

Chapitre 5 — Création d'une API avec FastAPI

1. C'est quoi ?

FastAPI est un **framework web Python** permettant de créer très rapidement une **API REST** moderne, rapide, documentée automatiquement.

2. Pourquoi on l'utilise ?

- Pour **exposer** les données SQLite à l'extérieur (UI web, autre app, client, etc.)
 - Pour **créer des routes HTTP** (`GET`, `POST`, etc.) faciles à tester (via `/docs`)
 - Pour **gérer la validation** des entrées automatiquement avec Pydantic
-

3. Vocabulaire

Terme	Rôle
<code>FastAPI()</code>	crée l'instance principale de l'API
<code>@app.get()</code>	décorateur pour route HTTP GET
<code>@app.post()</code>	décorateur pour route HTTP POST
<code>@app.put()</code>	décorateur pour modifier une ressource
<code>@app.delete()</code>	décorateur pour supprimer une ressource
<code>uvicorn</code>	serveur ASGI pour faire tourner l'API

4. Setup rapide

```
pip install fastapi uvicorn
```

5. Exemple minimal

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
def read_root():
    return {"msg": "Hello API"}
```

6. Lancer le serveur

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

- `main` = nom du fichier (main.py)
- `app` = nom de l'instance FastAPI
- `--reload` = redémarrage auto en dev

7. Interface interactive

Une fois lancé :

- Visite `http://127.0.0.1:8000/docs` pour tester l'API en live (Swagger UI)
- Visite `http://127.0.0.1:8000/redoc` pour la doc alternative

Résultat :

Tu as une API REST fonctionnelle, avec routes HTTP, doc automatique, testable en local, prête à servir tes données SQLite.