API Specification Document

The following are the 3 specific public API’s I have selected.

Cat Breed Classification API:  
 Link: <https://zylalabs.com/api-marketplace/machine+learning/cat+breed+classification+api/499>

Description: this API will accept an image URL of a cat. Using a machine learning algorithm will attempt to classify it as one or more breeds of cat.

Code Snippet:

import requests

url = <https://zylalabs.com/api/499/cat+breed+classification+api/373/pet+classification?url=https://i.guim.co.uk/img/media/c9b0aad22638133aa06cd68347bed2390b555e63/0_477_2945_1767/master/2945.jpg?width=1200&height=1200&quality=85&auto=format&fit=crop&s=97bf92d90f51da7067d00f8156512925>

headers = {'Authorization': 'Bearer YOUR\_API\_KEY'}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)

response snippet:

{

"results": [

{

"score": 0.9666043519973755,

"label": "Siamese cat, Siamese"

},

{

"score": 0.021663600578904152,

"label": "Egyptian cat"

},

{

"score": 0.0021568245720118284,

"label": "tabby, tabby cat"

},

{

"score": 0.001865892787463963,

"label": "lynx, catamount"

},

{

"score": 0.0007982752867974341,

"label": "Persian cat"

}

]

}

This API will be integrated into a method to return the cat name with the highest identification score

Cats Facts API:

Link: https://zylalabs.com/api-marketplace/data/cat+information+api/3561

Description: this API will provide cat facts per breed. A snippet of the response will list every kind of fact about a cat breed it will provide.

Code Snippet:

import requests

url = <https://zylalabs.com/api/3561/cat+information+api/3922/get+data?name=abyssinian>

headers = {'Authorization': 'Bearer YOUR\_API\_KEY'}

response = requests.request("GET", url, headers=headers, data=payload)

print(response.text)

Response Snippet:

[

{

"length": "12 to 16 inches",

"origin": "Southeast Asia",

"image\_link": "https://api-ninjas.com/images/cats/abyssinian.jpg",

"family\_friendly": 3,

"shedding": 3,

"general\_health": 2,

"playfulness": 5,

"children\_friendly": 5,

"grooming": 3,

"intelligence": 5,

"other\_pets\_friendly": 5,

"min\_weight": 6,

"max\_weight": 10,

"min\_life\_expectancy": 9,

"max\_life\_expectancy": 15,

"name": "Abyssinian"

}

]

This API will be integrated into a method to retrieve facts about a cat based on its Breed name.

YouTube API:

Link: https://developers.google.com/youtube/v3/docs/search/list

Description: The YouTube API is being used to provide a short list of 3-5 YouTube videos about the specified cat breed. It will return a list of video objects about a subject you send, the subject will be used to search YouTube.

The following code snippet calls the API

Code Snippet:

# API client library

import googleapiclient.discovery

# API information

api\_service\_name = "youtube"

api\_version = "v3"

DEVELOPER\_KEY = 'YOUR\_API\_KEY'

# API client

youtube = googleapiclient.discovery.build(

api\_service\_name, api\_version, developerKey = DEVELOPER\_KEY)

# Request body

request = youtube.search().list(

part="id,snippet",

type='video',

q="Siamese cats",

videoDuration='short',

videoDefinition='high',

maxResults=1

)

# Request execution

response = request.execute()

print(response)

The above code snippet returns the top results for “short” videos with “high” definition about “siamese cats”

And just like the previous two APIs, it returns a Json object structured like so  
{

"kind": "youtube#searchResult",

"etag": etag,

"id": {

"kind": string,

"videoId": string,

"channelId": string,

"playlistId": string

},

"snippet": {

"publishedAt": datetime,

"channelId": string,

"title": string,

"description": string,

"thumbnails": {

(key): {

"url": string,

"width": unsigned integer,

"height": unsigned integer

}

},

"channelTitle": string,

"liveBroadcastContent": string

}

}

This API will be integrated into a method to retrieve the top YouTube result for a specific breed of cat.

NOTE: The two Zyla based API’s (Cat Breed Classification, and Cat Information (Facts)) are subscription based, and project use of them will expire on 5/3/2024