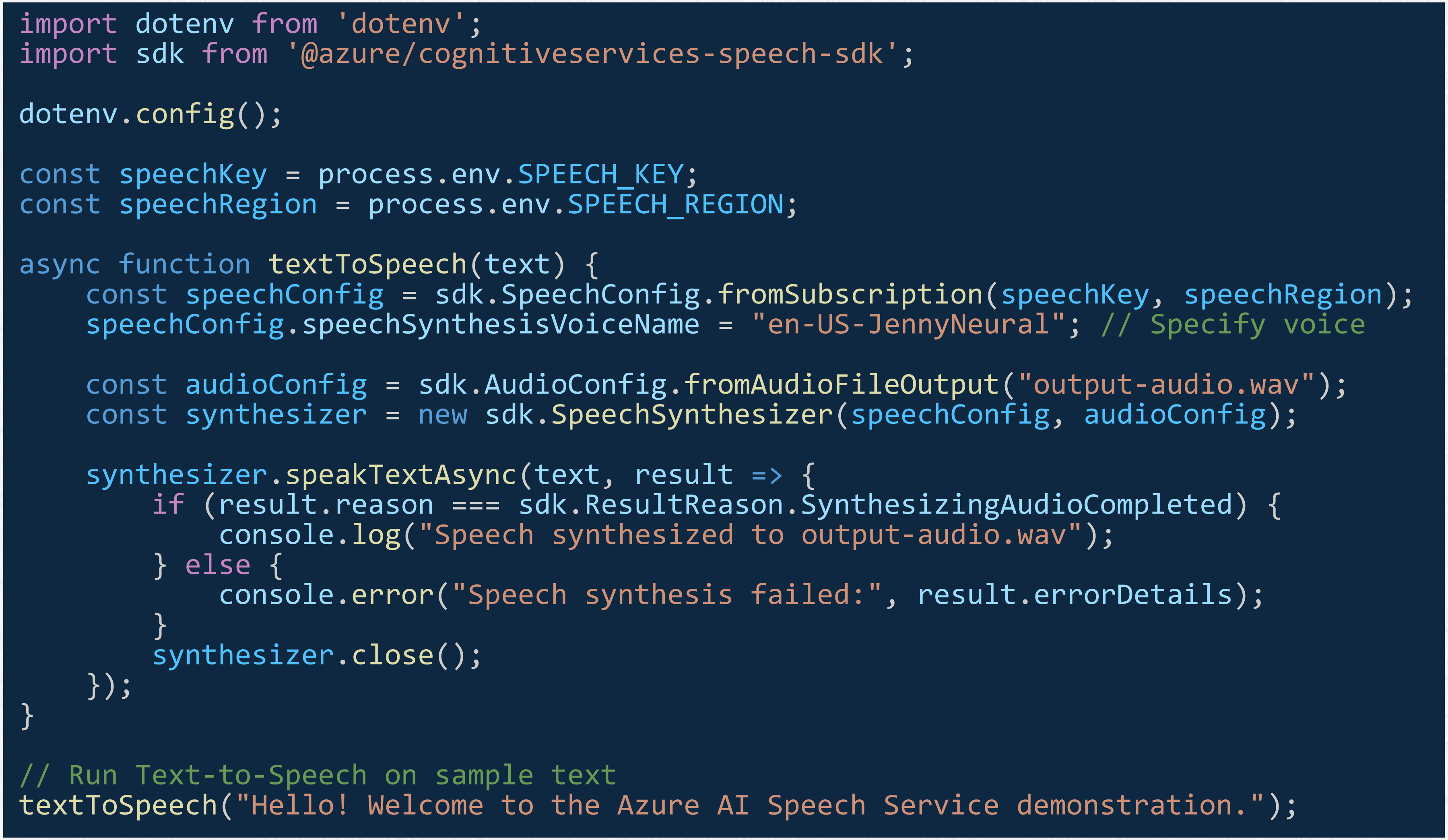
Speech-to-Text (STT). Converts spoken language into written text in real-time or from pre-recorded audio files.

Text-to-Speech (TTS). Converts written text into natural-sounding audio. Azure offers neural voices, which provide highly realistic speech synthesis.

Speech Translation. Provides real-time translation of spoken language, enabling multi-language conversations.

Speaker Recognition. Identifies and verifies speakers based on their unique voice characteristics.

Example Text to Speech



```javascript
import dotenv from 'dotenv';
import sdk from '@azure/cognitiveservices-speech-sdk';

dotenv.config();

const speechKey = process.env.SPEECH_KEY;
const speechRegion = process.env.SPEECH_REGION;

async function textToSpeech(text) {
    const speechConfig = sdk.SpeechConfig.fromSubscription(speechKey, speechRegion);
    speechConfig.speechSynthesisVoiceName = "en-US-JennyNeural"; // Specify voice

    const audioConfig = sdk.AudioConfig.fromAudioFileOutput("output-audio.wav");
    const synthesizer = new sdk.SpeechSynthesizer(speechConfig, audioConfig);

    synthesizer.speakTextAsync(text, result => {
        if (result.reason === sdk.ResultReason.SynthesizingAudioCompleted) {
            console.log("Speech synthesized to output-audio.wav");
        } else {
            console.error("Speech synthesis failed:", result.errorDetails);
        }
        synthesizer.close();
    });
}

// Run Text-to-Speech on sample text
textToSpeech("Hello! Welcome to the Azure AI Speech Service demonstration.");
```

Document Intelligence

Azure AI Document Intelligence (formerly known as Azure Form Recognizer) is a service in Azure that enables applications to extract structured information from various types of documents such as forms, receipts, invoices, and other documents.

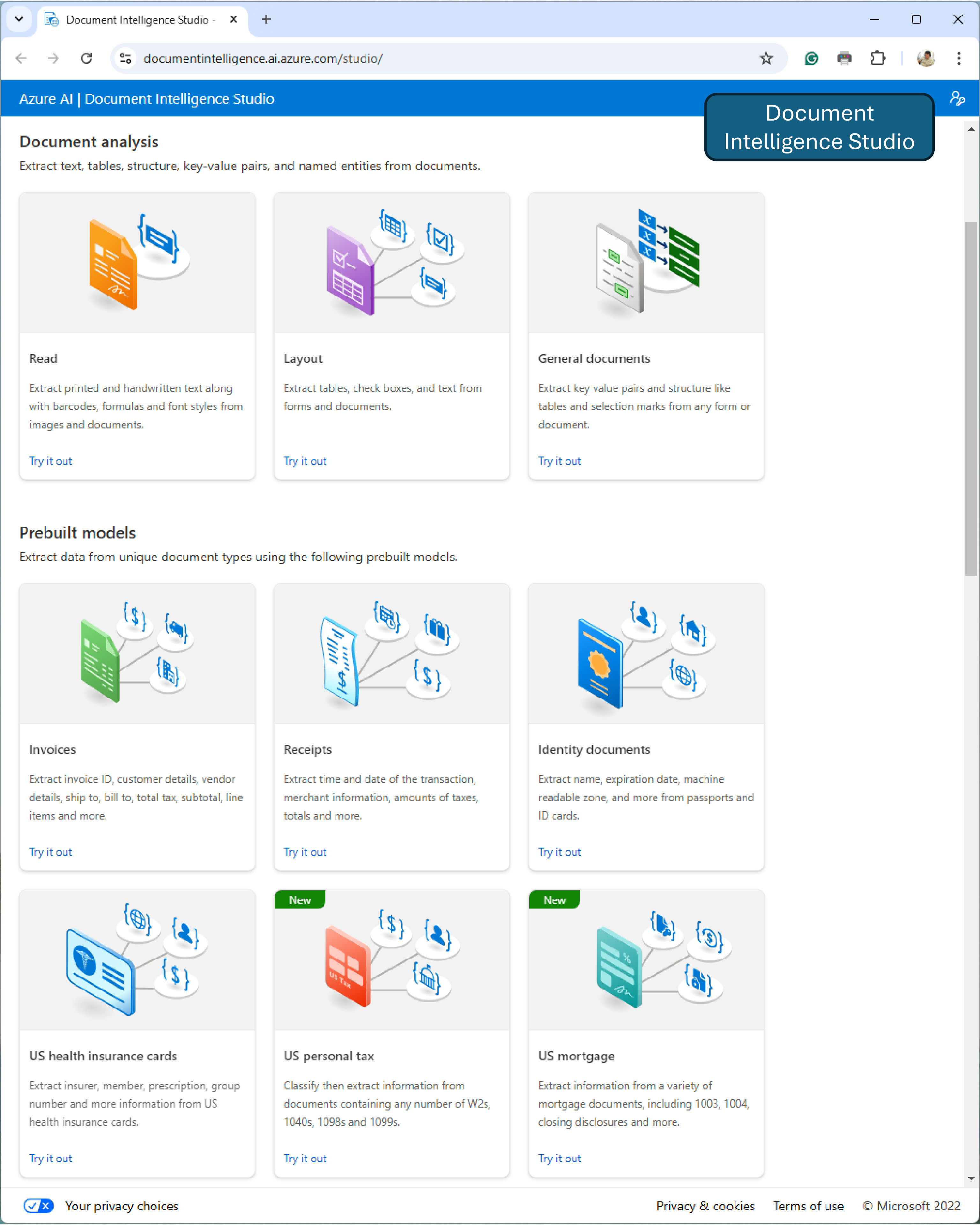
Key Capabilities of Azure AI Document Intelligence

Pre-built Models. Provides ready-to-use models for common document types: Invoices, Receipts, IDs, US Health Insurance Cards, US Personal Tax, US mortgage, US pay stubs, US bank statements, US checks, Credit cards, US marriage certificates, Contracts, Business cards

Custom Document Models. Allows you to train custom models for specific document types that don’t fit the pre-built templates.

Layout Extraction. Extracts text, tables, and other layout information from documents, making it easy to parse structured and semi-structured documents.

Key-Value Pair Extraction. Detects and extracts key-value pairs from documents, which is useful for forms and structured documents.



Document Intelligence Studio

Document Intelligence Studio -
documentintelligence.ai.azure.com/studio/
Azure AI | Document Intelligence Studio

Document analysis

Extract text, tables, structure, key-value pairs, and named entities from documents.

Read

Extract printed and handwritten text along with barcodes, formulas and font styles from images and documents.

Try it out

Layout

Extract tables, check boxes, and text from forms and documents.

Try it out

General documents

Extract key value pairs and structure like tables and selection marks from any form or document.

Try it out

Prebuilt models

Extract data from unique document types using the following prebuilt models.

