ITIS 6177 System Integration Final Project Documentation

Project Topic

The project was to consume third party vendor APIs provided by Microsoft Azure Cognitive Services. The project utilizes Azure's following Cognitive Services:

- Entity Linking
- Language Detection
- PII
- Question and Answers

Swagger Documentation: http://164.92.106.251:5000/api/docs/

Server URL : http://164.92.106.251:5000/

Tools Required

- Postman

Description for all the services used

Entity Linking

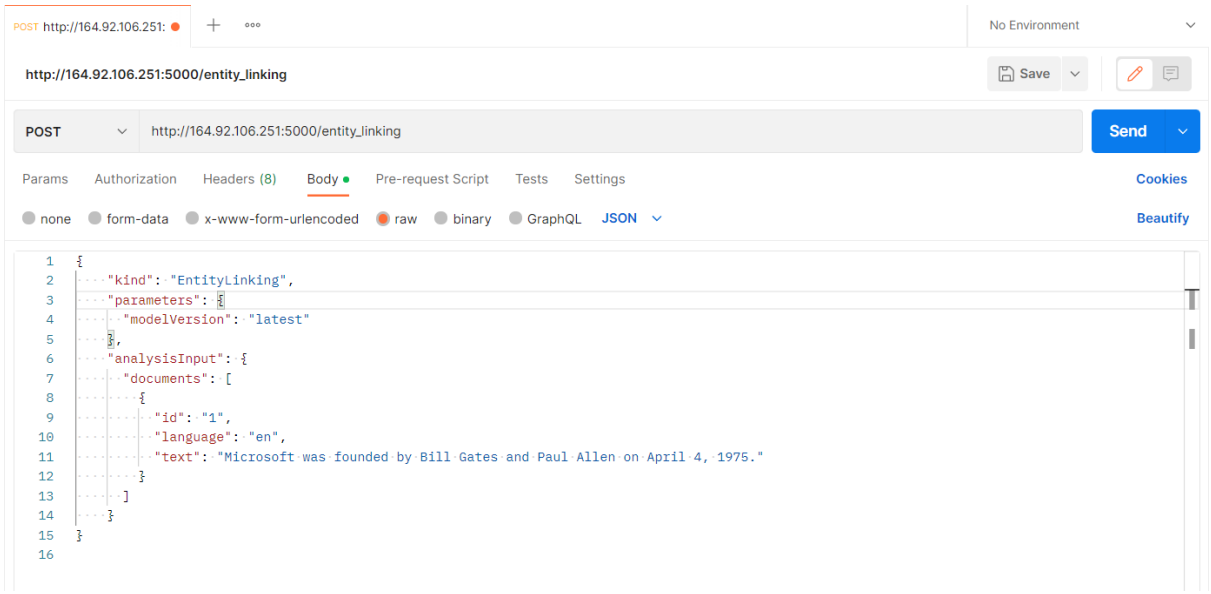
Entity linking is one of the features offered by Azure Cognitive Service for Language, a collection of machine learning and AI algorithms in the cloud for developing intelligent applications that involve written language. Entity linking identifies and disambiguates the identity of entities found in text. For example, in the sentence "We went to Seattle last week.", the word "Seattle" would be identified, with a link to more information on Wikipedia.

To run the following API in Postman,

Copy **`http://164.92.106.251:5000/entity_linking`** in the address bar and change the method to POST and add the following body as JSON:

```
{
    "kind": "EntityLinking",
    "parameters": {
      "modelVersion": "latest"
    },
    "analysisInput": {
      "documents": [
        {
          "id": "1",
          "language": "en",
          "text": "Microsoft was founded by Bill Gates and Paul Allen on April 4
, 1975."
        }
      ]
    }
}
```

Your Postman window should look like this:



Click Send to run it and you will receive the following response for the above body:

Body Cookies Headers (4) Test Results Status: 200 OK Time: 620 ms Size: 748 B Save Response

Pretty Raw Preview Visualize JSON


```
[
    {
        "data_source": "Wikipedia",
        "matches": {
            "confidence_score": 0.48,
            "text": "Microsoft"
        },
        "name": "Microsoft",
        "url": "https://en.wikipedia.org/wiki/Microsoft"
    },
    {
        "data_source": "Wikipedia",
        "matches": {
            "confidence_score": 0.52,
            "text": "Bill Gates"
        },
        "name": "Bill Gates",
        "url": "https://en.wikipedia.org/wiki/Bill_Gates"
    },
    {
        "data_source": "Wikipedia",
        "matches": {
            "confidence_score": 0.54,
```


Cookies Capture requests Bootcamp Runner Trash

The response is of the following format:

```
[
    {
      "name": "string",
      "url": "string",
      "data_source": "string",
      "matches": {
        "confidence_score": "string",
        "text": "string"
      }
    }
  ]
```

There can be multiple entities with their data source, url and their matches based on the confidence score and text.

