# **Coding Interview for Graduates**

Name: Suraj Kumar

Email: surajkumar150399@gmail.com

Phone: 7764946860

Github: <a href="https://github.com/suraj-raj-3000">https://github.com/suraj-raj-3000</a>

### **Coding Challenge:**

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?

#### **Solution:**

### **Prerequisites:**

- > Python
- > Flask
- ➤ Thunder Client (To pass the arguments I am using thunder client.)

#### **Software used:**

> visual Studio code editor

### **Key things to ensure that it is secure and reliable:**

Organizations have a lot to lose with unprotected APIs, so make security a priority and build it into your APIs as they're being developed. API security shouldn't be an afterthought or considered "someone else's responsibility."

#### **CSV** file is:

	Α	В	С	D
1	Α	В	С	
2	5	3	5	
3	7	8	6	
4	9	9	10	
5	4	5	9	
6	5	8	13	
7	2	6	8	
8				

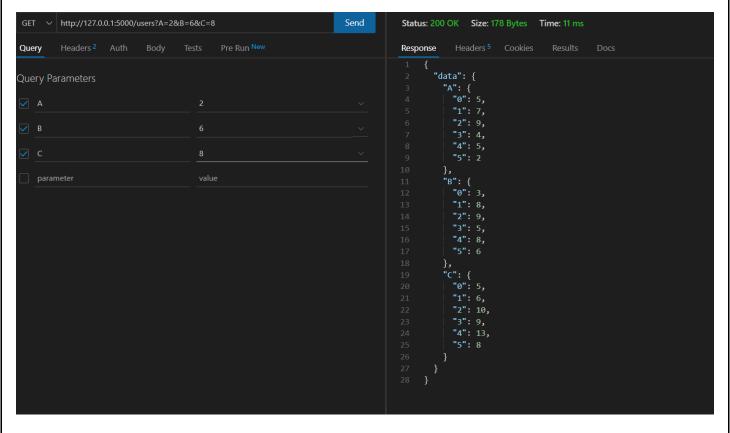
#### **Program:**

```
from flask import Flask
from flask_restful import Resource, Api, reqparse
import pandas as pd
import ast
app = Flask( name )
api = Api(app)
class Users(Resource):
    def get(self):
        data = pd.read csv('users.csv') # read CSV
        data = data.to dict()
        return {'data': data}, 200
    def post(self):
        parser = reqparse.RequestParser() # initialize
        parser.add_argument('A', required=True)
        parser.add_argument('B', required=True)
        parser.add_argument('C', required=True)
        args = parser.parse_args() # parse arguments to dictionary
        # create new dataframe containing new values
        new data = pd.DataFrame({
            'A': args['A'],
            'B': args['B'],
            'C': args['C'],
        })
        # read our CSV
        data = pd.read_csv('users.csv')
        if args['userId'] in list(data['userId']):
            return {
                'message': f"'{args['userId']}' already exists."
            }, 401
        else:
            # create new dataframe containing new values
            new data = pd.DataFrame({
                'userId': args['userId'],
                'name': args['name'],
                'city': args['city'],
                'locations': [[]]
            })
            # add the newly provided values
            data = data.append(new_data, ignore_index=True)
            data.to_csv('users.csv', index=False) # save back to CSV
            return {'data': data.to_dict()}, 200 # return data with 200 OK
api.add_resource(Users, '/users')
if __name__ == '__main__':
  app.run()
```

## **Output:**

```
Status: 200 OK
               Size: 105 Bytes Time: 11 ms
            Headers 5
                       Cookies
Response
                                            Docs
     {
       "data": {
         "A": {
           "0": 5,
           "1": 7,
           "2": 9
         "B": {
           "0": 3,
           "1": 8,
           "2": 9
         "C": {
           "0": 5,
           "1": 6,
           "2": 10
         }
```

### **After Add parameters**



- 1. How will you tackle the challenge above?
  - ➤ Using Python and Flask, I first construct an API endpoint, following which I use Pandas to read a CSV file, initialise a reqparse, add a new data Frame, and append more CSV files. He adds the data frame into the csv file whenever I make a Post request with a parameter.
- 2. What type of errors you would you check for?
  - ➤ I first verified that if a data frame departs, an error occurs, and then I verified that if the API is not operating, he returns a 401 Forbidden error.
- 3. How might a user break your code?
  - ➤ Someone might easily damage my code by providing a bad parameter because my code's validation is incomplete.