Invoices

Extract invoice ID, customer details, vendor details, ship to, bill to, total tax, subtotal, line items and more.

Try it out

Receipts

Extract time and date of the transaction, merchant information, amounts of taxes, totals and more.

Try it out

Identity documents

Extract name, expiration date, machine readable zone, and more from passports and ID cards.

Try it out

US health insurance cards

Extract insurer, member, prescription, group number and more information from US health insurance cards.

Try it out

New

US personal tax

Classify then extract information from documents containing any number of W2s, 1040s, 1098s and 1099s.

Try it out

New

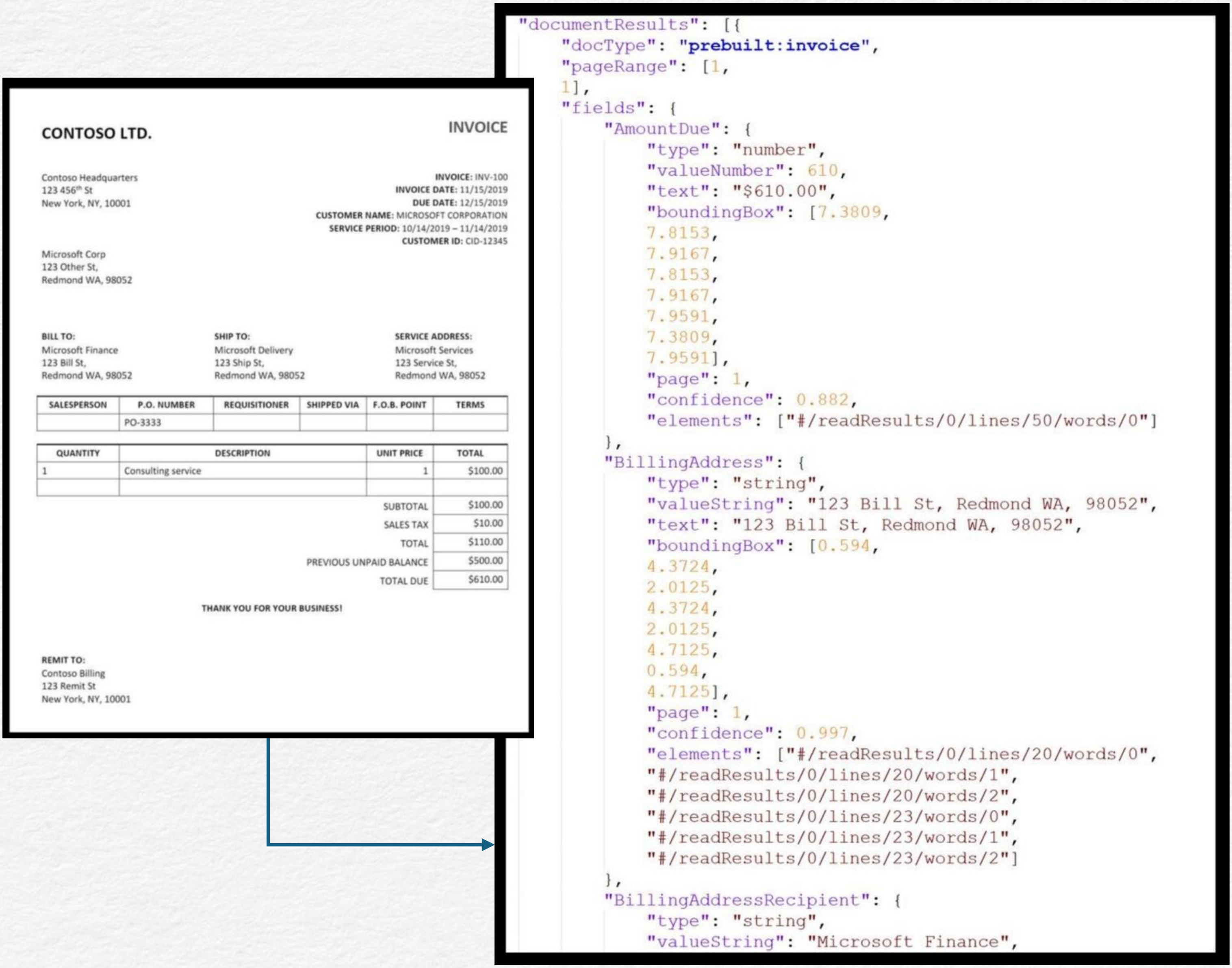
US mortgage

Extract information from a variety of mortgage documents, including 1003, 1004, closing disclosures and more.

Try it out

Document to JSON!

Extract text, key-value pairs, and tables accurately from diverse documents, forms, receipts, invoices, and cards—no need for manual labeling, extensive coding, or ongoing maintenance by document type.



CONTOSO LTD.

INVOICE

Contoso Headquarters
123 456th St
New York, NY, 10001

INVOICE: INV-100
INVOICE DATE: 11/15/2019
DUE DATE: 12/15/2019
CUSTOMER NAME: MICROSOFT CORPORATION
SERVICE PERIOD: 10/14/2019 – 11/14/2019
CUSTOMER ID: CID-12345

Microsoft Corp
123 Other St,
Redmond WA, 98052

BILL TO:
Microsoft Finance
123 Bill St,
Redmond WA, 98052

SHIP TO:
Microsoft Delivery
123 Ship St,
Redmond WA, 98052

SERVICE ADDRESS:
Microsoft Services
123 Service St,
Redmond WA, 98052

SALESPERSON	P.O. NUMBER	REQUISITIONER	SHIPPED VIA	F.O.B. POINT	TERMS
	PO-3333				

QUANTITY	DESCRIPTION	UNIT PRICE	TOTAL
1	Consulting service	1	$100.00
		SUBTOTAL	$100.00
		SALES TAX	$10.00
		TOTAL	$110.00
		PREVIOUS UNPAID BALANCE	$500.00
		TOTAL DUE	$610.00

THANK YOU FOR YOUR BUSINESS!

REMIT TO:
Contoso Billing
123 Remit St
New York, NY, 10001

```
"documentResults": [{
    "docType": "prebuilt:invoice",
    "pageRange": [1,
    1],
    "fields": {
        "AmountDue": {
            "type": "number",
            "valueNumber": 610,
            "text": "$610.00",
            "boundingBox": [7.3809,
            7.8153,
            7.9167,
            7.8153,
            7.9167,
            7.9591,
            7.3809,
            7.9591],
            "page": 1,
            "confidence": 0.882,
            "elements": ["#/readResults/0/lines/50/words/0"]
        },
        "BillingAddress": {
            "type": "string",
            "valueString": "123 Bill St, Redmond WA, 98052",
            "text": "123 Bill St, Redmond WA, 98052",
            "boundingBox": [0.594,
            4.3724,
            2.0125,
            4.3724,
            2.0125,
            4.7125,
            0.594,
            4.7125],
            "page": 1,
            "confidence": 0.997,
            "elements": ["#/readResults/0/lines/20/words/0",
            "#/readResults/0/lines/20/words/1",
            "#/readResults/0/lines/20/words/2",
            "#/readResults/0/lines/23/words/0",
            "#/readResults/0/lines/23/words/1",
            "#/readResults/0/lines/23/words/2"]
        },
        "BillingAddressRecipient": {
            "type": "string",
            "valueString": "Microsoft Finance",
```

Vision

The cloud-based Azure AI Vision service offers developers powerful algorithms for processing images and extracting valuable information.

By uploading an image or providing an image URL, Azure AI Vision can analyze visual content in multiple ways, tailored to your inputs and preferences.

Key services:

OCR

- Extract text from images

Spatial analysis

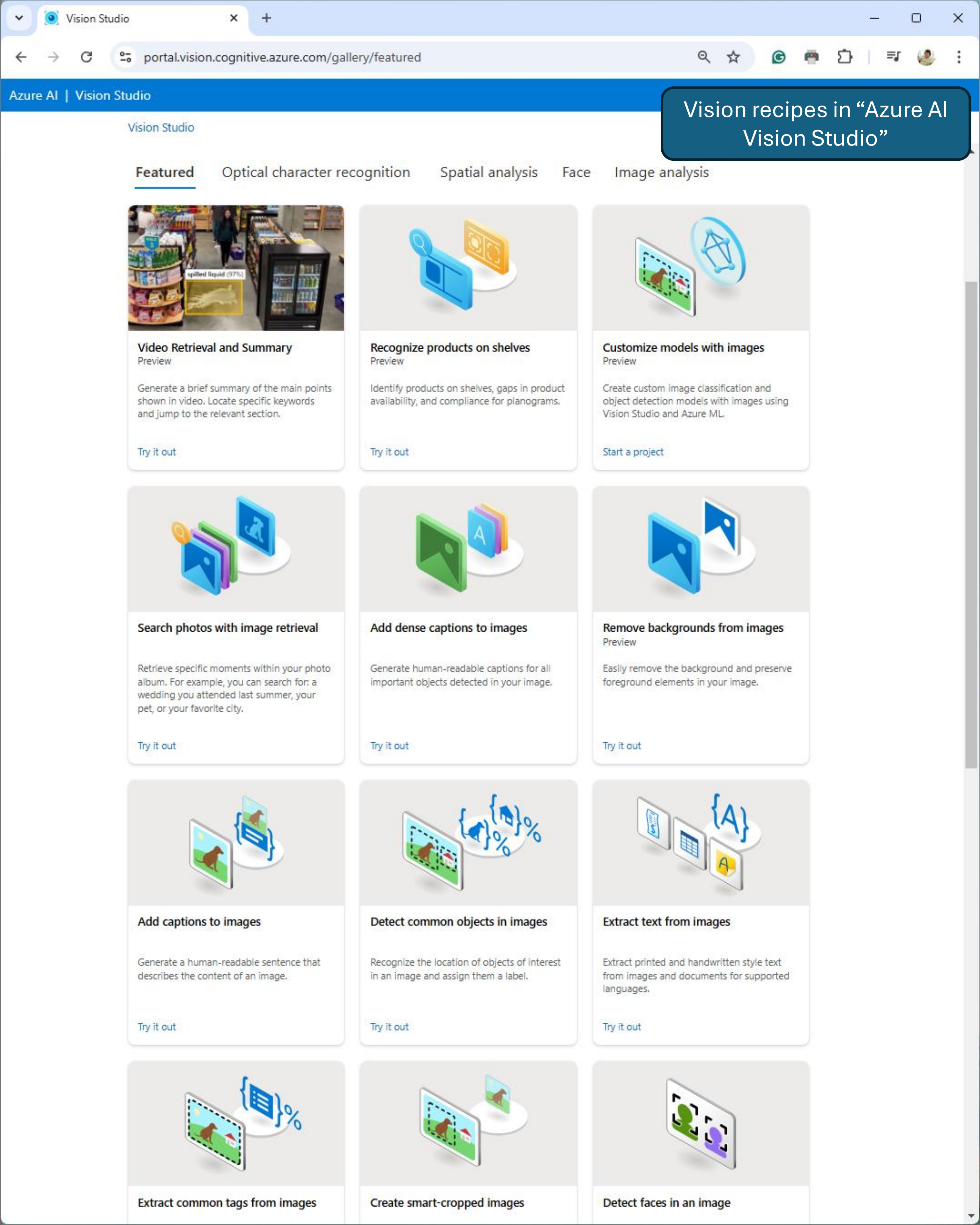
- Video Retrieval and Summary
- Count people in an area
- Detect when people cross a line
- Detect when people enter/exit a zone
- Monitor social distance

Face

- Detect faces in an image
- Liveness detection
- Portrait processing
- Photo ID matching

Image analysis

- Recognize products on shelves
- Search photos with image retrieval
- Remove background from images
- Add captions to images
- Detect common objects in images
- Extract common tags from images
- Detect sensitive content in images
- Create smart-cropped images



Vision recipes in “Azure AI Vision Studio”

Video Retrieval and Summary
Preview

Generate a brief summary of the main points shown in video. Locate specific keywords and jump to the relevant section.

