# JSON API Comparator Project Setup and Running Guide

1. Prerequisites

Before you start, make sure you have the following installed on your machine:

- Python 3: The project uses Python 3. You can verify if Python is installed by running:

  python3 --version

- pip: Ensure pip (Python's package installer) is available to install dependencies. You can check by running:

  pip --version

2. Clone the Project Repository

If the project is stored in a Git repository (e.g., on GitHub), clone it to your local machine using:

  git clone https://github.com/your-repo/json_comparator_tool.git

Otherwise, download the project folder as a ZIP file and extract it.

3. Set Up the Virtual Environment

In the root folder of the project (where requirements.txt is located), create a Python virtual environment to isolate dependencies.

1. Create a virtual environment:

  python3 -m venv .venv

2. Activate the virtual environment:

  On macOS/Linux:

source .venv/bin/activate

On Windows:

.\.venv\Scripts ctivate

You should see (.venv) in your terminal prompt, indicating the virtual environment is active.

4. Install Project Dependencies

With the virtual environment active, install all the dependencies listed in requirements.txt:

pip install -r requirements.txt

This will install Flask, requests, and other necessary packages.

5. Update the Flask Application (if necessary)

Make sure that the Flask app (app.py) is set up to run on the correct port (e.g., port 5001, or modify if needed).

The fetch request in the frontend (i.e., comparator.html) must also point to the correct port. Update this part of the code if your Flask app is running on a port other than 5001:

const res = await fetch('http://localhost:5001/proxy?url=' + encodeURIComponent(url)); // Update port if necessary

6. Run the Flask Application

Start the Flask app by running the following command:

python3 app.py

This will start the Flask development server. You should see output like:

```
 * Running on http://127.0.0.1:5001/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
```

7. Open the Application in a Browser

Now that Flask is running, open your browser and go to:

```
http://127.0.0.1:5001/
```

This will load your comparator.html page served by Flask on port 5001.

8. Using the JSON API Comparator

- Enter API URLs: In the input fields for "API A URL" and "API B URL", enter the URLs of the two APIs you want to compare.
- Click Compare: Click the Compare button to compare the two APIs. The results will be shown in a table, highlighting the differences between the two APIs.
- Search & Filter: Use the search bar to filter and highlight specific JSON paths.

9. Stopping the Flask Server

When you're done, you can stop the Flask development server by pressing CTRL + C in the terminal.

Additional Configuration (Optional)

Changing Ports

If port 5001 is already in use, you can change the port Flask uses in the app.py file:

```
app.run(debug=True, host='0.0.0.0', port=5002)  # Change the port number
```

## Deactivating the Virtual Environment

When you're done with the project, deactivate the virtual environment by running:

```
deactivate
```

## Git Ignore

Make sure you are not including unnecessary files, like the virtual environment (.venv), in version control. If using Git, add .venv to your .gitignore file:

```
.venv/
```

## Troubleshooting

"Failed to Fetch" Error: This error often happens when the fetch request cannot reach the backend. Ensure the Flask app is running and accessible at the correct URL. Verify CORS headers are properly set up.

Port Already in Use: If port 5001 is occupied, you can run Flask on a different port by changing the port parameter in app.run().

## Conclusion

By following these steps, you should have a working local setup for the JSON API Comparator project. This guide ensures that you can run the project locally in the future by setting up the environment, installing dependencies, and running Flask.