Name : Raman Kumar
Reg. No.: 12106871

Coding Interview for Graduates

Coding Challenge

Using https://api.publicapis.org/ complete the following tasks in any language/server.

1) Create an API that lists the title, description based on the category passed as an input parameter.
2) Create an API that would save a new entry with all the relevant properties which retrieves values from the endpoint GET /entries.
3) Question: what are the key things you would consider when creating/consuming an API to ensure that it is secure and reliable?

Theoretical Challenge

Suppose you have a CSV file with the data below.

A1: 5, A2: 7, A3: 9, B1: 3, B2: 8, B3: =4+5, C1: =5+A1, C2: =A2+B2, C3: =C2+B3

This can be represented in an excel sheet below:

	A	B	C
1	5	3	=5+A1
2	7	8	=A2+B2
3	9	=4+5	=C2+B3

I want a program that will take the CSV input above and produce CSV output with the results. If it is a value, then return a value. If it is a formula then calculate the formula and return the value of that formula.

1. How will you tackle the challenge above?
2. What type of errors you would you check for?
3. How might a user break your code?

Question 1 Solution

1.
To create an API that lists the title and description based on the category passed as an input parameter, you should use the following steps:
Use the requests library in Python to send a GET request to the endpoint https://api.publicapis.org/entries
Use the json() method to parse the response and extract the relevant data Use the filter() function to filter the entries based on the category passed as an input parameter and return the title and description of the filtered entries it can be in the form of csv/json formate

2.
To create an API that saves a new entry with all the relevant properties which retrieves values from the endpoint GET /entries,
you should use the following steps:
Use the requests library in Python to send a GET request to the endpoint https://api.publicapis.org/entries
Use the json() method to parse the response and extract the relevant data Use the user input to construct new data Use the requests library to send a POST request to the endpoint https://api.publicapis.org/entries with the new data as the payload Return a confirmation message indicating that the entry has been saved.

3.
When creating/consuming an API to ensure that it is secure and reliable, the following are some key things to consider:
Authentication and Authorization: Ensure that only authorized users can access the API and that the API can verify the identity of the user

Input validation: Ensure that all input is valid and in the expected format Rate

limiting: Limit the number of requests that can be made by a user or IP address to prevent abuse
Error handling: Ensure that the API can handle errors gracefully and return meaningful error messages
Caching: Use caching to improve performance and reduce the number of requests to the server
Monitoring: Monitor the API's performance and logs to detect any issues and troubleshoot problems

Code:

Question: what are the key things you would consider when creating/consuming an API to ensure that it is secure and reliable
Ans:

```
fetch(`https://api.publicapis.org/entries?category=${category_name}`)
  .then(response => response.json())
  .then(data => {
    // data.entries is an array of API entries
    // you can filter the entries by category_name and extract the title and
description properties
    const filteredEntries = data.entries.filter(entry => entry.category ===
category_name);
    const titleDescEntries = filteredEntries.map(entry => ({title: entry.API, description:
entry.Description}));
  });
```

Question 2 Solution

1)

To tackle the challenge of reading a CSV file with data that includes both values and formulas, I would use a combination of the following steps: Read the CSV file using a library such as pandas or csv Iterate through each cell in the file, and for each cell that is a formula, use a library such as python expr or eval() to evaluate the formula and replace the formula with its calculated value Write the modified data to a new CSV file

2)

Some types of errors that I would check for include: Incorrectly formatted formulas (e.g. missing operators or parentheses) Reference to non-existent cells in formulas Incorrectly formatted CSV file (e.g. missing or extra commas) Issues with reading or writing the CSV file (e.g. file not found or permission denied

3)

A user could break the code by: Providing a CSV file with a format that is not expected or supported Providing formulas that reference cells that are not present in the file Providing formulas that contain malicious code (e.g. code that deletes files or sends data to external servers) Providing a CSV file with a very large number of formulas or cells, causing the program to run out of memory or take a long time to execute