Docu ent AI

Introduction

This repository contains a collection of AWS La bda APIs that uses penAI's GPT-4 chat odels to extract infor ation fro docu ents using a predefined JS N sche a. The API uses `gpt-4-1106-preview` odel for `TXT` docu ents and `gpt-4-vision-preview` for `JPG/PNG/TIFF/BMP` docu ents. For `PDF` docu ents the API converts the PDF to i ages and uses the `gpt-4-vision-preview` odel. For ore infor ation on the odels, see penAI's API docu entation.

The application is deployed using AWS CDK and the following AWS services are used:

- `API Gateway` - anage API endpoints
- `Lambda` - anage sche a and extract data
- `DynamoDB` - store sche a, extracted data, and request status
- `S3` - store docu ents to be processed
- `SQS` - anage asynchronous processing of docu ents
- `Secrets Manager` - anage penAI API key
- `Systems Manager Parameter Store` - store application configuration and AWS resource na es
- `IAM` - anage per issions for AWS resources

All the La bda functions share a co on layer contained in the `src/layers/docai` folder, and the La bda functions are contained in the `src/functions` folder. While the CDK application is contained in the `resources` folder.

Dependencies

The following

- `OSX`, `Linux` or `Windows`
- Python 3.12
- Node 20.x
- AWS CDK 2.118.9
- Poetry 1.5.0

Python Dependencies

- boto3
- aws-la bda-powertools[parser,tracer]
- titoken
- jsonsche a
- openai
- py updf
- aws-cdk-lib

Python Develop ent Dependencies

- isort

- ruff
- pre-co it
- black
- ypy

Installation

The `docai` layer can be installed for develop ent purposes using `poetry`. Use the following co and to install the layer for local develop ent

```
cd src/layers/docai && poetry install
```

Deploy ent

The co and below synthesizes the CDK application and deploys all application resources to AWS

```
make deploy
```

API Usage

To use the API, you will need to create an API key in the API Gateway console. The API key should be passed in the `x-api-key` header of the request. In addition, the `<api-id>` should be replaced with the API ID of the deployed API.

Sche a Manage ent

A sche a is a JS N sche a that defines the structure of the data that to be extracted fro a docu ent. The description nodes of fields are used to provide pro pt instructions, hints, and exa ples to the odel on what and how to extract structured data fro the docu ent.

Create Schema

This endpoint creates a sche a that can be used to extract infor ation fro a docu ent.

```
curl -X POST \
    -H "Content-Type: application/json" \
    -H "x-api-key:<api-key>" \
    -d @sample.json \
    https://<api-id>.execute-api.ca-centra-1.amazonaws.com/prod/create-
schema
```

Sample Request Data

```
```

```
{
  "schema_name": "test_loe",
  "schema_description": "Test extracts data from letter of employment",
  "schema_definition": {
    "$id": "https://example.com/test_loe.schema.json",
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "title": "Employment Letter",
    "description": "A schema that represents data extracted from an
employment letter",
    "type": "object",
    "properties": {
      "employer_name": {
        "description": "The name of the employer",
        "type": ["null", "string"]
      }
    },
    "required": ["employer_name"],
    "additionalProperties": false
  }
}
```

Sample Response Data

```
{
  "OK": true,
  "result": {
    "schema_name": "test_loe",
    "schema_version": "rvByviQNVb",
    "number_of_tokens": 102
  }
}
```

Get Schema

This endpoint returns the sche a definition for a given sche a na e and version.

```
curl -X POST \
    -H "Content-Type: application/json" \
    -H "x-api-key:<api-key>" \
    -d @sample.json \
    https://<api-id>.execute-api.ca-centra-1.amazonaws.com/prod/get-schema
```

Sample Request Data

```
{
  "schema_name": "test_loe",
```

```
  "schema_version": "rvByviQNVb"
}
```

Sample Response Data

```
{
  "OK": true,
  "result": {
    "schema_name": "test_loe",
    "schema_description": "Test extracts data from letter of employment",
    "schema_definition": {
      "$schema": "https://json-schema.org/draft/2020-12/schema",
      "description": "A schema that represents data extracted from an
employment letter",
      "additionalProperties": false,
      "title": "Employment Letter",
      "type": "object",
      "properties": {
        "employer_name": {
          "type": ["null", "string"],
          "description": "The name of the employer"
        }
      },
      "required": ["employer_name"],
      "$id": "https://example.com/test_loe.schema.json"
    },
    "schema_version": "rvByviQNVb",
    "schema_status": "ACTIVE",
    "number_of_tokens": 102,
    "created_at": "2024-01-06T02:09:28.036892"
  }
}
```

List Schemas

This endpoint returns a list of sche as that have been created for a given sche a na e.

```
curl -X POST \
    -H "Content-Type: application/json" \
    -H "x-api-key:<api-key>" \
    -d @sample.json \
    https://<api-id>.execute-api.ca-centra-1.amazonaws.com/prod/list-
schema
```