Personally Identifiable Information (PII) detection

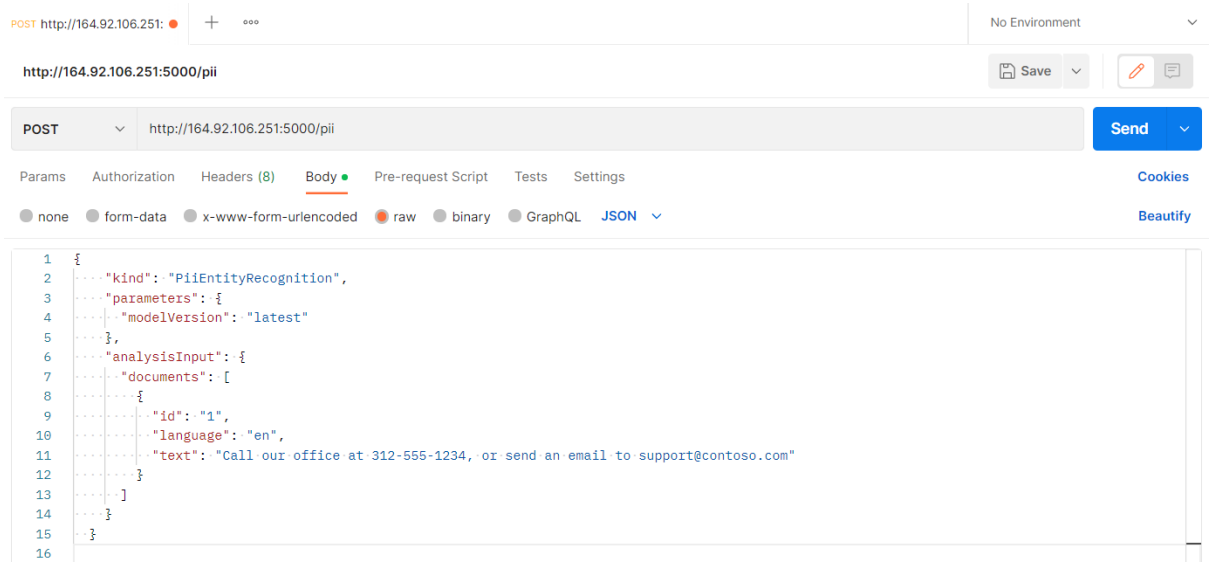
PII detection is one of the features offered by Azure Cognitive Service for Language, a collection of machine learning and AI algorithms in the cloud for developing intelligent applications that involve written language. The PII detection feature can identify, categorize, and redact sensitive information in unstructured text. For example: phone numbers, email addresses, and forms of identification.

To run the following API in Postman,

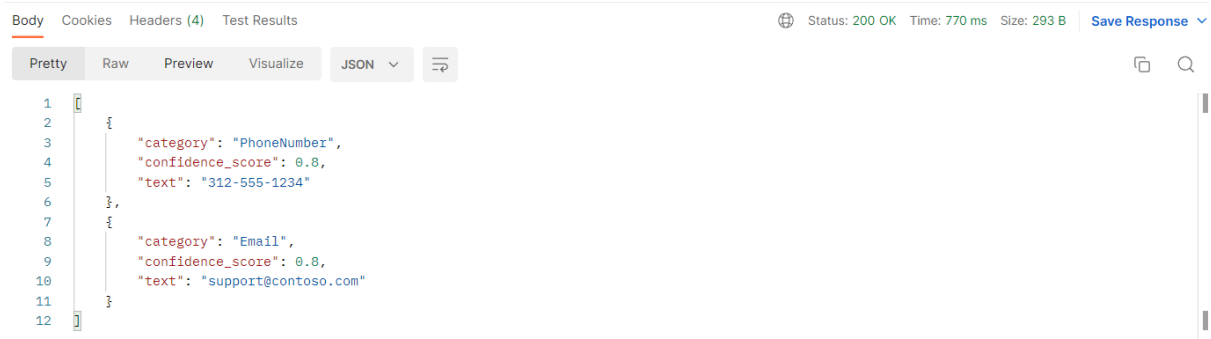
Copy **http://164.92.106.251:5000/pii** in the address bar and change the method to POST and add the following body as JSON:

```
{
    "kind": "PiiEntityRecognition",
    "parameters": {
      "modelVersion": "latest"
    },
    "analysisInput": {
      "documents": [
        {
          "id": "1",
          "language": "en",
          "text": "Call our office at 312-555-
1234, or send an email to support@contoso.com"
        }
      ]
    }
  }
```

