INFO 762 Final Project:

Custom API Specification Document

**Introduction:**

The custom RESTFUL API created manipulates an SQL database and is able to send and retrieve data from the database, and update values in the database.

It is based off FLASK framework. It is written in Python programming language. It is Cat themed.

The main programming components of the API are detailed below. Additionally python programs that interact with the API and the driver file are also detailed.

**Data Definition:**

The data definition created in the custom API is the “CatModel.” It contains 5 values. They are: id (primary key), name (name of cat), image\_url (image url of cat), youtube (youtube link to video about the cat), and facts (string with several facts about the breed).

A screen shot of a computer

Description automatically generated

1 Code Snippet of Data Definition

**Put and Update request parsing, Resource fields:**

When new data is packaged in a put request and sent the server, the data is mapped to each argument in the parser. The same is done for request data.

A screen shot of a computer code

Description automatically generated

2 Mapping parsed data for put and update requests

Each column in the database is mapped to a resource field of a specific type.

A screen shot of a computer

Description automatically generated

3 resource field mapping

**Resource Functions:**

Each Cat entry in the database can be considered a resource, so a resource named “Cat” is created to get, put, patch, and delete entries from the database. A code snippet of each is provided below.

A screen shot of a computer program

Description automatically generated

4 defining the Cat Resource Class and the get function

A computer code on a black background

Description automatically generated

5 defining the put function.

A screen shot of a computer program

Description automatically generated

6 defining the patch or "update" function

A computer screen shot of a computer code

Description automatically generated

7 Defining the Delete function, note: This was unused and is unfinished

A resource called “cat” is registered to the local machine hosted API server. This creates the local page where you can see the JSON version of the database information in a browser.

A screen shot of a computer

Description automatically generated

8 registering the cat resource.

**Public APIs integrated:**  
Three public API’s where integrated into the python project and produce information that the custom API can post to the server.

**Cat Classifier:**

Cat Classifier (INFOproj\_CATCLASSIFER\_API.py) is a python file that contains the method that interacts with the Cat Classifier API. The Cat Classifier API analyzes an image URL of a cat and returns a list of possible breed classifications. The breed with the highest score is returned.

A computer screen shot of text

Description automatically generated

9 The first part of the method, that sends the image URL to the API and returns a response

When a response is returned, a simple sorting algorithm is employed to get the name associated with the max score:  
A computer screen with text and numbers

Description automatically generated

10 getting the cat name based off ID score

The name is returned in string format.

**Cat Fact Retriever:**

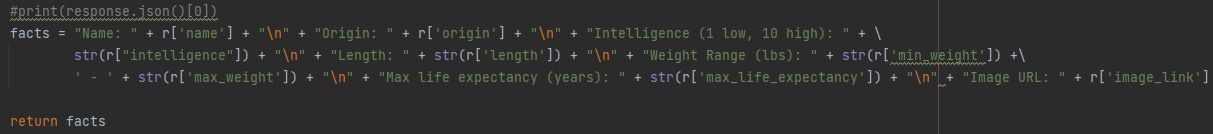
The Cat Fact retriever (INFOproj\_CATFACTS\_API.py) is the python file that contains the method that calls the Cat Fact API. The method creates a request to the Cat Fact API and queries for some basic facts about a specific breed name. The facts retrieved in the project are: Origin, Intelligence level, Length, weight range, max life expectancy, and another image URL if available.

A screenshot of a computer

Description automatically generated

11 making a fact request based on cat\_name (breed name) to the api

When a Json response is returned, the data is formatted into a multi-line string that can be easily read.



The fact string is returned.

**YouTube API:**  
The YouTube API (INFOproj\_YOUTUBE\_API.py) is integrated into a python file with a method to query YouTube for a top video search result based off a key word. The method using the YouTube API is used to retrieve the video URL of the top result, which is then stored in the database for its specific entry.

A screen shot of a computer program

Description automatically generated

12 creating a request to the youtube API and getting a repsonse.

When a response is returned, the video ID is extracted and added to the YouTube URL string. The total YouTube URL is returned.

A screen shot of a computer program

Description automatically generated

13 returning the YouTube URL.

**Custom API specific python files:**

**Put:**

Put.py is a python file that can post new information to the SQL database through the custom API. Its has two methods. The first is “put()” which is a method used to put some basic data in the database when it is first created.

A computer screen with text

Description automatically generated

14 adding some image URLs to the database.

The second method is “put\_new()” which can add a new entry with a new ID and image URL to the database.

A screen shot of a computer

Description automatically generated

15 put for a new database entry

I only put the image URL in the parameters since other functions can be used to fill in the database data based on the ID of the cat in the image.

**Get:**  
get.py is a python file with a simple method to get the Json response of a specific entry in the database.

A screenshot of a computer

Description automatically generated

16 the get method for a entry in the database

**Update:**

Update.py is a simple python file that contains a method used to update entries in the database. Specifically, it uses the request patch function to manipulate values in preexisting entries in the database.

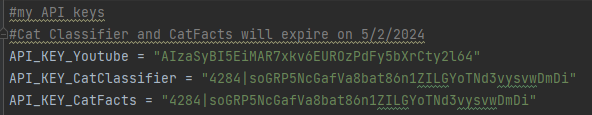
A computer screen shot of a black screen

Description automatically generated

17 the update function

**Global file:**

The Global file (INFOproj\_GLOBALS.py) is a simple file of variable constants. In the end, I only put the API keys here, even though I planned to find more uses for it.



18 API keys in the global file

**d**river.py is a simple and straightforward python script that uses all 4 APIs to add information the database. Each chunk is detailed below.

A screen shot of a computer

Description automatically generated

19 relevant imports to the driver file

First, we get a cat name from the database that already exists, and we get a YouTube video about that breed of cat. Then we post the YouTube video to that entry. Since we have the breed’s name already, we can use the cat fact API to get some facts about it, and post those too. Results retrieved are displayed in the console.

A computer screen shot of a computer program

Description automatically generated

20 First chunk of Driver file

Next, we get an image URL from the database, and run it through the cat classifier to get a breed name. then get a YouTube video and facts about the breed, and post all of it to the database.

A computer screen shot of a computer program

Description automatically generated

21 ID'ing an image of a cat, then filling in the rest of its information

Then, to showcase all functions at once, lets post a new entry with a new image URL and then collect all the information about that breed and post it to the database.

A computer screen shot of a computer code

Description automatically generated

22 Final block of driver file

**Additional:**

I have a plugin that allows me to view my database in real time as I make changes to it through my code.

A screenshot of a computer

Description automatically generated

23 The database

Also, here is a browser screenshot to see the information from the database in a webpage in Json format.

A screenshot of a computer

Description automatically generated

24 Json in the webpage

Links to all public API’s used can be found in the API specification document.