Sample Request Data

```
                    "biweekly",
                    "weekly",
                    "hourly"
                  ]
                }
              },
              "required": ["amount", "frequency"]
            }
          },
          "required": ["employer_name", "employee_name",
"employee_salary"]
        }
      },
      {
        "schema_description": "Test extracts data from letter of
employment",
        "created_at": "2024-01-05T17:18:35.254293",
        "number_of_tokens": 234,
        "schema_name": "test_loe",
        "schema_version": "Gcpgmg1c79",
        "schema_status": "ACTIVE",
        "schema_definition": {
          "description": "Schema to extract the employer name, employee
name, and employee salary from a letter of employment",
          "additionalProperties": false,
          "$schema": "http://json-schema.org/draft/2020-12/schema",
          "type": "object",
          "properties": {
            "employer_name": {
              "type": ["string", "null"],
              "description": "The name of the employer in the letter of
employment"
            },
            "employee_name": {
              "type": ["string", "null"],
              "description": "The name of the employee in the letter of
employment"
            },
            "employee_salary": {
              "description": "The salary information of the employee in
the letter of employment",
              "type": "object",
              "properties": {
                "amount": {
                  "type": ["number", "null"],
                  "description": "The amount of the salary"
                },
                "frequency": {
                  "type": ["string", "null"],
                  "description": "The frequency of the salary payment",
                  "enum": [
                    "annually",
                    "monthly",
                    "biweekly",
```

```
                    "weekly",
                    "hourly"
                  ]
                }
              },
              "required": ["amount", "frequency"]
            }
          },
          "required": ["employer_name", "employee_name",
"employee_salary"]
        }
      }
    ]
  }
}
```

Delete Schema

This endpoint deletes a sche a for a given sche a na e and version.

```
curl -X POST \
    -H "Content-Type: application/json" \
    -H "x-api-key:<api-key>" \
    -d @sample.json \
    https://<api-id>.execute-api.ca-centra-1.amazonaws.com/prod/delete-
schema
```

Sample Request Data

```
{
  "schema_name": "test_loe",
  "schema_version": "rvByviQNVb"
}
```

Sample Response Data

```
{
  "OK": true,
  "result": {
    "message": "Schema deleted successfully"
  }
}
```

Data Extraction

These endpoints are used to extract data fro docu ents using a predefined sche a. For `image/*` and `application/pdf` docu ents, the content is first converted to a `base64` string and then passed to the API. The API will then convert the `base64` content to an i age and then extract the data fro the i age. For `text/plain` docu ents, the content is passed directly to the API.

Extract Data

This endpoint extracts data fro a docu ent using a predefined sche a. It process docu ent in an `ONLINE` anner, eaning that the docu ent is processed when the API is called. The API will return the extracted data in the response.

```
curl -X POST \
    -H "Content-Type: application/json" \
    -H "x-api-key:<api-key>" \
    -d @sample.json \
    https://<api-id>.execute-api.ca-centra-1.amazonaws.com/prod/extract-
data
```

Sample Request Data

```
{
  "schema_name": "test_loe",
  "schema_version": "rvByviQNVb",
  "content": "This is a letter of employment for John Smith. John Smith is
employed by Acme Inc. John Smith's salary is $100,000.00 per year.",
  "mime_type": "text/plain"
}
```

Sample Response Data

```
{
  "OK": true,
  "result": {
    "request_id": "d4e0fbfb-ac56-4166-9f50-9ad0ba592f9f",
    "data": {
      "employer_name": "Acme Inc."
    }
  }
}
```

The `request_id` can be used to retrieve the extracted data using `/get-result` endpoint if the result is required at a later ti e.

Extract Data Batch

This endpoint extracts data fro docu ents using a predefined sche a in a batch ode. The request is processed asynchronously and the extracted data can be retrieved using the `/get-result` endpoint.

```
curl -X POST \
    -H "Content-Type: application/json" \
    -H "x-api-key:<api-key>" \
    -d @sample.json \
    https://<api-id>.execute-api.ca-centra-1.amazonaws.com/prod/extract-
data-batch
```

Sample Request Data

```
{
  "schema_name": "test_loe",
  "schema_version": "rvByviQNVb",
  "content": "This is a letter of employment for John Smith. John Smith is
employed by Acme Inc. John Smith's salary is $100,000.00 per year.",
  "mime_type": "text/plain"
}
```

Sample Response Data

```
{
  "OK": true,
  "result": {
    "request_id": "97e5e420-17e1-4e12-bd8c-8d4771e6f3b5",
    "status": "QUEUED",
    "created_at": "2024-01-06T02:31:19.891472"
  }
}
```

Get Result

This endpoint retrieves the extracted data for a given `request_id`. The `request_id` fro an `/extract-data` or `/extract-data-batch` request can be used to retrieve the extracted data.

```
curl -X POST \
    -H "Content-Type: application/json" \
    -H "x-api-key:<api-key>" \
    -d @sample.json \
    https://<api-id>.execute-api.ca-centra-1.amazonaws.com/prod/get-result
```

Sample Request Data

```
{
  "request_id": "97e5e420-17e1-4e12-bd8c-8d4771e6f3b5"
}
```

Sample Response Data

```
{
  "OK": true,
  "result": {
    "created_at": "2024-01-06T02:31:25.974506",
    "request_id": "97e5e420-17e1-4e12-bd8c-8d4771e6f3b5",
    "status": "COMPLETED",
    "data": {
      "employer_name": "Acme Inc."
    }
  }
}
```