Aryan Karthik

Objective: Write a python program to optimally parse and read a JSON file. Further, the program will write to fields that can be inserted into a Key-Value data structure (python dictionary). The dictionary is equivalent to a Key-Value database like Firebase.

This document is intended to explain the algorithmic approach, pseudocode for the program (implemented in python), and the overall structure.

Pseudocode (main.py):

1. Read JSON file (file.json)
2. Define a dictionary with a schema
3. Use temporary lists to store values of each type in a loop
4. Assign temp lists to respective values in the previously defined dictionary

Description:

My python program is a simple way to assign JSON values into a Key-Value object with python. After reading the JSON into a variable with json.loads(), I defined a dictionary that uses the schema of the JSON as Keys with empty values. I then used temp lists to insert each element within a loop that iterates through the “records” values with the len() function. I also used an incrementor variable (i) to insert values with the insert() function. I then assigned each list as a value to its respective key in the dictionary.

To decrease runtime I decided not to use nested loops and opted for a more simplistic approach. I tested the idea of one temp list retrieving values from the file and dumping them into the dictionary, but that caused unnecessary complications and use of excess memory.

After the program finishes running, a dictionary will store every value of the JSON file as lists keyed to the id of the values. Values can be called in the order they appear in the JSON file.

This could be further extended to write the same key and value data structure format into a database such as Firebase.