Try it out

Recognize products on shelves
Preview

Identify products on shelves, gaps in product availability, and compliance for planograms.

Try it out

Customize models with images
Preview

Create custom image classification and object detection models with images using Vision Studio and Azure ML.

Start a project

Search photos with image retrieval

Retrieve specific moments within your photo album. For example, you can search for: a wedding you attended last summer, your pet, or your favorite city.

Try it out

Add dense captions to images

Generate human-readable captions for all important objects detected in your image.

Try it out

Remove backgrounds from images
Preview

Easily remove the background and preserve foreground elements in your image.

Try it out

Add captions to images

Generate a human-readable sentence that describes the content of an image.

Try it out

Detect common objects in images

Recognize the location of objects of interest in an image and assign them a label.

Try it out

Extract text from images

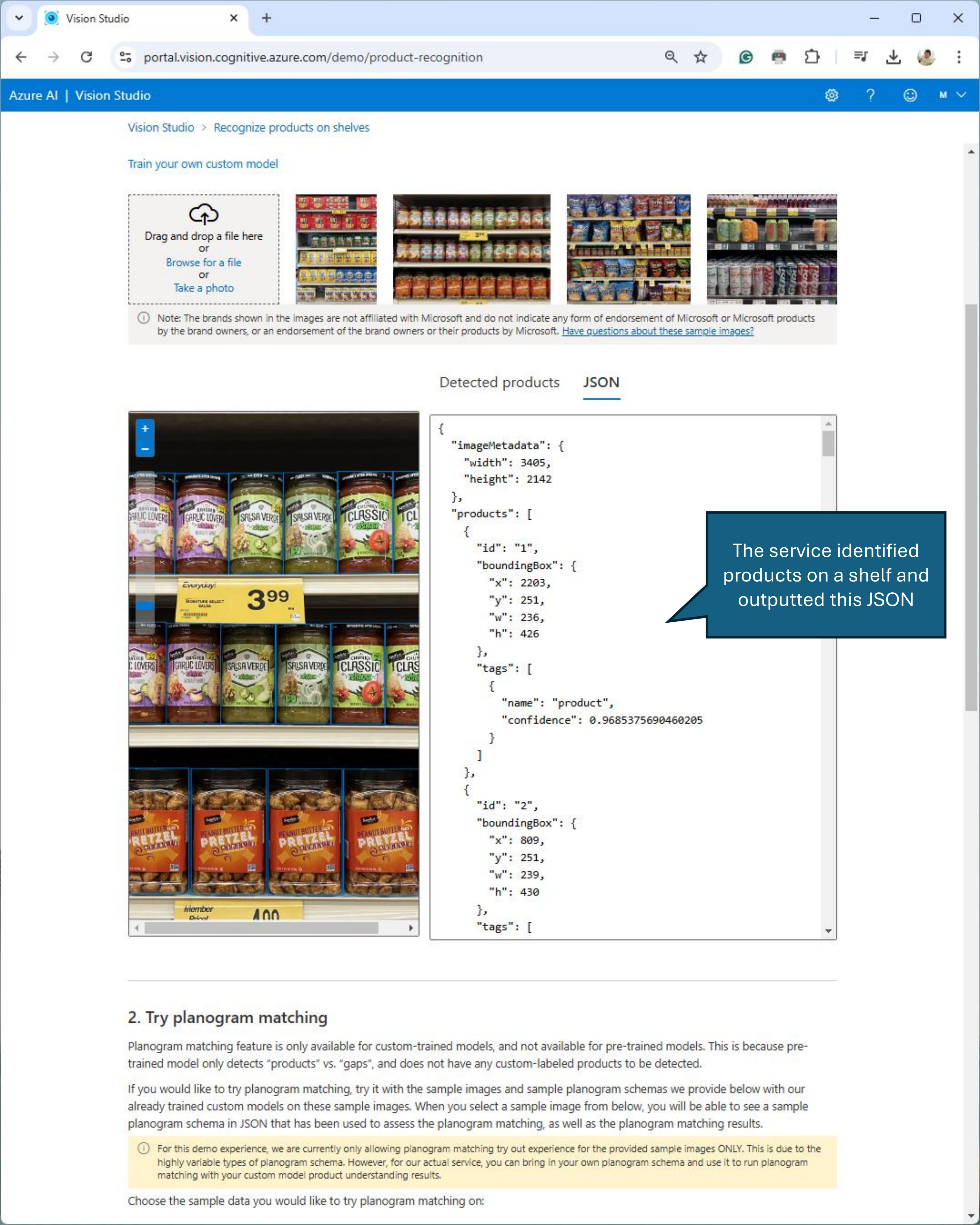
Extract printed and handwritten style text from images and documents for supported languages.

Try it out

Extract common tags from images

Create smart-cropped images

Detect faces in an image


Vision Studio
portal.vision.cognitive.azure.com/demo/product-recognition
Azure AI | Vision Studio
Vision Studio > Recognize products on shelves
Train your own custom model
Drag and drop a file here
or
Browse for a file
or
Take a photo
Detected products
JSON

ⓘ Note: The brands shown in the images are not affiliated with Microsoft and do not indicate any form of endorsement of Microsoft or Microsoft products by the brand owners, or an endorsement of the brand owners or their products by Microsoft. Have questions about these sample images?

```
{
  "imageMetadata": {
    "width": 3405,
    "height": 2142
  },
  "products": [
    {
      "id": "1",
      "boundingBox": {
        "x": 2203,
        "y": 251,
        "w": 236,
        "h": 426
      },
      "tags": [
        {
          "name": "product",
          "confidence": 0.9685375690460205
        }
      ]
    },
    {
      "id": "2",
      "boundingBox": {
        "x": 809,
        "y": 251,
        "w": 239,
        "h": 430
      },
      "tags": [
```

The service identified products on a shelf and outputted this JSON

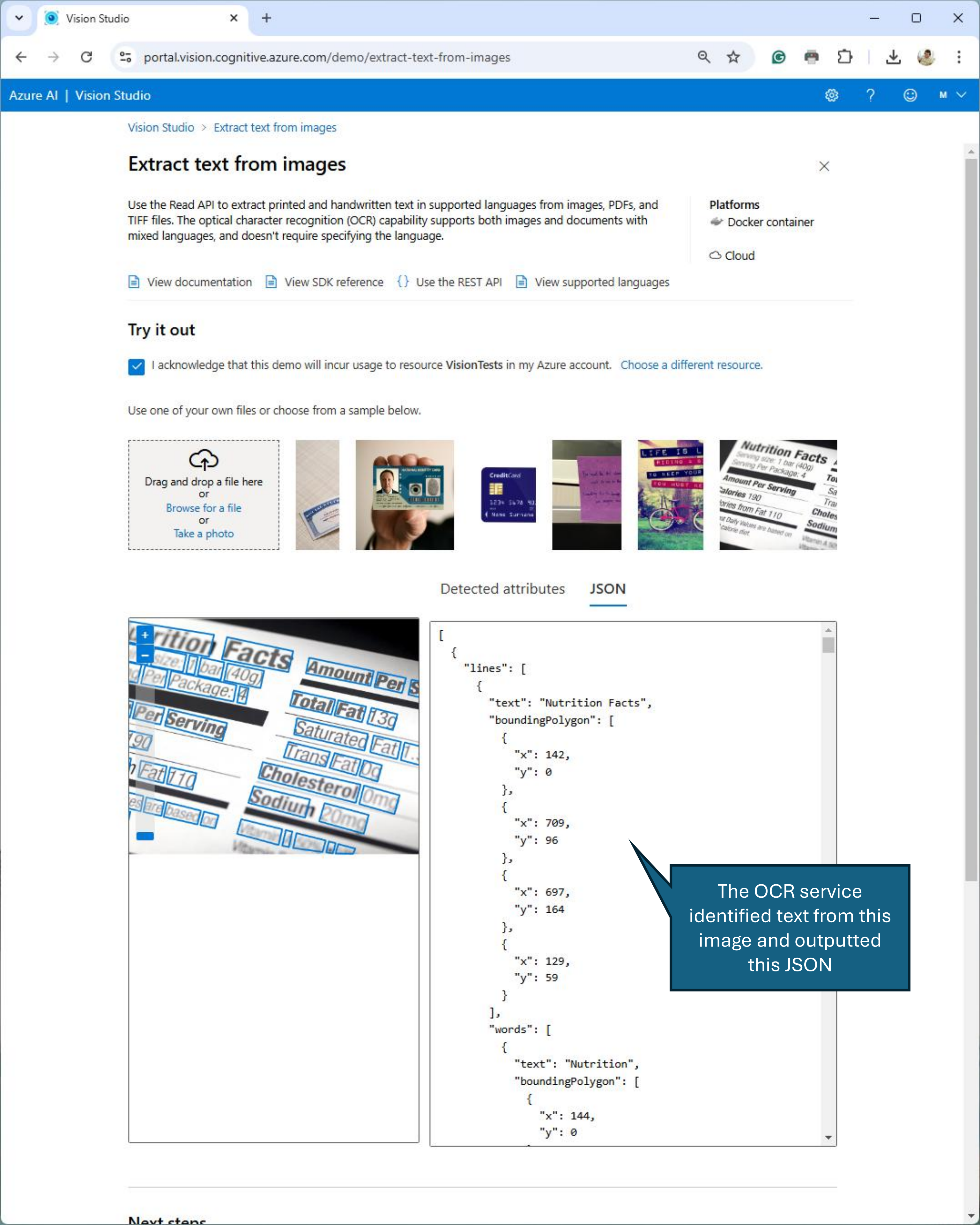
2. Try planogram matching

Planogram matching feature is only available for custom-trained models, and not available for pre-trained models. This is because pre-trained model only detects "products" vs. "gaps", and does not have any custom-labeled products to be detected.

