



# Scénario cours-api

---

Cette doc n'avais pas pour finalité de servir de cours, mais plutot de la documentation annexe pour les apprenants. Cette doc se base sur la doc officielle de FastAPI et SQLAlchemy, ainsi que sur des ressources internes pour compléter les notions.

## Sommaire

- **I. Introduction et concepts d'une API**
  - [01-definition-api](#)
  - [02-types-api](#)
  - [03-format-de-donnees](#)
  - [04-codes-retour-http](#)
  - [05-methodes-http](#)
- **III. Création d'une api REST avec FastAPI**
  - [07-organisation-projet](#)
    - uniquement jusque **Structure recommandée**
  - [01-introduction-fastapi](#)
  - [02-definition-endpoints](#)
  - [03-schema](#)
  - [04-swagger](#)
- **II. Modélisation et manipulation de la bdd avec SQLAlchemy**
  - 00-installation-et-introduction-cursor
    - [01-Installation-configuration-mariaDB-Dbeaver](#) (OPTIONNEL)
    - [02-Connexion-requetage-bdd](#) (OPTIONNEL)
  - [01-introduction-sqlmodel](#)
  - [02-creeer-des-tables](#)
  - [03-mise-en-place-bdd](#)
  - [04-requetage-simple](#)
- **III. Création d'une api REST avec FastAPI**
  - [05-dependances-injection](#) (+adaptation au sqlalchemy mis en place)
  - [06-gestion-erreurs](#)
  - [07-organisation-projet](#)
    - uniquement après **Structure recommandée**
- **II. Modélisation et manipulation de la bdd avec SQLAlchemy**
  - [05-requetage-one-to-many](#)
  - [06-requetage-many-to-many](#)

## Aller plus loin

Dans cet doc, il y a aussi les notions de tests automatisés, de déploiement, etc. qui ne seront pas abordé pendant ce cours. Vous pouvez les consulter si vous le souhaitez. Sachant qu'il vous sera demandé de faire des tests unitaires dans le projet final, il est conseillé de lire au moins la partie sur Pytest.

- **IV. Tests automatisés avec Pytest**
  - [01\\_pourquoi\\_tester](#)
  - [02\\_types\\_de\\_tests](#)
  - [03\\_introduction\\_pytest](#)
  - [04\\_httpx\\_et\\_client\\_test](#)
  - [05\\_fixtures](#)
  - [06\\_couverture\\_code\\_ci](#)
- **V. Environnement de développement et de déploiement avec Docker**
  - [01-pourquoi-docker](#)
  - [02-concepts-docker](#)
  - [03-creation-image-personnalisee-avec-Dockerfile](#)
  - [04-orchestration-avec-docker-compose](#)
  - [05-executer-et-tester-avec-dockers](#)
  - [06-bonnes-pratiques-docker](#)

Pour avoir un retour au plus tot, à la fin du projet dédié, on vous fournira le dossier : **VI. Exemple projet FastAPI SQLAlchemy** qui contiendra un exemple complet d'une API REST avec FastAPI et SQLAlchemy mais uniquement avec 1 seule entité (pour simplifier).