Your Postman window should look like this:



Click Send to run it and you will receive the following response for the above body:



The response is of the following format:

```
[
    {
      "category": "string",
      "confidence_score": "string",
      "text": "string"
    }
]
```

where category is the category of the identitified entity.

Language Detection

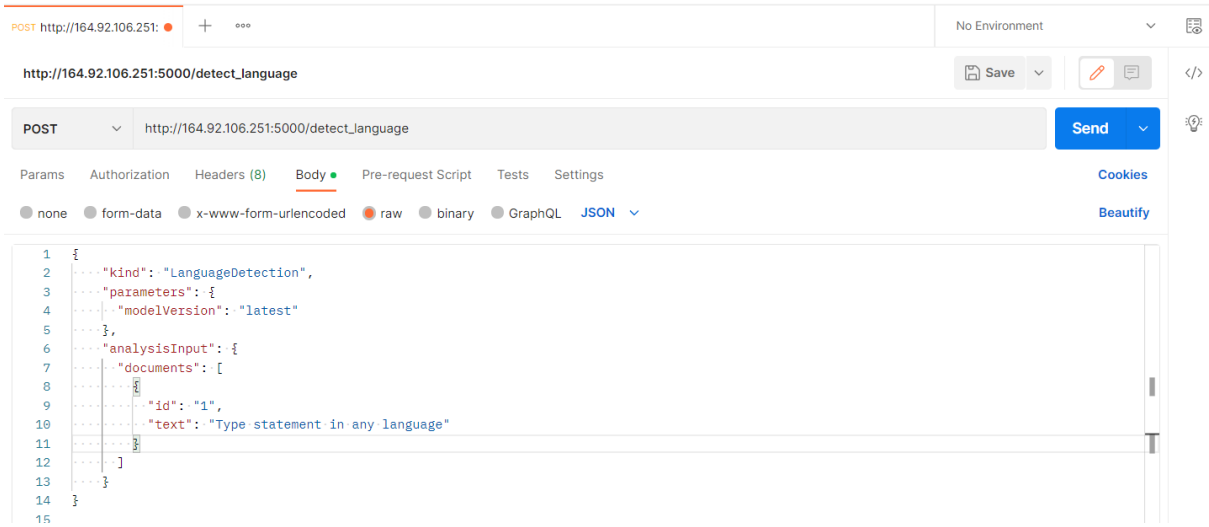
Language detection is one of the features offered by Azure Cognitive Service for Language, a collection of machine learning and AI algorithms in the cloud for developing intelligent applications that involve written language. Language detection can detect the language a document is written in, and returns a language code for a wide range of languages, variants, dialects, and some regional/cultural languages.

To run the following API in Postman,

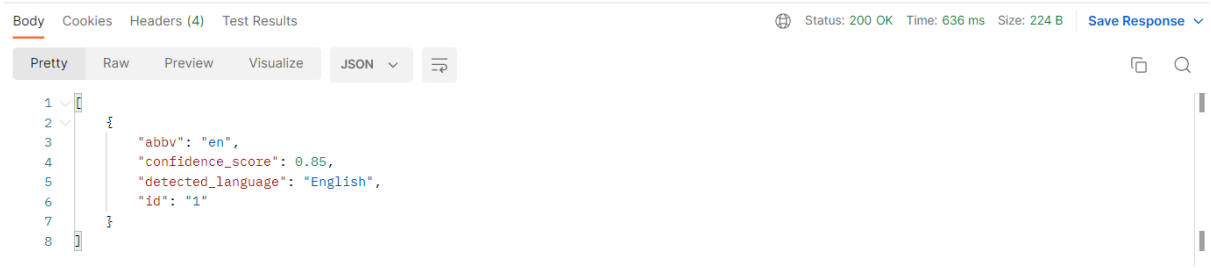
Copy **http://164.92.106.251:5000/detect_language** in the address bar and change the method to POST and add the following body as JSON:

```
{
    "kind": "LanguageDetection",
    "parameters": {
      "modelVersion": "latest"
    },
    "analysisInput": {
      "documents": [
        {
          "id": "1",
          "text": "Type statement in any language"
        }
      ]
    }
}
```

