🚀 FastAPI Tutorial: Build Your First API

Step 1: Install FastAPI and Uvicorn

FastAPI is a lightweight and fast Python framework for building APIs. We also need **uvicorn**, a server to run our API.

🔹 **Run this command in your terminal:**

```
pip install fastapi uvicorn pydantic
```

Step 2: Create a Simple API

Now, let's create a **basic FastAPI app**.

🔹 **Create a new Python file**:

📂 **project_folder**

├── `main.py`

🔹 **Write the following code in `main.py`:**

```
from fastapi import FastAPI

# Create an instance of FastAPI
app = FastAPI()

# Define a simple GET endpoint
@app.get("/")
def home():
    return {"message": "Welcome to FastAPI!"}
```

🔹 **What this code does:**

- It imports FastAPI and creates an app instance.
- Defines an **endpoint (`/`)** that returns a JSON response.

Step 3: Run the API

We use **Uvicorn** to start our FastAPI server.

🔹 **Run this command in the terminal:**

```
uvicorn main:app --reload
```

- `main` → Name of the Python file (`main.py`).
- `app` → The FastAPI instance we created.
- `--reload` → Enables auto-reload when code changes.

🔹 **Expected output in the terminal:**

```
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
```

Step 4: Test the API

✅ Open your browser and go to:
🔗 **http://127.0.0.1:8000/**

You should see:

```
{"message": "Welcome to FastAPI!"}
```

Step 5: Explore Swagger UI

FastAPI automatically generates interactive API documentation.

✅ Open **Swagger UI**:
🔗 **http://127.0.0.1:8000/docs**

You'll see an interactive UI where you can test your API.

✅ Or visit **Redoc** documentation:
🔗 **http://127.0.0.1:8000/redoc**

Step 6: Add Query Parameters

Let's modify our API to accept a **name parameter**.

🔹 **Update `main.py`:**

```
@app.get("/greet/")
def greet(name: str):
    return {"message": f"Hello, {name}!"}
```

🔹 **How to test:**
Visit: **http://127.0.0.1:8000/greet/?name=Alice**
Response:

```
{"message": "Hello, Alice!"}
```

Step 7: Add a Request Body

Now, let's add a **POST endpoint** that receives a **JSON request body**.

🔹 **Update `main.py`:**

```
from pydantic import BaseModel

# Define a model for request body validation
class User(BaseModel):
    name: str
    age: int

@app.post("/users/")
def create_user(user: User):
    return {"message": f"User {user.name} of age {user.age}
created!"}
```

🔹 **How to test:**

1. Go to **http://127.0.0.1:8000/docs**.
2. Click **POST /users/**.
3. Click **"Try it out"** → Enter this JSON:

```
{
  "name": "John",
  "age": 30
}
```

4. Click **"Execute"**.
5. Response:

```
{"message": "User John of age 30 created!"}
```

Step 8: Deploy the API (Local Deployment)

To deploy on a real server, you can run:

```
uvicorn main:app --host 0.0.0.0 --port 8000
```

For cloud deployment, you can use **Docker, AWS, or Render**.

Conclusion

🎯 You've built a **simple FastAPI** with: ✅ A **GET request** (`/greet/`) with a query parameter.
✅ A **POST request** (`/users/`) with a request body.
✅ Auto-generated **Swagger UI** documentation.

In the project you will learn how to create an API to apply machine learning model on user data.