If you would like to try planogram matching, try it with the sample images and sample planogram schemas we provide below with our already trained custom models on these sample images. When you select a sample image from below, you will be able to see a sample planogram schema in JSON that has been used to assess the planogram matching, as well as the planogram matching results.

ⓘ For this demo experience, we are currently only allowing planogram matching try out experience for the provided sample images ONLY. This is due to the highly variable types of planogram schema. However, for our actual service, you can bring in your own planogram schema and use it to run planogram matching with your custom model product understanding results.

Choose the sample data you would like to try planogram matching on:


Vision Studio
portal.vision.cognitive.azure.com/demo/extract-text-from-images
Azure AI | Vision Studio
M

Vision Studio > Extract text from images

Extract text from images

Use the Read API to extract printed and handwritten text in supported languages from images, PDFs, and TIFF files. The optical character recognition (OCR) capability supports both images and documents with mixed languages, and doesn't require specifying the language.

Platforms
Docker container
Cloud

View documentation | View SDK reference | Use the REST API | View supported languages

Try it out

☑ I acknowledge that this demo will incur usage to resource **VisionTests** in my Azure account. Choose a different resource.

Use one of your own files or choose from a sample below.

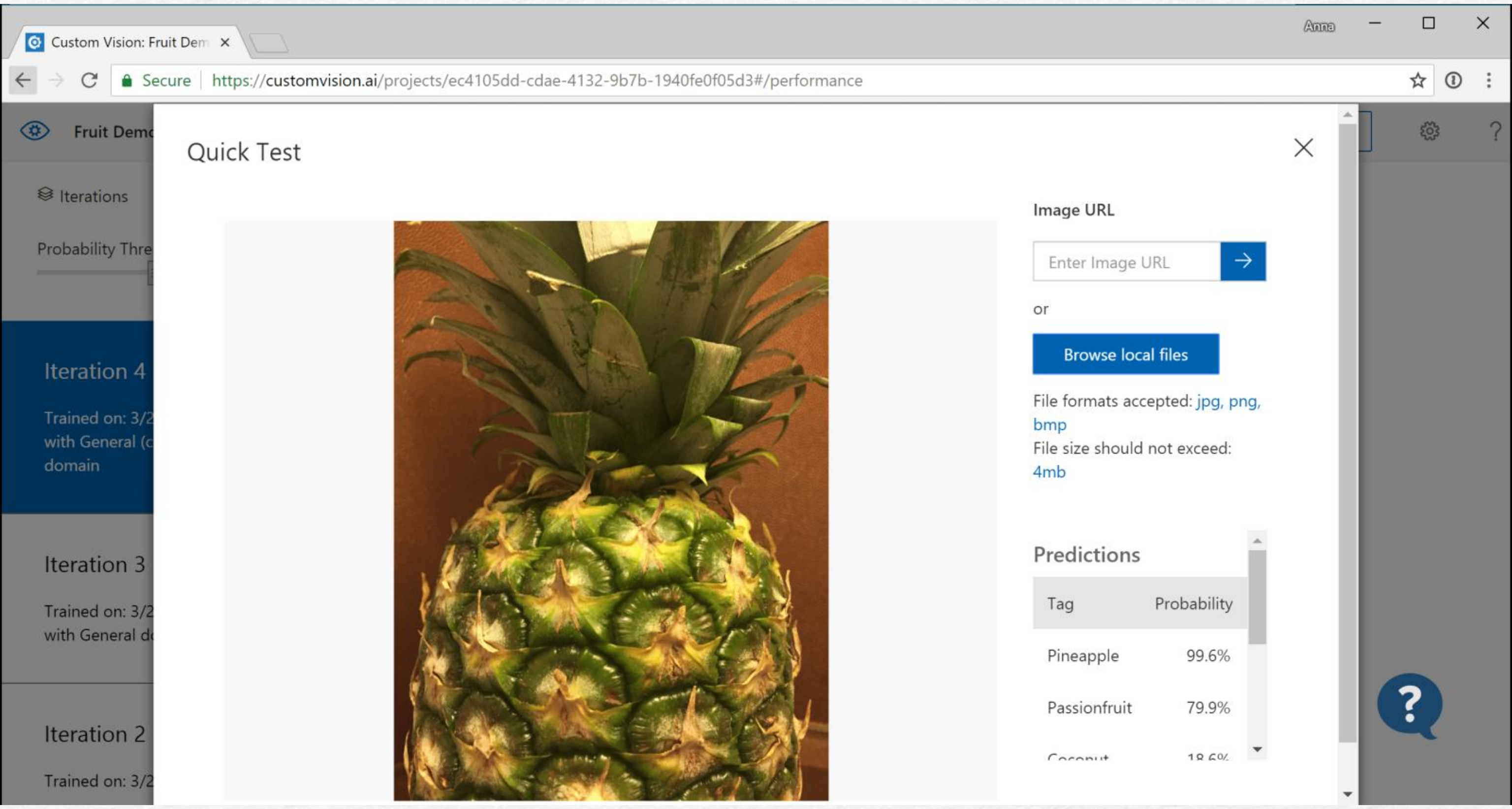
```
[
  {
    "lines": [
      {
        "text": "Nutrition Facts",
        "boundingPolygon": [
          {
            "x": 142,
            "y": 0
          },
          {
            "x": 709,
            "y": 96
          },
          {
            "x": 697,
            "y": 164
          },
          {
            "x": 129,
            "y": 59
          }
        ],
        "words": [
          {
            "text": "Nutrition",
            "boundingPolygon": [
              {
                "x": 144,
                "y": 0
```

The OCR service identified text from this image and outputted this JSON

Next steps

Custom Vision

Azure AI Custom Vision enables developers to build, train, and deploy custom image classification and object detection models tailored to specific use cases. Unlike pre-built models, Custom Vision allows users to create models that recognize objects and categories unique to their applications by uploading and labeling their own images.



Trained on: 3/2
with General (c
domain

Trained on: 3/2
with General d

Trained on: 3/2

File formats accepted: jpg, png, bmp
File size should not exceed: 4mb

Tag	Probability
Pineapple	99.6%
Passionfruit	79.9%
Coconut	18.6%

The Custom Vision service leverages a machine learning algorithm to analyze images for specific custom features. You provide sets of images that either contain or lack the visual characteristics you want to identify, labeling each image with custom tags during submission.

Key Capabilities of Azure AI Custom Vision

Image Classification. Identifies and categorizes images based on custom-defined labels.

Example: A manufacturer can use image classification to categorize defective and non-defective parts in an assembly line, automating quality control.

Object Detection. Detects and localizes multiple objects within an image, providing bounding boxes around detected objects.

Custom Training with Labeled Images. Users upload their own labeled datasets, allowing Custom Vision to learn to recognize specific items.

Example: Train the model to recognize different grades or ripeness levels of produce, such as apples or tomatoes, based on visual characteristics like color, size, and surface quality. This helps automate quality grading in food processing plants.

Iteration and Model Improvement. Allows for iterative training, where users can refine their model by uploading new data and retraining to improve accuracy.

Face

The Azure AI Face service is part of Azure's AI Services suite, specifically designed to detect, analyze, and identify human faces in images. This service provides a range of face recognition features that are useful for applications requiring facial analysis, such as verifying identity, organizing photos, analyzing demographics, or enhancing security.

Key Capabilities of Azure AI Face Service

Face Detection. Detects human faces in images and returns information such as face coordinates, bounding boxes, and rotation.

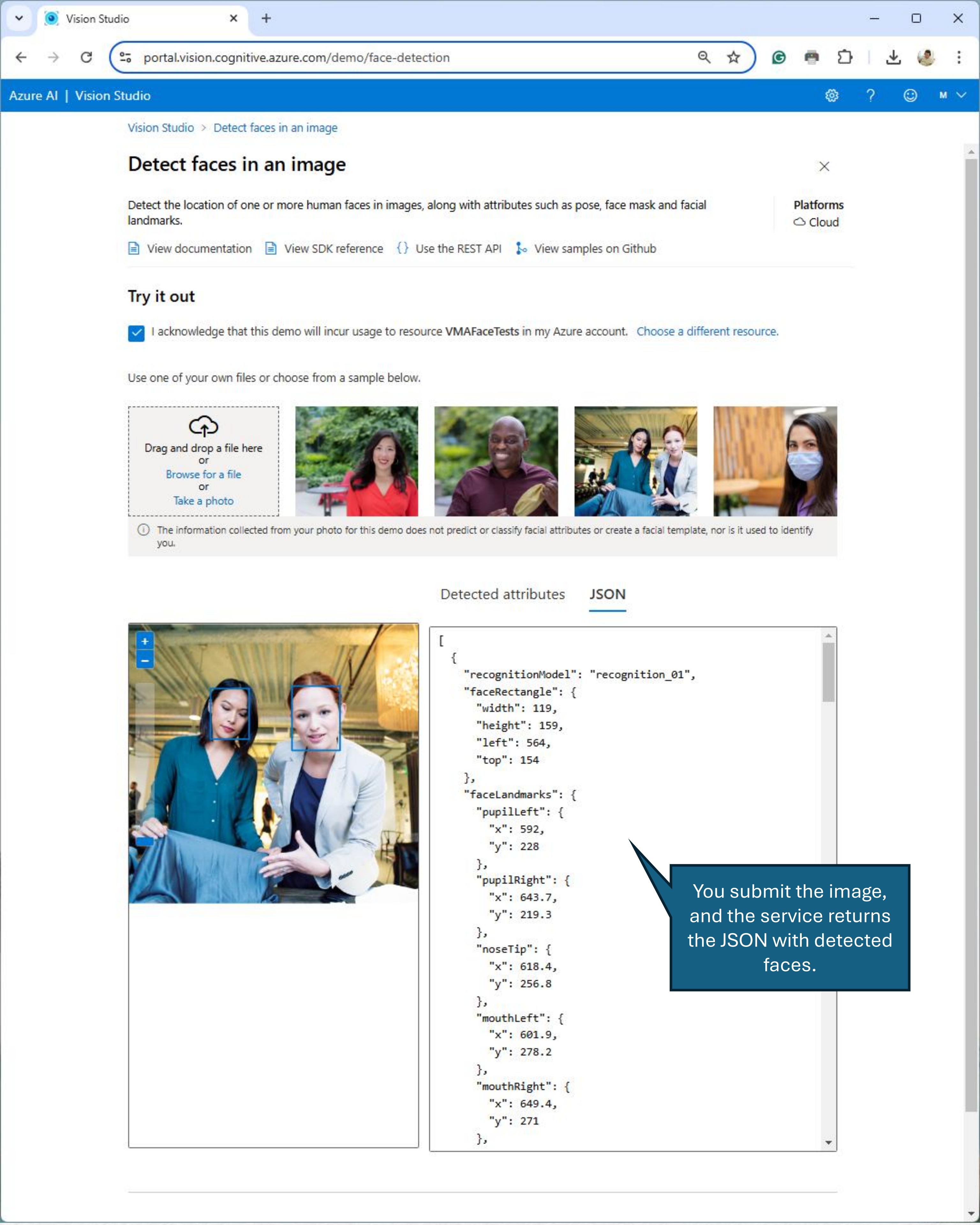
Face Attributes Analysis. Analyzes facial attributes, including age, gender, emotions, facial hair, and head pose.