Your Postman window should look like this:



Click Send to run it and you will receive the following response for the above body:



The response is of the following format:

```
{
    "detected_language": "string",
    "id": "string",
    "confidence_score": "string",
    "abbv": "string"
}
```

where detected_language is the language detected by the service along with its abbv and confidence_score.

Question Answering

Question answering provides cloud-based Natural Language Processing (NLP) that allows you to create a natural conversational layer over your data. It is used to find the most appropriate answer for any input from your custom knowledge base (KB) of information.

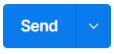
We can also use question answering without a knowledge base with the prebuilt question answering REST API, which is called via query-text. In this case, we provide question answering with both a question and the associated text records we would like to search for an answer at the time the request is sent.

To run the following API in Postman,

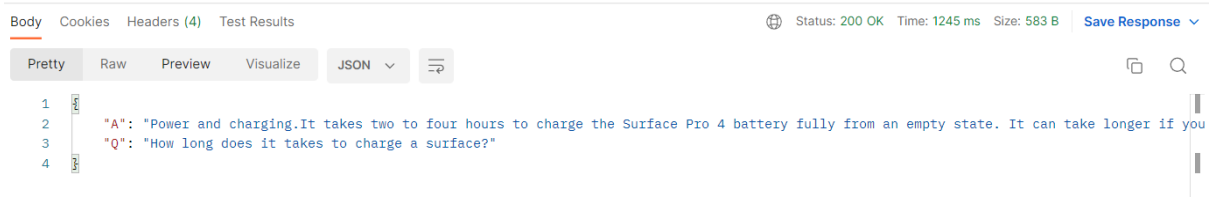
Copy http://164.92.106.251:5000/qa in the address bar and change the method to POST and add the following body as JSON:

```
 {
     "question": "How long does it takes to charge a surface?",
     "language": "en",
     "stringIndexType": "Utf16CodeUnit",
     "records": [
       {
         "id": "doc1",
         "text": "Power and charging.It takes two to four hours to charge the Su
rface Pro 4 battery fully from an empty state. It can take longer if you're usin
g your Surface for power-
intensive activities like gaming or video streaming while you're charging it"
       }
     ]
   }
```

Your Postman window should look like this:

Click 

to run it and you will receive the following response for the above body:



The response is of the following format:

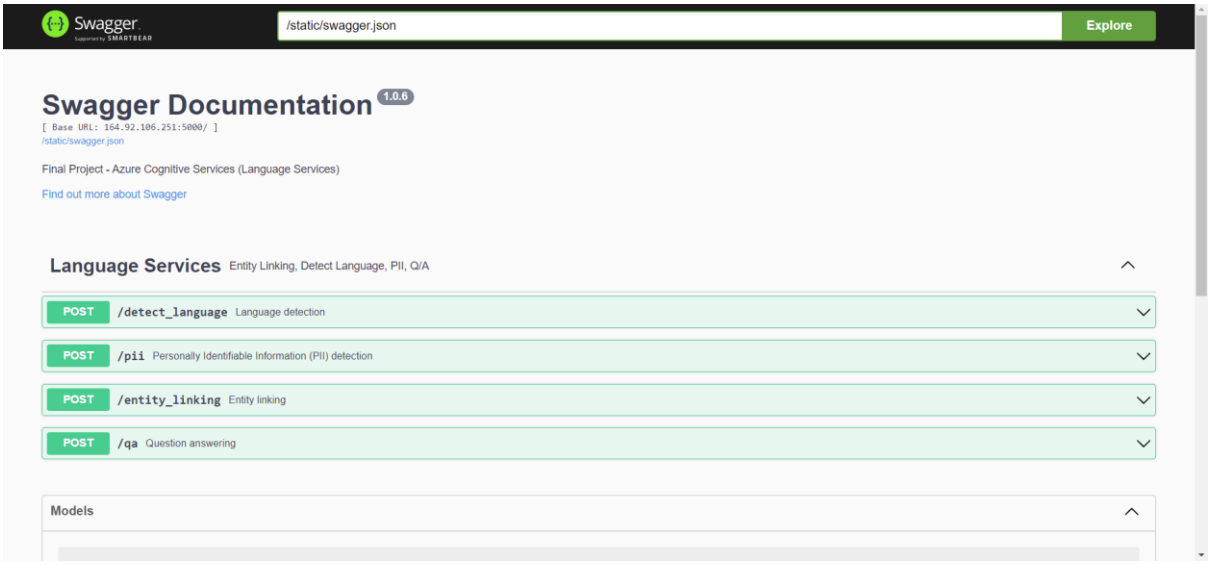
```
{
    "A": "string",
    "Q": "string"
}
```

