# RESTful API Project

# Project Narrative

# Hector Avalos

# INFO 762 Interoperability

# Professor: Dr. Grover Walters

# Date: 4/27/2022

## Abstract

This project aims to This project aims to develop a unified platform that leverages three distinct Application Programming Interfaces (APIs): a fiscal data API, an API for gaming giveaways, and an API for price comparisons of PC games (What is an API, 2024). The goal is to create a comprehensive service that provides users with insights into fiscal data, access to gaming opportunities, and information on game prices.

The first API, the fiscal data API, will be used to fetch and analyze financial data related to a currencies country, the name of the currencies, and the exchange rates. This data will help users make informed decisions in the financial domain. The second API focuses on gaming giveaways, allowing users to access information about ongoing giveaways, promotions, and free game offers across various gaming platforms. This feature enhances user engagement and provides opportunities for gamers to explore new titles without financial barriers. The third API facilitates price comparisons for PC games, enabling users to compare prices.

By aggregating and presenting this data in a user-friendly format, the platform assists gamers in finding the best deals and making cost-effective purchase decisions. The project's implementation will involve integrating these APIs seamlessly, ensuring data consistency, and reliability. Overall, this multi-API integration project aims to deliver a valuable platform catering to gamers looking for gaming opportunities and cost-effective game purchases.

## Introduction

Representational State Transfer (REST) is an application that complies with a set of rules (Gupta, 2023). The REST API must have a uniform interface, client-server based, stateless, cacheable, have a layered system and allow for code on demand. This paper seeks to describe an intersection of resources through a Driver Application that converts some data from three public APIs, then transforms the data and sends it to a Custom API. To begin, I reviewed a list of public APIs provided by the professor. In the selection process, I was looking for three APIs that are cohesive in nature. This proved to be a difficult and integral part of the process because I found that many APIs sharing commonality required an API key or an API authentication. My first attempt involved the integration of a weather API, an event booking API, and flight comparison API. However, it was difficult to call information using an API key, and I was unsuccessful in effectively manipulating the data received. I decided to filter my API selection to APIs with no API key or API authentication, eventually settling on an API that provides fiscal data, one that provides gaming giveaways, and an API for price comparisons of PC games.

## Motivations and Background

APIs offer diverse functionalities and services across various domains. Recognizing the potential value of integrating multiple APIs to create a platform with these capabilities, this project seeks to combine data from disparate sources. Through the process of integrating a fiscal data API, a gaming giveaways API, and a PC game price comparison API, the platform can offer users a holistic view of both financial and gaming realms.

The primary motivation is to empower users with actionable insights and information. Gamers benefit from accessing information about game prices and can compare prices to make optimal gaming-related choices. Additionally, users can view fiscal data from different countries, currencies, and exchange rates. In summary, the motivations and background for this project are rooted in the desire to create a platform delivering actionable insights, to potentially uncover market gaps by leveraging the integration of fiscal data, gaming opportunities, and PC game price comparisons through APIs.

## Proposed Approach

The proposed approach for developing the integrated platform involved multiple steps: API selection, creating an API specification document for each of the public APIs and one for the Custom API, develop a Custom API and a Driver Application that converts some of the data from the public functions, save the data file in .CSV format.

Using Anaconda Navigator and Jupyter Notebook as an experimental environment I used pieces of code to call data from the public APIs.

A screen shot of a computer program

Description automatically generated

However, I replaced the base URL and endpoints to work for the specific public APIs I chose to work with. Additionally for each API I used the following code to filter columns to display only the ones I wanted to use.

A screen shot of a computer code

Description automatically generated

After I began using Visual Studio Code to create the database that would hold the data.

A screen shot of a computer

Description automatically generated

The code displayed, db.create\_all(), was used to create the database that holds the data from the Driver Application. The class DataModel creates a table within the database to hold data from columns derived from each of the three APIs. The values are as follows: country, currency, and exchange\_rate derived from the fiscal API, title, and worth derived from the gaming giveaway API, and lastly, gameID, and salePrice derived from the price comparisons for PC games. The Driver application merges a pull of the three APIs and pushes the data into the custom API. Afterward as discussed above the custom API pushes the data into the database.

## Evaluation

The success and effectiveness of the integrated platform combining a fiscal data API, a gaming giveaways API, and a PC game price comparison API can be evaluated based on multiple key criteria. I was able to create the server, driver, and call information from three public APIs effectively creating the desired result of exercising skills in innovation and technology of an API. Where I fell short is in the ability to GET information into the Driver Application to Post to the Custom API.

A computer screen with a cloud

Description automatically generated

## Experimental results

As mentioned above the design of the Custom API was successful. After running the full CustomAPI.py code in the dedicated terminal the result displays the server URL.

A screen shot of a computer program

Description automatically generated

To confirm the table is created in the database I downloaded Sqlite Browser from the URL [DB Browser for SQLite (sqlitebrowser.org)](https://sqlitebrowser.org/) (2023). This can be seen in the image below,

A close-up of a computer screen

Description automatically generated

Sqlite Browser also allows me to confirm if the Driver Application was successful in Getting information from the public APIs and Posting them to the Custom API to then populate the data in the database for the client to call in Get functions. Yet, after multiple attempts to configure the Driver app I was unsuccessful in Getting the data from the public APIs even after the aforementioned data manipulation.

A screenshot of a computer

Description automatically generated

The image above shows the table created in the database with no values.

## Threats to Validity

While striving for a comprehensive and effective integrated platform, it's essential to acknowledge and address potential threats to the validity of the project's outcomes. These threats can impact the accuracy, reliability, and generalizability of findings and conclusions. The following threats to validity should be considered and mitigated:

* Data Accuracy and Quality
* API Reliability and Availability
* Biases in Data Analysis and Interpretation

By proactively identifying and mitigating these threats to validity, the project can enhance the credibility, reliability, and impact of its outcomes, ensuring that the integrated platform delivers valuable and trustworthy results to users and stakeholders.

## Conclusion and Future Work

The development of the integrated platform combining a fiscal data API, a gaming giveaways API, and a PC game price comparison API represents a significant achievement in my understanding of interoperability concepts and stands as a potential to deliver valuable insights and opportunities to users. Key accomplishments of this project were,

* Calling data from three public APIs
* Developing API specifications documents for the three public APIs, and a Custom API.
* Creation of a Custom API
* Creation of a database to hold data transformed from the Custom API

Future work will involve versioning to ensure the Driver API effectively Gets useful data to post to the custom API. By pursuing this avenue for future work, the integrated platform can continue to evolve, including my understanding of interoperability concepts as they have been taught in this course.

## References

DB browser for sqlite. (2023). Retrieved from https://sqlitebrowser.org/

Gupta, L. (2023). What is REST? Retrieved from https://restfulapi.net/

Public-apis/public-apis: A collective list of Free Apis. (n.d.). Retrieved from https://github.com/public-apis/public-apis?tab=readme-ov-file

What is an API (Application Programming Interface)? (2024). Retrieved from https://aws.amazon.com/what-is/api/