

ReadMe

Setup and Dependencies

Dependencies

This code uses the following Python libraries:

- `FastAPI`: A modern, fast (high-performance), web framework for building APIs.
- `pydantic`: A data validation and parsing library.
- `dotenv`: A library for reading environment variables from a `.env` file.
- `os`: A Python standard library module for interacting with the operating system.
- `typing`: A module for supporting type hints.

Environment Setup

1. Make sure you have Python installed on your system.
2. Install the required Python packages using pip:

```
pip install fastapi pydantic python-dotenv
```

3. Create a `.env` file in the same directory as your Python script. This file should contain the `INTERVALS` environment variable with a comma-separated list of intervals in the format `start-end`. For example:

```
INTERVALS=3-4.1,8.5-8.7,4-4.5,0-1.1,31.5-41.27
```

Logic and Usage

This code defines a FastAPI application with two endpoints:

1. `/insertSamples/` (POST)

- **Input:** This endpoint expects a JSON request with a list of floating-point numbers to insert into the samples.

Example:

```
{
  "data": [8.1, 8.2, 30, 4.2, 31.51, 1, 41.27]
}
```

- **Output:** It responds with a JSON message indicating that the samples were inserted successfully.

2. `/metrics/` (GET)

- **Input:** No request data is required. It's a simple GET request.
- **Output:** It calculates and returns metrics, including interval counts, sample mean, sample variance, and outliers.

Endpoint Details

- The `insert_samples` endpoint inserts the provided data into a thread-safe `samples` list using a lock to ensure that multiple requests don't interfere with each other when modifying the data.
- The `get_metrics` endpoint calculates interval counts, sample mean, sample variance, and identifies outliers. It returns these metrics in JSON format.
- The code reads intervals from the `.env` file and stores them in the `intervals` list.
- The `is_within_intervals` function checks if a value is within any of the defined intervals.
- Sample mean and variance are calculated using standard mathematical formulas.
- This code is designed to handle concurrent requests safely using a lock (`samples_lock`) to protect access to the `samples` data structure.

To run the FastAPI application, execute this script “`uvicorn app:app --host 0.0.0.0 --port 8000 --reload`”, and your API will be available at `http://localhost:8000` by default. You can make requests to the `/insertSamples/` and `/metrics/` endpoints using your favorite API client (e.g., `curl`, Postman, or a web browser).

You can customize the host and port by passing additional arguments to the `uvicorn` command if needed.