Workshop-RestApplication-Python

Your Flask application sets up two routes at `/hello` to handle both `GET` and `POST` requests. Here's a breakdown of the implementation:

1. **`GET` Request on `/hello`**:
    - The route `@app.route('/hello', methods=['GET'])` responds to `GET` requests.
    - It simply returns the string `"Hello, World!"`.
2. **`POST` Request on `/hello`**:
    - The route `@app.route('/hello', methods=['POST'])` responds to `POST` requests.
    - It retrieves JSON data from the request body using `request.get_json()`.
    - It fetches the value associated with the key `name` using `data.get('name', 'World')`, defaulting to `"World"` if `name` is not provided.
    - A JSON response is returned with a `message` key, containing `"Hello, <name>"`.

How to Test It

Using `curl` (Command Line):

- **GET Request**:

```
curl http://127.0.0.1:5000/hello
```

Output:

```
Hello, World!
```

```
curl http://127.0.0.1:5000/friends
```

Output:

```
[
  "Koen"
]
```

- **POST Request**:

```
```

```
curl -X POST http://127.0.0.1:5000/hello -H "Content-Type:
application/json" -d '{"name": "Jeffeke"}'
```

Output:

```
{
  "message": "Hello, Jeffeke"
}
```

```
  curl -X POST http://127.0.0.1:5000/friend -H "Content-Type:
application/json" -d '{"name": "Koen"}'
```

Output:

```
{
  "message": "Welcome, Koen"
}
```

Using Postman:

- **GET Request**: Select `GET`, enter `http://127.0.0.1:5000/hello`, and send the request.
- **POST Request**: Select `POST`, enter `http://127.0.0.1:5000/hello`, set `Content-Type` to `application/json`, and provide JSON in the body, e.g., `{"name": "Jeffeke"}`.

Notes:

- This app runs locally at `http://127.0.0.1:5000` by default.
- Make sure Flask is installed (`pip install flask`) and run the script using `python app.py`.
- For production, consider using a WSGI server like Gunicorn or uWSGI.