Face Verification. Verifies whether two faces belong to the same person, commonly used in identity verification.

Face Identification. Identifies a face by comparing it with faces stored in a known group of people. This requires a training set of labeled faces.

Face Grouping. Groups similar faces together without knowing the identities, useful for organizing unlabeled faces.

Emotion Detection. Detects emotions like happiness, sadness, surprise, anger, and more.



Vision Studio > Detect faces in an image

Detect faces in an image

Detect the location of one or more human faces in images, along with attributes such as pose, face mask and facial landmarks.

Platforms
Cloud

Try it out

☑ I acknowledge that this demo will incur usage to resource **VMAFaceTests** in my Azure account. Choose a different resource.

Use one of your own files or choose from a sample below.

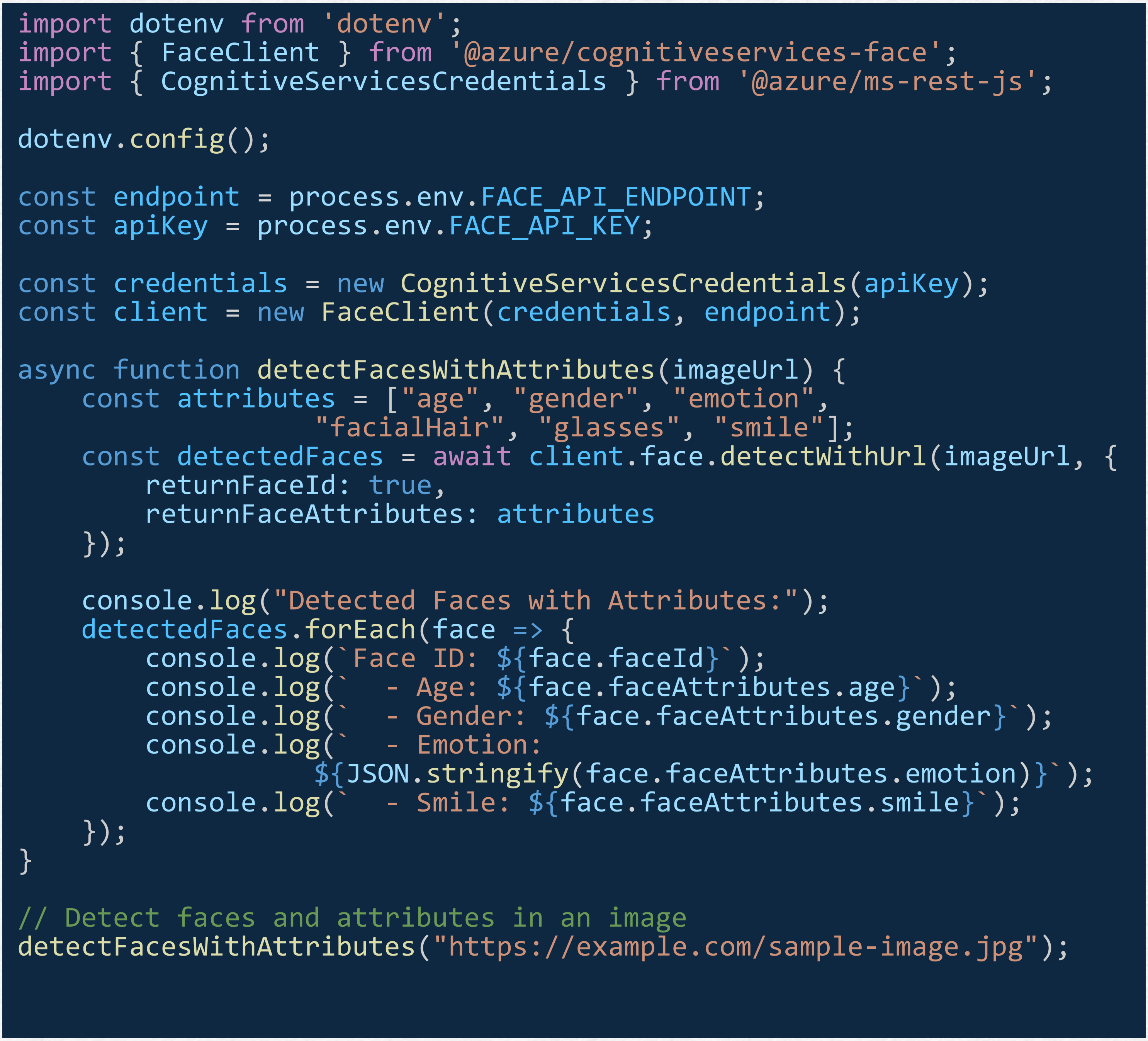
The information collected from your photo for this demo does not predict or classify facial attributes or create a facial template, nor is it used to identify you.

```
[
  {
    "recognitionModel": "recognition_01",
    "faceRectangle": {
      "width": 119,
      "height": 159,
      "left": 564,
      "top": 154
    },
    "faceLandmarks": {
      "pupilLeft": {
        "x": 592,
        "y": 228
      },
      "pupilRight": {
        "x": 643.7,
        "y": 219.3
      },
      "noseTip": {
        "x": 618.4,
        "y": 256.8
      },
      "mouthLeft": {
        "x": 601.9,
        "y": 278.2
      },
      "mouthRight": {
        "x": 649.4,
        "y": 271
      },
```

You submit the image, and the service returns the JSON with detected faces.

Face Detection and Attributes Analysis

This example demonstrates how to detect faces in an image and analyze basic attributes such as age, gender, and emotion.



```javascript
import dotenv from 'dotenv';
import { FaceClient } from '@azure/cognitiveservices-face';
import { CognitiveServicesCredentials } from '@azure/ms-rest-js';

dotenv.config();

const endpoint = process.env.FACE_API_ENDPOINT;
const apiKey = process.env.FACE_API_KEY;

const credentials = new CognitiveServicesCredentials(apiKey);
const client = new FaceClient(credentials, endpoint);

async function detectFacesWithAttributes(imageUrl) {
    const attributes = ["age", "gender", "emotion",
                  "facialHair", "glasses", "smile"];
    const detectedFaces = await client.face.detectWithUrl(imageUrl, {
        returnFaceId: true,
        returnFaceAttributes: attributes
    });

    console.log("Detected Faces with Attributes:");
    detectedFaces.forEach(face => {
        console.log(`Face ID: ${face.faceId}`);
        console.log(`  - Age: ${face.faceAttributes.age}`);
        console.log(`  - Gender: ${face.faceAttributes.gender}`);
        console.log(`  - Emotion:
                  ${JSON.stringify(face.faceAttributes.emotion)}`);
        console.log(`  - Smile: ${face.faceAttributes.smile}`);
    });
}

// Detect faces and attributes in an image
detectFacesWithAttributes("https://example.com/sample-image.jpg");
```

Translator

Azure AI Translator is another Azure AI service that enables developers to integrate language translation capabilities into applications. It supports real-time or batch translation across over 100 languages and dialects, making it useful for a wide range of scenarios, such as multilingual customer support, global communication, content localization, and website translation.

Key Features of Azure AI Translator

Text Translation. Translates text from one language to another in real-time or in batches.

Document Translation. Translates entire documents while preserving their original layout and format.

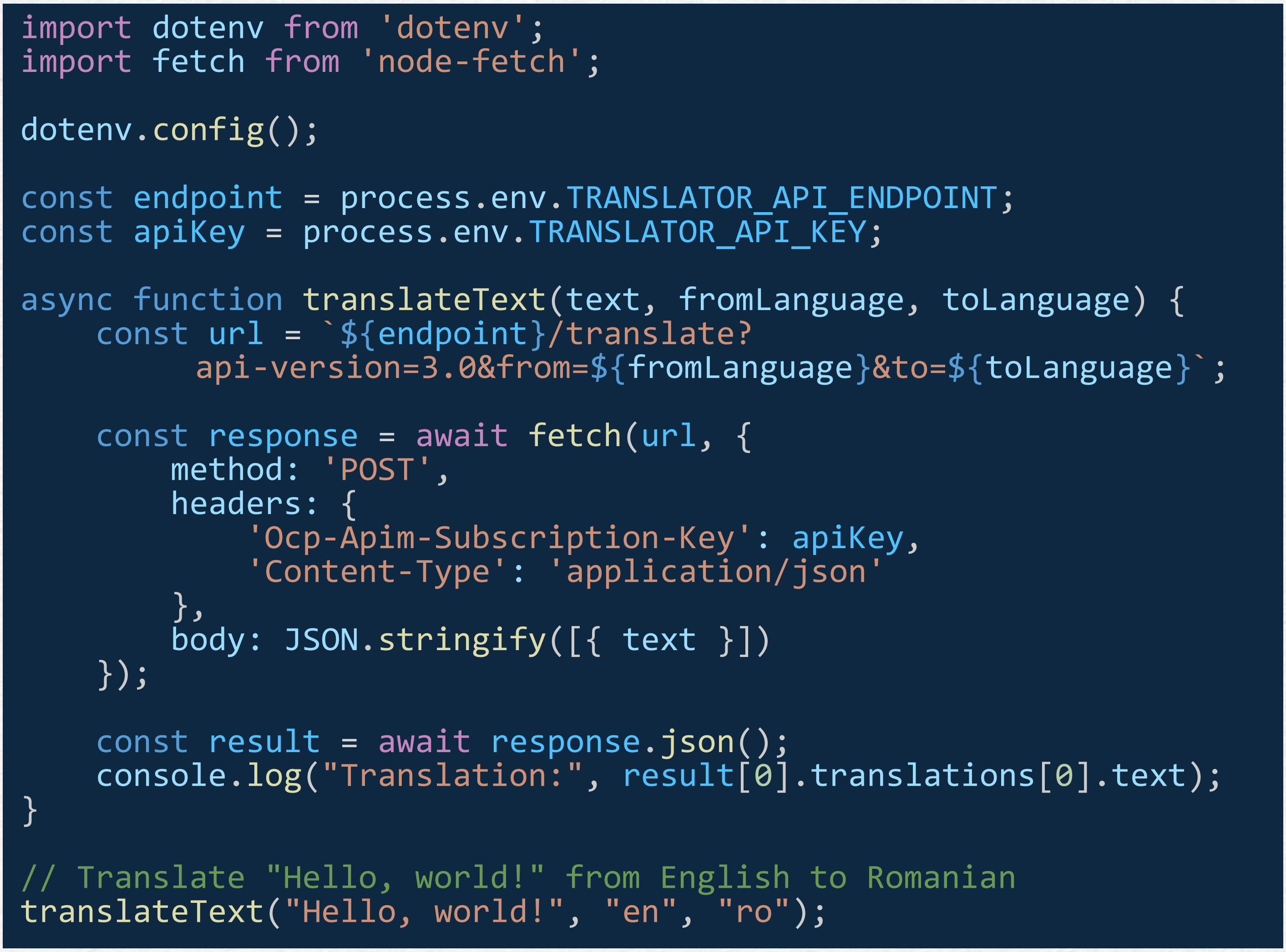
Speech Translation. Translates spoken language in real-time, allowing people to communicate verbally across language barriers.

