SQLModel & FastAPI

https://sqlmodel.tiangolo.com/ & https://fastapi.tiangolo.com/

https://sqlmodel.tiangolo.com/tutorial/fastapi/#learning-fastapi

Response Model

Senza, le API docs UI non conoscono lo schema (es. 1)
Ma quale classe è adatta ?

Modelli multipli e arricchiti

La classe PyDantic/SQLModel non è adeguata per tutti i verbi (es. 2)
Ad esempio: il client può mandare gli id (che al client API sembra opzionale!)

Serve un modello per la scrittura: senza id
Uno per la lettura con id non opzionale (es. 3)

Uno per update (es. 4)

Vista arricchita con le relazioni (vedere app “fastcash” completa)

FastAPI

Gestione limiti e paginazioni

Es. 4

Gestione session con dependency injection

Es. 5
Tavole successive per approfondimenti

FastAPI

Dependency Injections in FastAPI

Approccio potente e leggibile per riutilizzare logiche all'interno dei progetti

In generale la "dependency injection" è un sistema per istanziare oggetti e quelli da cui questi dipendono
Nel codice basterà dichiarare come un oggetto dovrebbe essere creato e il sistema risolve la catena di dipendenze a runtime

FastAPI permette di dichiarare gli oggetti e le variabili semplicemente avendoli come argomenti di funzioni nel path

E' un modo per wrappare una qualche logica che ritorna dei valori o oggetti, farci qualcosa e alla fine restituirli pronti per essere inoculati/"injected"

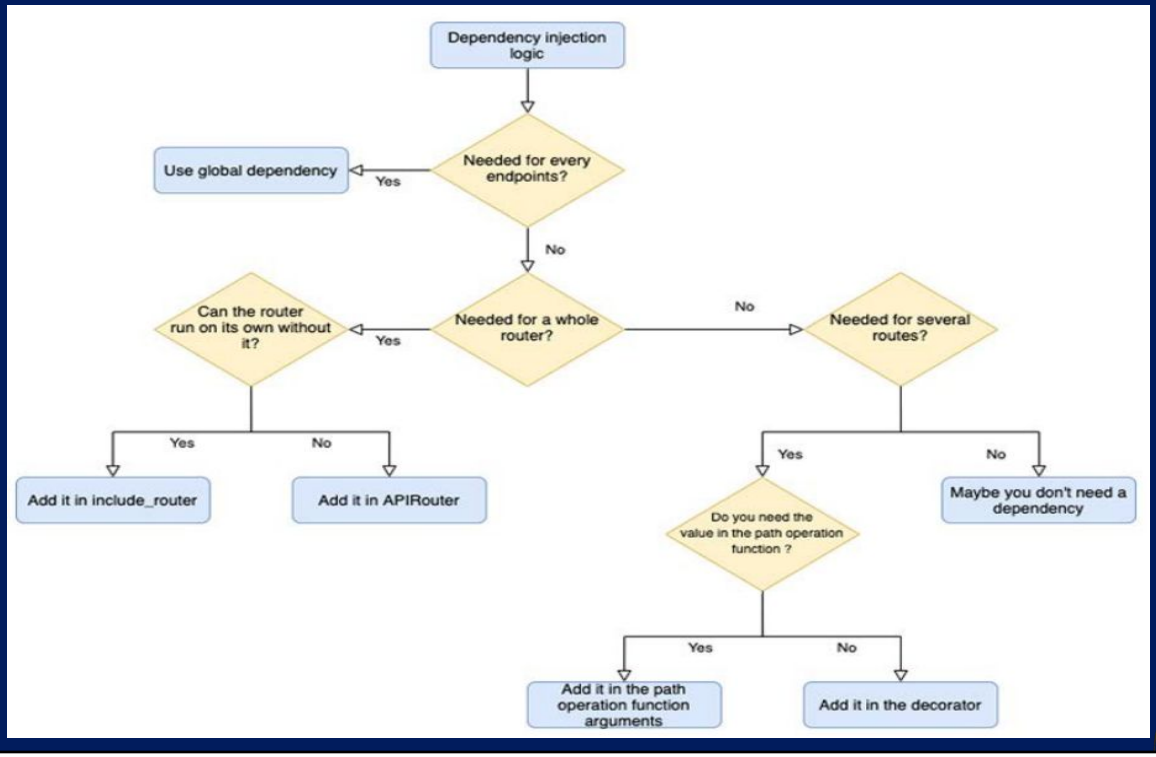
Si possono usare le dependencies a livello di path, router o globale (app)

Use cases: gestire la sessione, get_X_or_404, token/bearer auth (a livello globale)

FastAPI

Dependency Injections in FastAPI

A che livello?



FastAPI