FastAPI Reading: Four Short Pages

For Lab 5

FastAPI is the Python library you'll build Lab 5's application programming interface (API) with. Its official tutorial is short and clear, and these four pages cover every FastAPI pattern Lab 5 uses.

Optional, for reference. The two lecture videos cover what you need for class. Use these pages when you want more depth on a pattern, or another example of it, while you build Lab 5. About 30 minutes for all four, if you type each example into a `main.py` in your codespace and run it as you read.

Two things to translate

- **Starting the server.** Wherever the tutorial says `fastapi dev main.py`, type `uvicorn main:app --reload` instead. The `fastapi` command isn't installed in the course codespace, and `uvicorn` is the program it runs for you anyway.
- **Request bodies.** The tutorial makes them from Pydantic's `BaseModel`. Lab 5 uses SQLModel, and a SQLModel class *without* `table=True` works exactly the same way.

The four pages

Page	Read	Lab 5 uses it for
First Steps	From the top through *Interactive API docs*, then *Recap, step by step*, Steps 1–5	a route, and `/docs`
Path Parameters	From the top through *Data validation*	`/ideas/{idea_id}`, and why `/ideas/abc` is a 422
Request Body	From the top through *Request body + path parameters*	the body of `POST /ideas`, and of a vote
Handling Errors	*Use HTTPException* only	the 404s

Check yourself

You should be able to answer these out loud, without the pages open.

1. Your route is `@app.get("/ideas/{idea_id}")` with `idea_id: int`. Someone requests `/ideas/abc`. What status code comes back, and does your function run?
2. A `POST` arrives with a JSON (JavaScript Object Notation) body that's missing a required field. Who rejects it, and with what status code?
3. What's the difference, to the client, between your code raising `HTTPException(404)` and your code crashing?