**Fast API**

**Introduction:**

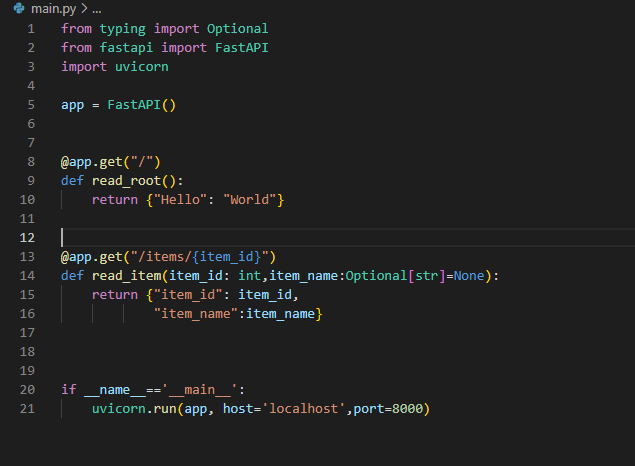
FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints.

The key features are:

* **Fast**: Very high performance, on par with **NodeJS** and **Go** (thanks to Starlette and Pydantic). [One of the fastest Python frameworks available](https://fastapi.tiangolo.com/#performance).
* **Fast to code**: Increase the speed to develop features by about 200% to 300%. \*
* **Fewer bugs**: Reduce about 40% of human (developer) induced errors. \*
* **Intuitive**: Great editor support. Completion everywhere. Less time debugging.
* **Easy**: Designed to be easy to use and learn. Less time reading docs.
* **Short**: Minimize code duplication. Multiple features from each parameter declaration. Fewer bugs.
* **Robust**: Get production-ready code. With automatic interactive documentation.
* **Standards-based**: Based on (and fully compatible with) the open standards for APIs: [OpenAPI](https://github.com/OAI/OpenAPI-Specification" \t "_blank) (previously known as Swagger) and [JSON Schema](https://json-schema.org/).

**Get Request:**

* Create a file main.py with:



**Put and Post Request:**.

Here I will use Pydantic **BaseModel** class that will act as a request body.

When we need to send some data from client to API, we send it as a request body. In other words, a request body is data sent by client to server. On the other hand, response body is the data the API sends back to the client. APIs always provide some response. However, clients do not need to send request body in every case

