Passage à SQL Alchemy 2.0

Jacques Fize, Pierre Narcisi

Sommaire

1. Nouveautés de SQLAlchemy 2.0
2. Modification des requêtes
3. Définition des modèles en 1.3 VS 2.0
4. RoadMap

Nouveautés de SQLAlchemy 2.0

Changements majeurs dans SQLA

- Nouveau style de requêtes
  - Plus de similarités avec une requête SQL
  - Moins de comportements implicites
- Nouvelle définition des classes ORM
  - Typage des colonnes avec les types primaires de Python (`int`, `float`, `str`, ...)
- Amélioration des performances (*bulk insert*)

Ce qui change dans Geonature

Pour les utilisateurs

- Pas de changement

Pour les développeurs

- Modification des requêtes nécessaires
- Modification dans la définition des modèles avec SQLAlchemy
- Abandon des Query Class

Modifications des requêtes

Abandon de `Session.query()`

- La création de requête passe par `select()` ou `db.select()`
- Conséquence de : `Query` de SQLA devient legacy :
  Voir : https://flask-sqlalchemy.palletsprojects.com/en/3.1.x/queries/#select et
  https://docs.sqlalchemy.org/en/20/orm/queryguide/query.html#legacy-query-api

```
query = db.session.query(User).first()
# devient (limit(1) était implicite en 1.3)
query = db.session.scalars(db.select(User).limit(1)).first()
```

Utilisation de `scalar` pour retournée des objets ORM

- L'utilisation de `Session.execute(query).all()` retourne une liste de tuple typé par `Row` !
- Pour récupérer des objets ORM : utiliser `Session.scalars(query)`, `query.scalar()`, `Session.execute(query).scalar_one()`

```
result = db.session.execute(db.select(User)).all()
print([type(r) for r in result])
#[<class 'sqlalchemy.engine.row.Row'>,...]
result = db.session.scalars(db.select(User)).all()
print([type(r) for r in result])
#[<class 'pypnusershub.db.models.User'>,...]
```

Des requêtes plus explicites

Plus de similarités avec une requête SQL, moins de comportements implicites

```
query = db.session.query(User).count()
# Devient
session.scalar(
  select(func.count()).
  select_from(User)
)
```

Voir : https://docs.sqlalchemy.org/en/20/changelog/migration_20.html#migration-orm-usage

Evolution de flask-sqlalchemy

- Utilisation des `query class` devient *legacy*. SQLAlchemy abandonne la classe `Query`. (*Voir exemple ci-dessous*)

```
class StationQuery(GeoFeatureCollectionMixin, Query):
    def filter_by_params(self, params):
        ...
class Station(db.Model):
    query_class = StationQuery

Station.query.filter_by_params(...)
```

- Discussion avec les développeurs[1] de flask-sqlalchemy
  Voir https://github.com/pallets-eco/flask-sqlalchemy/issues/1287
- **Solution Envisagée** : Remplacer par des fonctions qui produisent les requêtes souhaités

```
def filter_by_params(**params):
    query = db.select(Station)
    for par,value in params.items():
      query =query.where(getattr(Station,par)==value)
    return query

result = db.session.scalars(filter_by_params).all()
```

Définition des modèles : 1.3 VS 2.0

Déclaration des modèles (Version SQLA 1.3)

```
[import]

class User(Base):
    __tablename__ = "user_account"
    id = Column(Integer, primary_key=True)
    name = Column(String(30)) # SQLA type
    fullname = Column(String)
    addresses = relationship(
        "Address", back_populates="user", cascade="all, delete-orphan"
    )
```

Déclaration des modèles (Version SQLA 2.0)

```
[import]

class User(Base):
    __tablename__ = "user_account"
    id: Mapped[int] = mapped_column(primary_key=True) # Python type instead of SQLA ones
    name: Mapped[str] = mapped_column(String(30)) # Except when specific
    fullname: Mapped[Optional[str]]
    addresses: Mapped[List["Address"]] = relationship(
        back_populates="user", cascade="all, delete-orphan"
    ) # back_populates is prefered to backref
```

RoadMap

RoadMap

1. **Maj de SQLAlchemy vers 1.4**
   Maj des requêtes :white*check_mark:*
   Maj des tests 🏃
   Maj des modules : _Import, *Export*, *Monitoring* et *Dashboard* ✅
2. **Mise à jour des dépendances**
   Abandon du support de Debian 10 ✅
   Passage à Flask 3.0.0 ✅
3. **Maj vers SQLAlchemy 2.0**
   Mise à jour des modèles (définition des tables)
   Dernier changements dans les tests

Questions ?