



Requêtage simple avec SQLAlchemy

Sommaire

- **Requêtage simple avec SQLAlchemy**
 - **Sommaire**
 - **Sessions avec SQLAlchemy**
 - **Pourquoi utiliser `with` ?**
 - **Et après une modification ?**
 - **Introduction aux opérations CRUD**
 - **Exemple de modèle utilisé**
 - **Créer un enregistrement**
 - **Lire des enregistrements**
 - **Mettre à jour un enregistrement**
 - **Supprimer un enregistrement**

Sessions avec SQLAlchemy

Pour lire ou écrire dans la base, on utilise une **session** :

```
from sqlalchemy import Session

with Session(engine) as session:
    # écrire ou lire ici
    ...
```

Pourquoi utiliser **with** ?

- Gère automatiquement l'ouverture et la fermeture de la connexion
- Évite les erreurs de fuite de session
- Permet un code plus propre et robuste

Et après une modification ?

```
session.commit()
```

- Nécessaire pour **valider** les insertions, mises à jour ou suppressions
- Sans **commit()**, les données ne sont **pas enregistrées**

Introduction aux opérations CRUD

On parle d'opérations **CRUD** pour désigner :

Terme	Action
Create	Créer une nouvelle ligne
Read	Lire une ou plusieurs lignes
Update	Modifier une ligne existante
Delete	Supprimer une ligne

Exemple de modèle utilisé

```
class Hero(SQLModel, table=True):
    id: int | None = Field(default=None, primary_key=True)
    name: str
    secret_name: str
```

Créer un enregistrement

```
new_hero = Hero(name="Superman", secret_name="Clark Kent")

with Session(engine) as session:
    session.add(new_hero)
    session.commit()
```

Lire des enregistrements

```
from sqlmodel import select

with Session(engine) as session:
    statement = select(Hero).where(Hero.name == "Superman")
    result = session.exec(statement).first()
    print(result)
```

- `.first()` → renvoie le premier résultat ou `None`
- `.one()` → renvoie exactement un résultat ou lève une exception si aucun ou plusieurs résultats
- `.all()` → renvoie une **liste de tous les résultats**

Mettre à jour un enregistrement

```
with Session(engine) as session:
    hero = session.get(Hero, 1)
    hero.name = "Batman"
    session.add(hero)
    session.commit()
```

⚠ On utilise `session.add(...)` même pour une mise à jour

Supprimer un enregistrement

```
with Session(engine) as session:
    hero = session.get(Hero, 1)
    session.delete(hero)
    session.commit()
```