

[登录](#)

2019年04月30日 阅读 326

[关注](#)

Flask框架从入门到精通之模型创建与添加(十二)

知识点: 1、模型类创建 2、模型类关系 3、模型数据添加

一、概况

在Flask-SQLAlchemy中, 插入、修改、删除操作, 均由数据库会话管理。会话用`db.session`表示。在准备把数据写入数据库前, 要先将数据添加到会话中然后调用`commit()`方法提交会话。

数据库会话是为了保证数据的一致性, 避免因部分更新导致数据不一致。提交操作把会话对象全部写入数据库, 如果写入过程发生错误, 整个会话都会失效。

数据库会话也可以回滚, 通过`db.session.rollback()`方法, 实现会话提交数据前的状态。

二、创建模型类

我们用王者荣耀类型和英雄中的关系来创建模型类。比如射手类型对应了很多英雄:

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
import pymysql

pymysql.install_as_MySQLdb()
app = Flask(__name__)
# 设置连接数据库的URL
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:123456@127.0.0.1:3306/db_flask'

# 设置每次请求结束后会自动提交数据库中的改动
app.config['SQLALCHEMY_COMMIT_ON_TEARDOWN'] = True

# 数据库和模型类同步修改
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = True

# 查询时会显示原始SQL语句
app.config['SQLALCHEMY_ECHO'] = True

db = SQLAlchemy(app)

# 类型
class Type(db.Model):
    # 表名
    __tablename__ = 'tbl_types'
```

[复制代码](#)

```

# 数据库真正存在的字段
id = db.Column(db.Integer, primary_key=True) # 主键
name = db.Column(db.String(32), unique=True) # 名字

# 数据库中不存在的字段, 只是为了查找和反向查找。
# backref: 在关系的另一模型中添加反向引用
heros = db.relationship("Hero", backref='type')

# 英雄
class Hero(db.Model):
    # 表名
    __tablename__ = 'tbl_heros'
    # 数据库真正存在的字段
    id = db.Column(db.Integer, primary_key=True) # 主键
    name = db.Column(db.String(64), unique=True) # 名字
    gender = db.Column(db.String(64)) # 性别

    # 外键 一个射手对应很多英雄
    type_id = db.Column(db.Integer, db.ForeignKey("tbl_types.id"))

```

三、创建表和添加数据

上面我们已经提到在Flask-SQLAlchemy中, 插入、修改、删除操作, 均由数据库会话管理。

```

if __name__ == '__main__':
    # 0.0.0.0代表任何能代表这台机器的地址都可以访问
    # app.run(host='0.0.0.0', port=5000) # 运行程序

    db.drop_all() # 清楚数据库所有数据 只试用第一次运行
    db.create_all() # 创建表

    type1 = Type(name='射手')
    db.session.add(type1) # 添加到会话
    db.session.commit() # 提交

    type2 = Type(name='坦克')
    db.session.add(type2)
    db.session.commit()

    type3 = Type(name='法师')
    type4 = Type(name='刺客')
    db.session.add_all([type3, type4]) # 添加多个
    db.session.commit()

    hero1 = Hero(name='后羿', gender='男', type_id=type1.id)
    hero2 = Hero(name='程咬金', gender='男', type_id=type2.id)
    hero3 = Hero(name='王昭君', gender='女', type_id=type3.id)
    hero4 = Hero(name='安琪拉', gender='女', type_id=type3.id)
    hero5 = Hero(name='兰陵王', gender='男', type_id=type4.id)

    db.session.add_all([hero1, hero2, hero3, hero4, hero5]) # 添加多个
    db.session.commit()

```

[复制代码](#)

我们登入本地数据库查看一下：

```
mysql> show tables;
```

```

+-----+
| Tables_in_db_flask |
+-----+
| tbl_heros          |
| tbl_types          |
+-----+

```

<https://jikedachang.blog.csdn.net>

查询英雄类型：

```
mysql> select * from tbl_types;
```

```

+----+-----+
| id | name  |
+----+-----+
| 4  | 刺客  |
| 2  | 坦克  |
| 1  | 射手  |
| 3  | 法师  |
+----+-----+

```

<https://jikedachang.blog.csdn.net>

查询英雄：

```
mysql> select * from tbl_heros;
```

```

+----+-----+-----+-----+
| id | name      | gender | type_id |
+----+-----+-----+-----+
| 1  | 后羿      | 男     | 1       |
| 2  | 程咬金     | 男     | 2       |
| 3  | 王昭君     | 女     | 3       |
| 4  | 安琪拉     | 女     | 3       |
| 5  | 兰陵王     | 男     | 4       |
+----+-----+-----+-----+

```

<https://jikedachang.blog.csdn.net>

欢迎关注我的公众号：





文章分类



后端



文章标签



Flask



相关推荐

