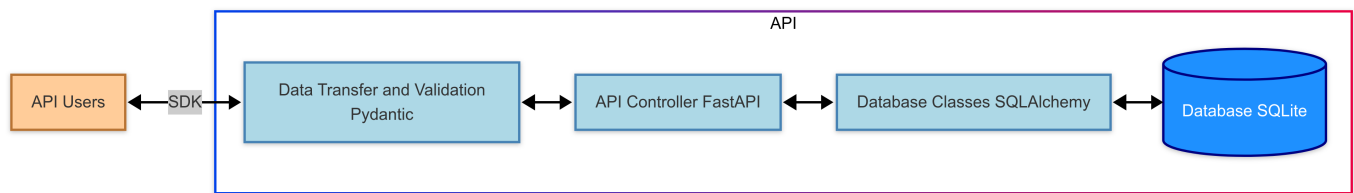


Définition d'une API et Explication de l'architecture de notre API de films



Explication du diagramme

Une API (Application Programming Interface) est une interface qui permet à des applications ou des utilisateurs d'interagir avec un système. Ce diagramme représente comment une API fonctionne pour gérer des données et interagir avec une base de données.

Étape par étape :

1. Les utilisateurs de l'API (**API Users**)

- Ce sont les personnes ou applications qui utilisent l'API pour envoyer ou récupérer des données.
- Pour interagir avec l'API, ils utilisent un **SDK** (Software Development Kit), qui est une bibliothèque (un package) Python facilitant l'envoi de requêtes.

2. Le transfert et la validation des données (**Pydantic**)

- Lorsque l'utilisateur envoie des requêtes à l'API, elles passent d'abord par **Pydantic**. Nous parlerons davantage de Pydantic dans une autre session.
- Pydantic vérifie que les données sont correctes (par exemple, s'il manque une valeur ou si un type est incorrect).

3. Le contrôleur API (**FastAPI**)

- FastAPI est le cœur de l'API. Il reçoit les requêtes des utilisateurs, traite les données et décide de ce qu'il faut faire (ex. : insérer de nouvelles données, récupérer des informations, etc.).
- Il agit comme un intermédiaire entre l'utilisateur et la base de données.

4. Les classes de base de données (**SQLAlchemy**)

- SQLAlchemy est une bibliothèque qui permet de communiquer avec la base de données de manière organisée.
- Il traduit les requêtes Python en instructions compréhensibles par la base de données.

5. La base de données (**SQLite**)

- SQLite est la database où se trouvent les données.
- L'API envoie des requêtes pour récupérer des données de la database SQLite.

En résumé :

- L'utilisateur envoie des données via l'**SDK**.

- Ces données sont **validées** (**Pydantic**).
 - L'API décide quoi faire (**FastAPI**).
 - Si nécessaire, elle stocke ou récupère des données via **SQLAlchemy**.
 - La base de données **SQLite** garde les informations de manière structurée.
-

L'API fonctionne comme un **restaurant moderne avec une tablette pour commander** :

1. **Le client (API Users)** arrive au restaurant et veut commander un plat.
2. **Le menu numérique (SDK en Python)** lui permet de passer commande facilement sans parler directement au serveur. Il peut sélectionner un plat en quelques clics.
3. **Le serveur (FastAPI)** reçoit la commande, la vérifie et la transmet en cuisine.
4. **Le chef (SQLAlchemy)** prépare le plat en récupérant les ingrédients depuis **la réserve (SQLite, la base de données)**.
5. Une fois le plat prêt, **le serveur revient avec la commande** et la sert au client.

Pourquoi le SDK est important ?

C'est comme une tablette qui facilite la commande : au lieu d'écrire une requête compliquée ou d'appeler directement le serveur, le client peut utiliser une interface simple et intuitive (le SDK) pour interagir avec l'API.