where Q is the question asked and A is the answers given by the service based on the KB provided.

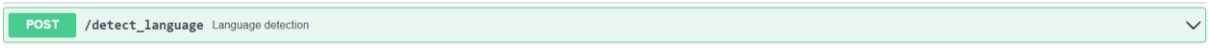
Steps to run APIs on Swagger

To run the APIs on Swagger follow these steps:

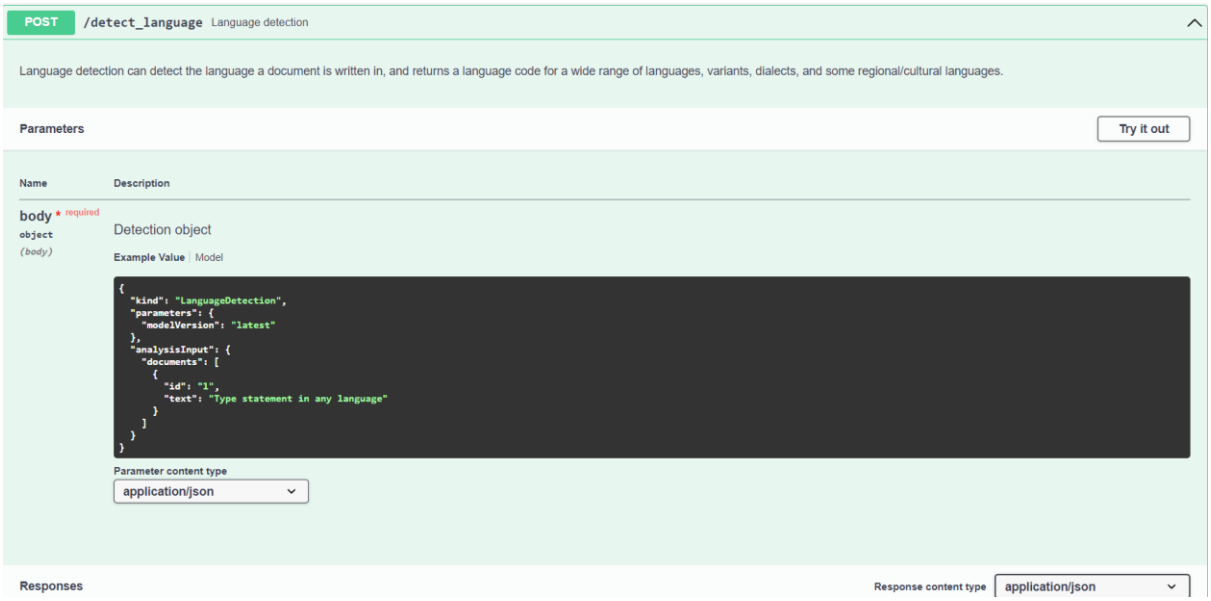
- Open this link on your browser: http://164.92.106.251:5000/api/docs/
  You will see the following page:

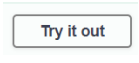


- Now, to run any of the following APIs, click on the API. For instance, click on

