How to Create a Web API using Flask Framework in Python

This tutorial will guide you through create Web API using Flask Framework.

For creating a webapi in python using flask you need flask-restful package.

pip install flask-restful

Above command will install flask-restful package

Flask Restful is an extension for Flask that adds support for building REST APIs in Python using Flask as the back-end. It encourages best practices and is very easy to set up. Flask restful is very easy to pick up if you're already familiar with flask.

In flask_restful, the main building block is a resource. Each resource can have several methods associated with it such as GET, POST, PUT, DELETE, etc. for example, there could be a resource that calculates the square of a number whenever a get request is sent to it. Each resource is a class that inherits from the Resource class of flask_restful. Once the resource is created and defined, we can add our custom resource to the api and specify a URL path for that corresponding resource.

```
# using flask_restful
from flask import Flask, jsonify, request
from flask_restful import Resource, Api

# creating the flask app
app = Flask(__name__)
# creating an API object
api = Api(app)

# making a class for a particular resource
# the get, post methods correspond to get and post requests
# they are automatically mapped by flask_restful.
# other methods include put, delete, etc.
class Hello(Resource):

    # corresponds to the GET request.
    # this function is called whenever there
    # is a GET request for this resource
    def get(self):

        return jsonify({'message': 'hello world'})

    # Corresponds to POST request
    def post(self):

        data = request.get_json()     # status code
        return jsonify({'data': data}), 201
```

```python
# another resource to calculate the square of a number
class Square(Resource):

    def get(self, num):

        return jsonify({'square': num**2})


# adding the defined resources along with their corresponding urls
api.add_resource(Hello, '/')
api.add_resource(Square, '/square/<int:num>')


# driver function
if __name__ == '__main__':

    app.run(debug = True)
```