Emily Zhou

Professor Williamson

DS 2002

16 Oct 2024

DS 2002 ETL Project Reflection

The ETL project I developed is a filtering tool for Pokémon, specifically designed to filter by generation and type. Additionally, I incorporated a utility to calculate the total of a Pokémon's base stats, which is particularly useful given the data-intensive nature of the franchise. Throughout the course of this project, I encountered several challenges, ranging from determining the input types to producing the final output of the ETL processor. Before discussing the challenges that I encountered with the API, I want to emphasize that the CSV was exceptionally well-formatted and straightforward, making the processes of filtering and calculating the total base stats remarkably simple since it followed a similar flow as previous projects that we had done in class.

One significant hurdle arose when I found a reliable CSV for standard Pokémon data, as the next step involved figuring out how to handle sample JSON input. The most widely used Pokémon API, PokeAPI, offers a wealth of information. However, the structure of its data posed challenges for filtering by generation. Instead of directly filtering, I had to make API calls for every Pokémon ID within the generation range, which is incorporated into the code. As a result, a minimum of 100 API calls were required each time a user selected JSON as the input option, considerably slowing down the process. Despite this inefficiency, the information was comprehensive.

Another challenge I faced was the sheer volume of data packed into each JSON file. I had to sift through around 6,000 lines of code just to determine if the API included information on total base stats, which it did not. Consequently, I added this column myself. While the versatility of the API is impressive due to the wealth of information it contains, the method of making specific Pokémon calls could have been more efficient. Unfortunately, the only option was to hit the endpoint over 100 times for a single execution of the code.

Despite these challenges, the overall utility of the project is significant. Many Pokémon games rely heavily on the data associated with each Pokémon, which opens the door for various enhancements to the processor, such as filtering or sorting columns. For instance, users could easily view Pokémon with the highest attack. More importantly, the ETL processor's ability to parse both JSON and CSV files, modify them, and load the information into a database is its greatest strength. Handling different file formats and processing data is essential for any future data project. The concepts and practices I gained from this ETL project will provide a solid foundation for future endeavors.