Transliteration. Converts text from one script to another within the same language (e.g., Hindi written in Latin script).

Language Detection. Detects the language of input text automatically, which is useful when the language isn’t specified.

Example of using Text Translation

This example demonstrates how to translate text from one language to another.



```
import dotenv from 'dotenv';
import fetch from 'node-fetch';

dotenv.config();

const endpoint = process.env.TRANSLATOR_API_ENDPOINT;
const apiKey = process.env.TRANSLATOR_API_KEY;

async function translateText(text, fromLanguage, toLanguage) {
    const url = `${endpoint}/translate?
         api-version=3.0&from=${fromLanguage}&to=${toLanguage}`;

    const response = await fetch(url, {
        method: 'POST',
        headers: {
            'Ocp-Apim-Subscription-Key': apiKey,
            'Content-Type': 'application/json'
        },
        body: JSON.stringify([{ text }])
    });

    const result = await response.json();
    console.log("Translation:", result[0].translations[0].text);
}

// Translate "Hello, world!" from English to Romanian
translateText("Hello, world!", "en", "ro");
```

Language

Azure AI Language is a cloud-based service in the Azure AI Services suite that provides natural language processing (NLP) capabilities.

Key Capabilities of Azure AI Language Service

Sentiment Analysis. Determines the sentiment of text (positive, neutral, or negative), along with the sentiment score.

Named Entity Recognition (NER). Recognizes entities like names, organizations, dates, and locations within text, categorizing them into predefined types.

Language Detection. Identifies the language of a given text, which is useful for multilingual applications.

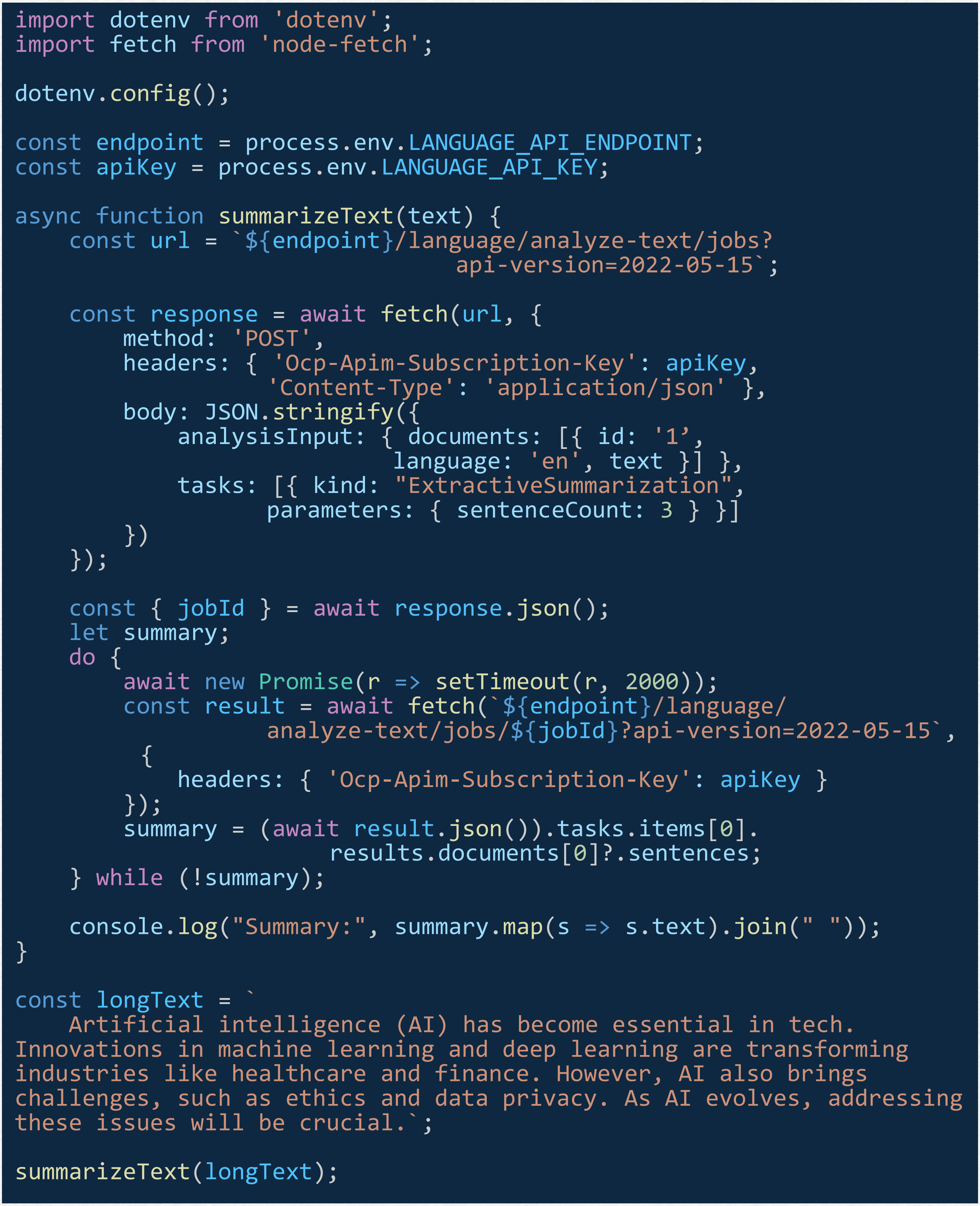
Key Phrase Extraction. Identifies the main points or key phrases in a text.

Text Summarization. Provides concise summaries of long documents, helping readers get the main points quickly.

Question Answering. Finds answers to questions based on a set of documents or a knowledge base.

Conversational Language Understanding. Understands intent and extracts relevant information from conversations, which is useful for building intelligent chatbots.

Example of using Text Summarization from Azure AI Language service



```
import dotenv from 'dotenv';
import fetch from 'node-fetch';

dotenv.config();

const endpoint = process.env.LANGUAGE_API_ENDPOINT;
const apiKey = process.env.LANGUAGE_API_KEY;

async function summarizeText(text) {
    const url = `${endpoint}/language/analyze-text/jobs?
                                 api-version=2022-05-15`;

    const response = await fetch(url, {
        method: 'POST',
        headers: { 'Ocp-Apim-Subscription-Key': apiKey,
                   'Content-Type': 'application/json' },
        body: JSON.stringify({
            analysisInput: { documents: [{ id: '1’,
                            language: 'en', text }] },
            tasks: [{ kind: "ExtractiveSummarization",
                   parameters: { sentenceCount: 3 } }]
        })
    });

    const { jobId } = await response.json();
    let summary;
    do {
        await new Promise(r => setTimeout(r, 2000));
        const result = await fetch(`${endpoint}/language/
                   analyze-text/jobs/${jobId}?api-version=2022-05-15`,
         {
            headers: { 'Ocp-Apim-Subscription-Key': apiKey }
        });
        summary = (await result.json()).tasks.items[0].
                       results.documents[0]?.sentences;
    } while (!summary);

    console.log("Summary:", summary.map(s => s.text).join(" "));
}

const longText = `
    Artificial intelligence (AI) has become essential in tech.
Innovations in machine learning and deep learning are transforming
industries like healthcare and finance. However, AI also brings
challenges, such as ethics and data privacy. As AI evolves, addressing
these issues will be crucial.`;

summarizeText(longText);
```

Video Indexer

Azure AI Video Indexer is a service in the Azure AI Services suite that provides advanced video analysis capabilities. This service is especially useful for media companies, broadcasters, and content creators.

Key Capabilities of Azure AI Video Indexer

Speech-to-Text and Transcription. Transcribes spoken words into text, making it easy to search through video content by dialogue.

Face and Emotion Recognition. Detects and identifies faces, along with emotions such as happiness, sadness, or anger.

Scene and Shot Detection. Recognizes changes in scenes and shots, allowing the video to be segmented into logical parts.

Object and Activity Detection. Identifies objects and actions within the video, like "car," "running," or "applause."