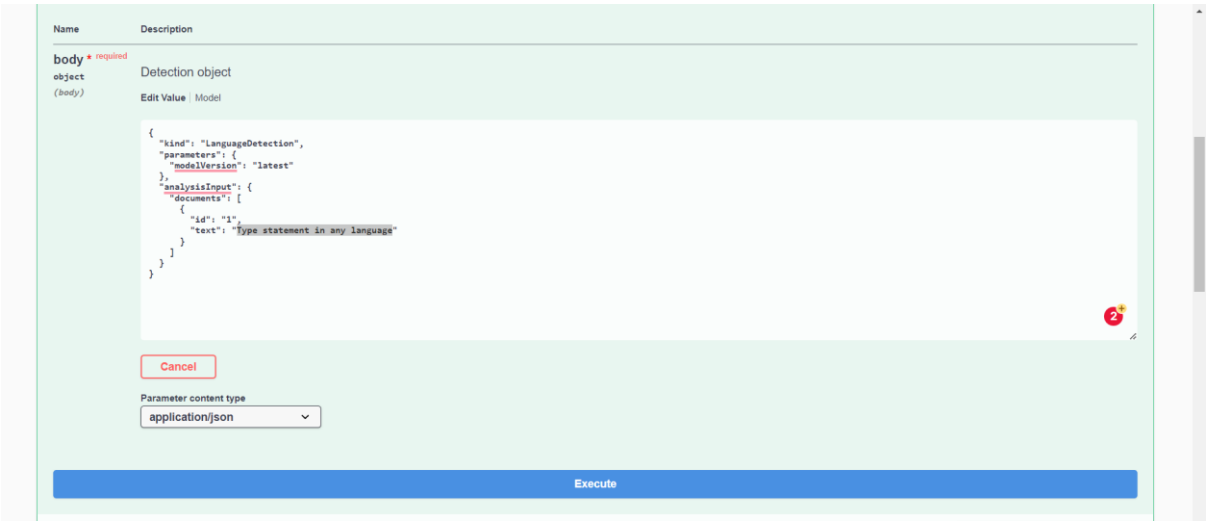


You will see the following content:

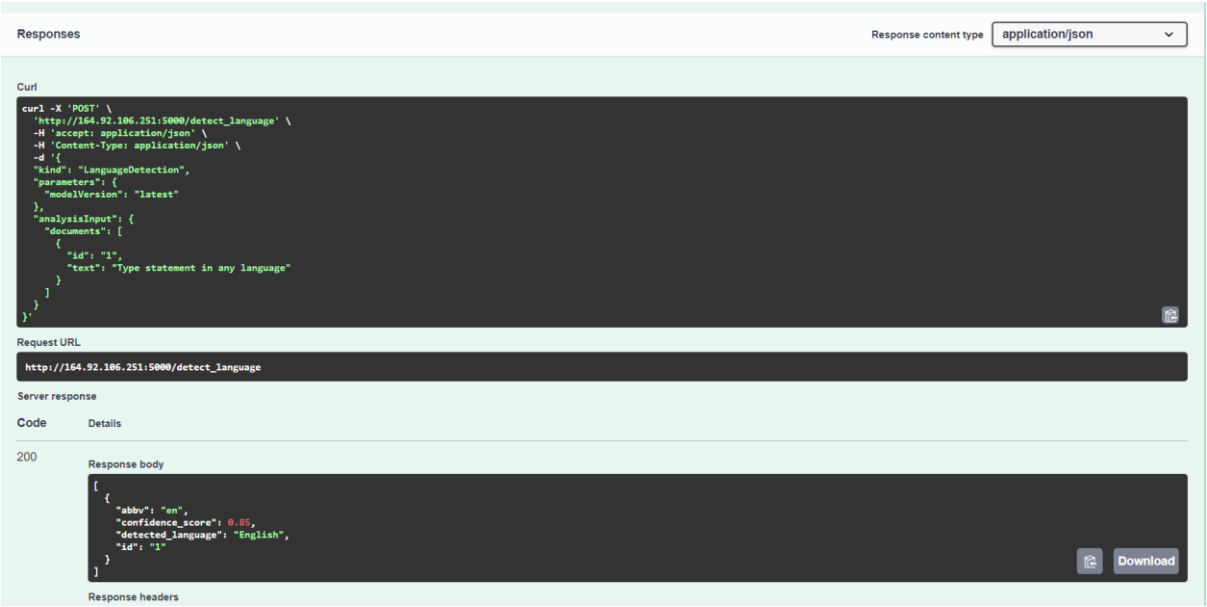


- To test the API, click on 

which will result in the following:



- There is a sample input in the body description. You can change text of it and then click on **Execute** to see the output.
- At last, you will see the following output based on the sample input given:



You can run all the other APIs in the same way. Every API has its sample body and you can change the text of them but remember don't remove any keys which are present. If you do you will get an error.