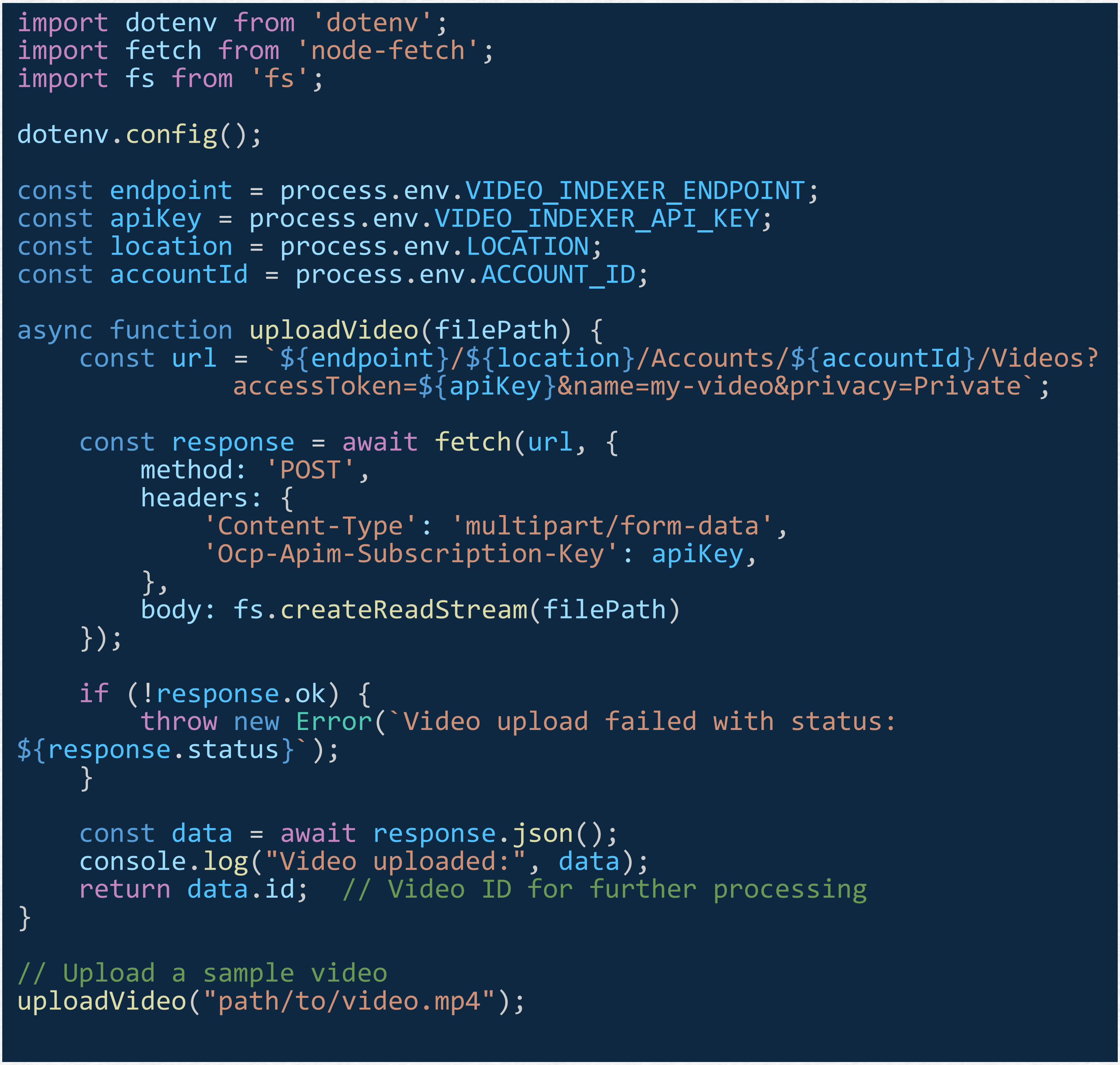
Topic and Sentiment Analysis. Extracts topics discussed in the video and analyzes sentiment based on the text and audio content.

Optical Character Recognition (OCR). Detects and extracts text displayed in videos, such as subtitles, captions, or on-screen information.

Keyword Extraction and Named Entity Recognition (NER). Extracts keywords and named entities such as people, locations, and organizations mentioned in the video.

Example – Part I: Upload a video

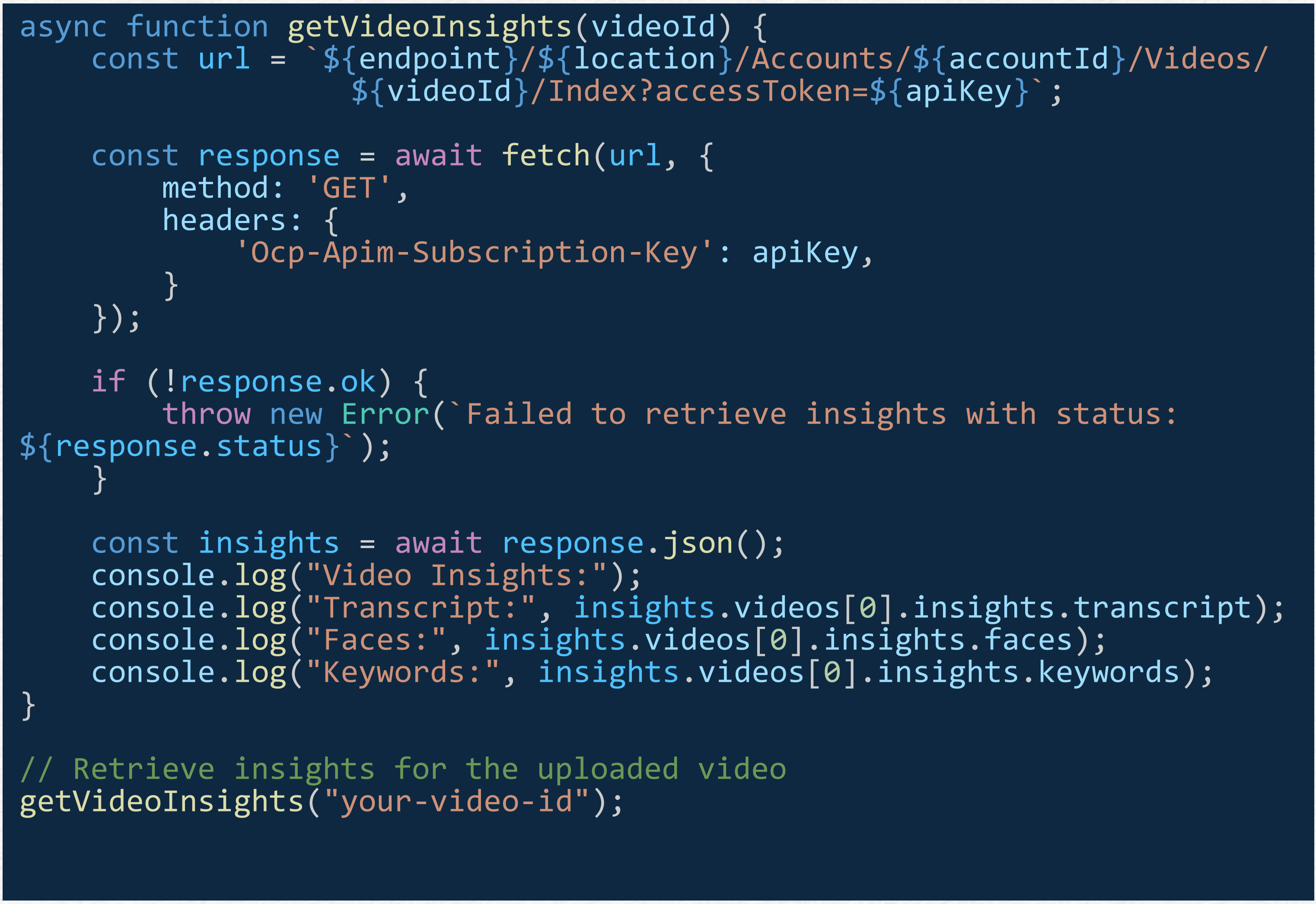
The first step in using Video Indexer is to upload a video for analysis.



```
import dotenv from 'dotenv';
import fetch from 'node-fetch';
import fs from 'fs';

dotenv.config();

const endpoint = process.env.VIDEO_INDEXER_ENDPOINT;
const apiKey = process.env.VIDEO_INDEXER_API_KEY;
const location = process.env.LOCATION;
const accountId = process.env.ACCOUNT_ID;

async function uploadVideo(filePath) {
    const url = `${endpoint}/${location}/Accounts/${accountId}/Videos?
              accessToken=${apiKey}&name=my-video&privacy=Private`;

    const response = await fetch(url, {
        method: 'POST',
        headers: {
            'Content-Type': 'multipart/form-data',
            'Ocp-Apim-Subscription-Key': apiKey,
        },
        body: fs.createReadStream(filePath)
    });

    if (!response.ok) {
        throw new Error(`Video upload failed with status:
${response.status}`);
    }

    const data = await response.json();
    console.log("Video uploaded:", data);
    return data.id;  // Video ID for further processing
}

// Upload a sample video
uploadVideo("path/to/video.mp4");
```

Example – Part II: Retrieve Insights from a Video

After uploading the video, you can retrieve various insights, such as transcription, face recognition, and scene segmentation.



```
async function getVideoInsights(videoId) {
    const url = `${endpoint}/${location}/Accounts/${accountId}/Videos/
                   ${videoId}/Index?accessToken=${apiKey}`;

    const response = await fetch(url, {
        method: 'GET',
        headers: {
            'Ocp-Apim-Subscription-Key': apiKey,
        }
    });

    if (!response.ok) {
        throw new Error(`Failed to retrieve insights with status:
${response.status}`);
    }

    const insights = await response.json();
    console.log("Video Insights:");
    console.log("Transcript:", insights.videos[0].insights.transcript);
    console.log("Faces:", insights.videos[0].insights.faces);
    console.log("Keywords:", insights.videos[0].insights.keywords);
}

// Retrieve insights for the uploaded video
getVideoInsights("your-video-id");
```

Output: The transcript, recognized faces, and extracted keywords from the video.

Use Case: Useful for building searchable metadata for video content, making it easy to find specific moments or people.

Immersive Reader

Azure AI Immersive Reader is a service that provides reading and comprehension support, making it easier for users to engage with and understand text. It's designed to help improve reading comprehension, especially for individuals with dyslexia, visual impairments, or language processing difficulties. Immersive Reader is valuable for educational applications, e-learning platforms, and any application aiming to improve accessibility.

Key Capabilities of Azure AI Immersive Reader

Text-to-Speech. Converts text into spoken audio, allowing users to listen to content.

Word and Line Highlighting. Highlights text as it's read, making it easier for users to follow along.

Translation. Translates text into different languages to support multilingual users.

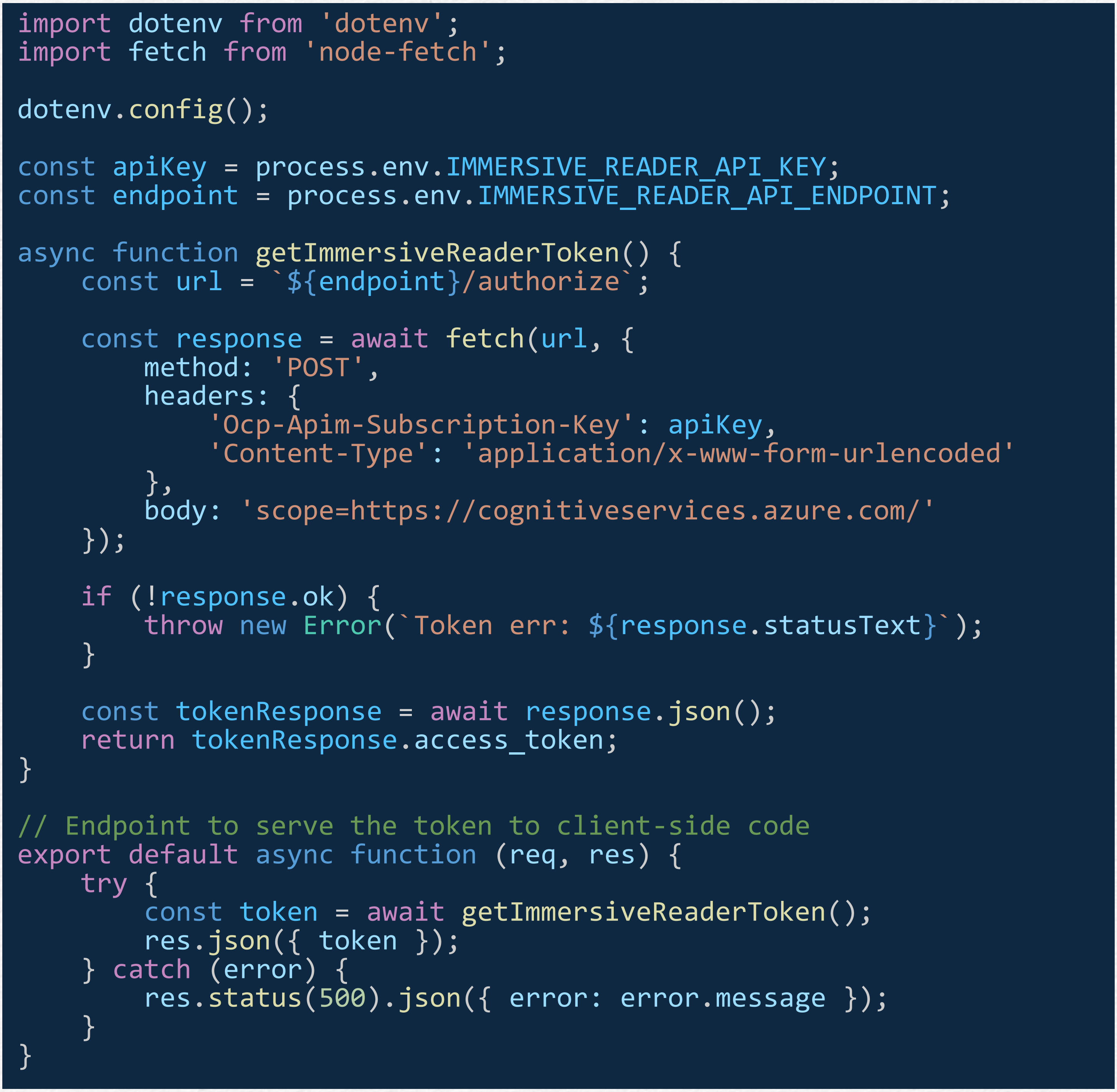
Grammar Support. Breaks down words by syllables and identifies parts of speech like nouns, verbs, adjectives, and adverbs.

Line Focus. Focuses on one or a few lines at a time, reducing visual clutter and improving readability.

Picture Dictionary. Displays images for certain words, which helps users better understand the meaning.

Example - Part I: Basic server endpoint

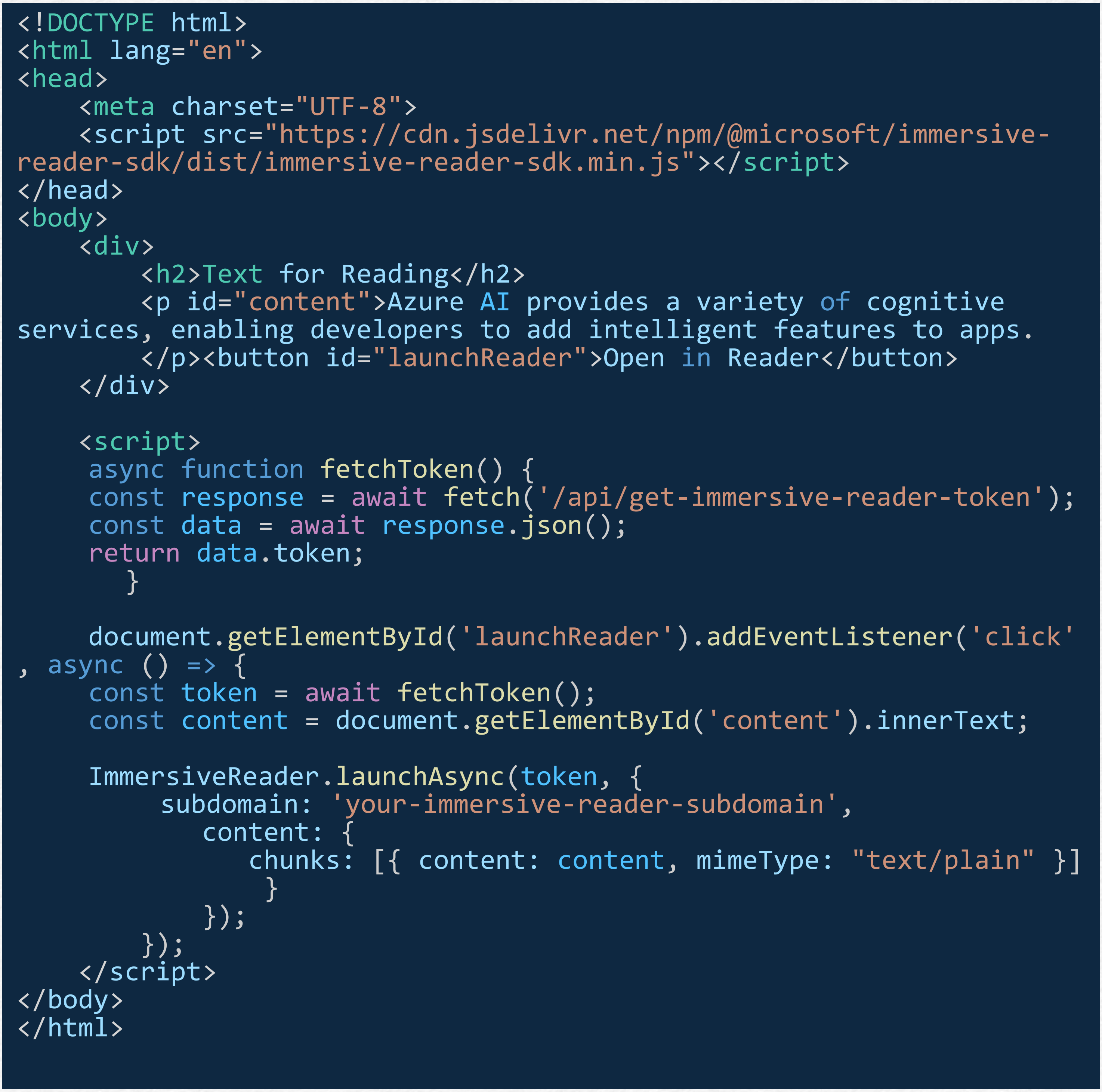
In this example, we'll integrate Immersive Reader into a web application, where users can highlight and read aloud a piece of text. Here's a basic server endpoint to get a token:



```
import dotenv from 'dotenv';
import fetch from 'node-fetch';

dotenv.config();

const apiKey = process.env.IMMERSIVE_READER_API_KEY;
const endpoint = process.env.IMMERSIVE_READER_API_ENDPOINT;

async function getImmersiveReaderToken() {
    const url = `${endpoint}/authorize`;

    const response = await fetch(url, {
        method: 'POST',
        headers: {
            'Ocp-Apim-Subscription-Key': apiKey,
            'Content-Type': 'application/x-www-form-urlencoded'
        },
        body: 'scope=https://cognitiveservices.azure.com/'
    });

    if (!response.ok) {
        throw new Error(`Token err: ${response.statusText}`);
    }

    const tokenResponse = await response.json();
    return tokenResponse.access_token;
}

// Endpoint to serve the token to client-side code
export default async function (req, res) {
    try {
        const token = await getImmersiveReaderToken();
        res.json({ token });
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
}
```

Example - Part II: Client-side

Use the token in the client-side to initialize Immersive Reader.
The token and text content are passed to Immersive Reader for text-to-speech, highlighting, and more.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <script src="https://cdn.jsdelivr.net/npm/@microsoft/immersive-
reader-sdk/dist/immersive-reader-sdk.min.js"></script>
</head>
<body>
    <div>
        <h2>Text for Reading</h2>
        <p id="content">Azure AI provides a variety of cognitive
services, enabling developers to add intelligent features to apps.
        </p><button id="launchReader">Open in Reader</button>
    </div>

    <script>
     async function fetchToken() {
     const response = await fetch('/api/get-immersive-reader-token');
     const data = await response.json();
     return data.token;
       }

     document.getElementById('launchReader').addEventListener('click'
, async () => {
     const token = await fetchToken();
     const content = document.getElementById('content').innerText;

     ImmersiveReader.launchAsync(token, {
         subdomain: 'your-immersive-reader-subdomain',
            content: {
               chunks: [{ content: content, mimeType: "text/plain" }]
                }
            });
        });
    </script>
</body>
</html>
```

From cloud to on-prem

Azure AI services offer Docker containers (for select Azure AI services) that allow you to run the same APIs available in Azure within your own on-premises environment. These containers provide the flexibility to bring Azure AI capabilities closer to your data, supporting needs around compliance, security, and operational requirements.

Metering information

The Azure AI containers are required to submit metering information for billing purposes.

The host should allowlist port 443 and the following domains:

*.cognitive.microsoft.com
*.cognitiveservices.azure.com

Further reading on containers

Container support is currently available for a subset of Azure AI services.

See page: https://learn.microsoft.com/en-us/azure/ai-services/cognitive-services-container-support

Marian Veteanu
Technology Architect and Product Leader

Excited to join an organization where I can make an impact!

Let’s connect and explore opportunities—message me!

https://www.linkedin.com/in/mveteanu/
https://x.com/mveteanu