1.迁移简介

Flask 中没有提供数据库的迁移操作。有时候开发过程发现数据库需要做修改，最简单直接的放式就是删除数据库重建，但是这会导致测试数据丢失。

在Flask中可以使用Flask-Migrate扩展，来实现数据迁移。并且集成到Flask-Script中，所有操作通过命令就能完成。

Flask-Migrate提供了一个MigrateCommand类，将数据库迁移命令添加到Flask-script 的manager对象中。实现命令行的方式完成迁移操作

最重要的就是回退版本问题（django也存在版本回退操作）

安装扩展模块
```
pip install flask-migrate===2.7.0 -i
https://pypi.tuna.tsinghua.edu.cn/simple
```

2.创建模型类

```
# -*- encoding: utf-8 -*-
from flask import Flask
from flask_script import Manager, Shell
from flask_sqlalchemy import SQLAlchemy
from flask_migrate import Migrate


app = Flask(__name__)  # type:Flask
manager = Manager(app)  # 创建 flask_script扩展管理 对象
```

```
# 连接数据库
app.config['SQLALCHEMY_DATABASE_URI'] =
'mysql+pymysql://root:qwe123@127.0.0.1:3306/py'
# 自动提交数据库中的改动
app.config['SQLALCHEMY_COMMIT_ON_TEARDOWN'] = True
# 追踪对象
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = True

# 创建数据库对象
db = SQLAlchemy(app)
# 第一个参数是Flask的实例，第二个参数是Sqlalchemy数据库实例
migrate = Migrate(app, db)

# manager 是Flask-Script的实例，这条语句在flask-Script中添加一个
db命令
manager.add_command('db', MigrateCommand)
# 最新的flask_migrate，已经去除了MigrateCommand，可以直接在命令行
中使用
# 如果需要使用MigrateCommand的话 降版本 flask_migrate 2.7.0


class Movie(db.Model):
    """创建一个电影模型类"""
    # 表名
    __tablename__ = 'movie'
    # id主键列，整数类型，自增
    id = db.Column(db.Integer, primary_key=True)
    # name，可变长字符串类型
    name = db.Column(db.String(20))
    # 关系字段，不是数据库中真实存在的字段，而是为了方便查询添加的
属性
    cast = db.relationship('Cast', backref='movie')

    def __repr__(self):
        return "Movie: %s " % self.name


class Cast(db.Model):
    """演员模型类"""
    __tablename__ = 'cast'
```

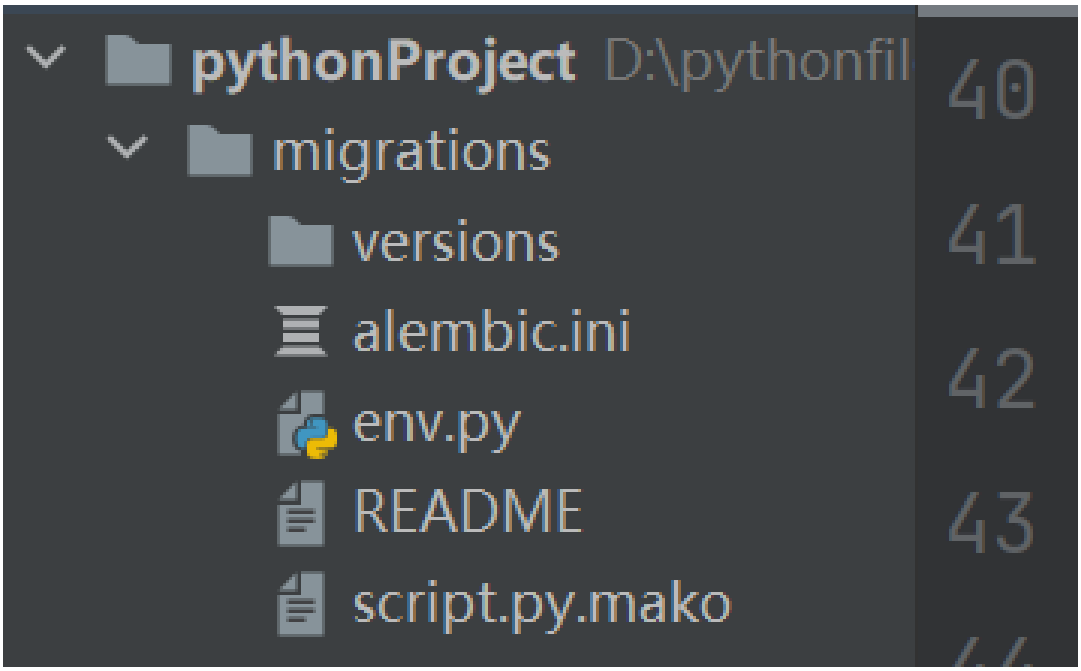
```
    # id主键列，整数类型，自增
    id = db.Column(db.Integer, primary_key=True)
    # name，可变长字符串类型
    name = db.Column(db.String(20))
    # 外键关联id
    movie_id = db.Column(db.Integer,
db.ForeignKey('movie.id'))

    def __repr__(self):
        return "Cast: %s " % self.name


if __name__ == '__main__':
    # app.run()
    manager.run()
```

3.创建迁移仓库

```
(flask_bei) D:\pythonfile\pythonProject>python kuoza.py db init
Creating directory D:\pythonfile\pythonProject\migrations ...  done
Creating directory D:\pythonfile\pythonProject\migrations\versions ...  done
Generating D:\pythonfile\pythonProject\migrations\alembic.ini ...  done
Generating D:\pythonfile\pythonProject\migrations\env.py ...  done
Generating D:\pythonfile\pythonProject\migrations\README ...  done
Generating D:\pythonfile\pythonProject\migrations\script.py.mako ...  done
Please edit configuration/connection/logging settings in 'D:\\pythonfile\\pythonProject\\migrations\\alembic.ini' before proceeding.
```

```
python kuoza.py db init
```

执行命令之后会在项目文件下生成一个 migrations 文件夹，生成的迁移文件都会保存在 versions目录下。



4.生成迁移文件

```
# 根据模型类生成迁移文件。 -m 参数是版本备注，，相当于git中的-m参数
python kuoza.py db migrate -m '第一次迁移'
```

```
(flask_bei) D:\pythonfile\pythonProject>python kuoza.py db migrate -m '第一次迁移'
INFO  [alembic.runtime.migration] Context impl MySQLImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.autogenerate.compare] Detected added table 'movie'
INFO  [alembic.autogenerate.compare] Detected added table 'cast'
Generating D:\pythonfile\pythonProject\migrations\versions\b54b5fb04980_第一次迁移.py ...  done
```

5.执行迁移

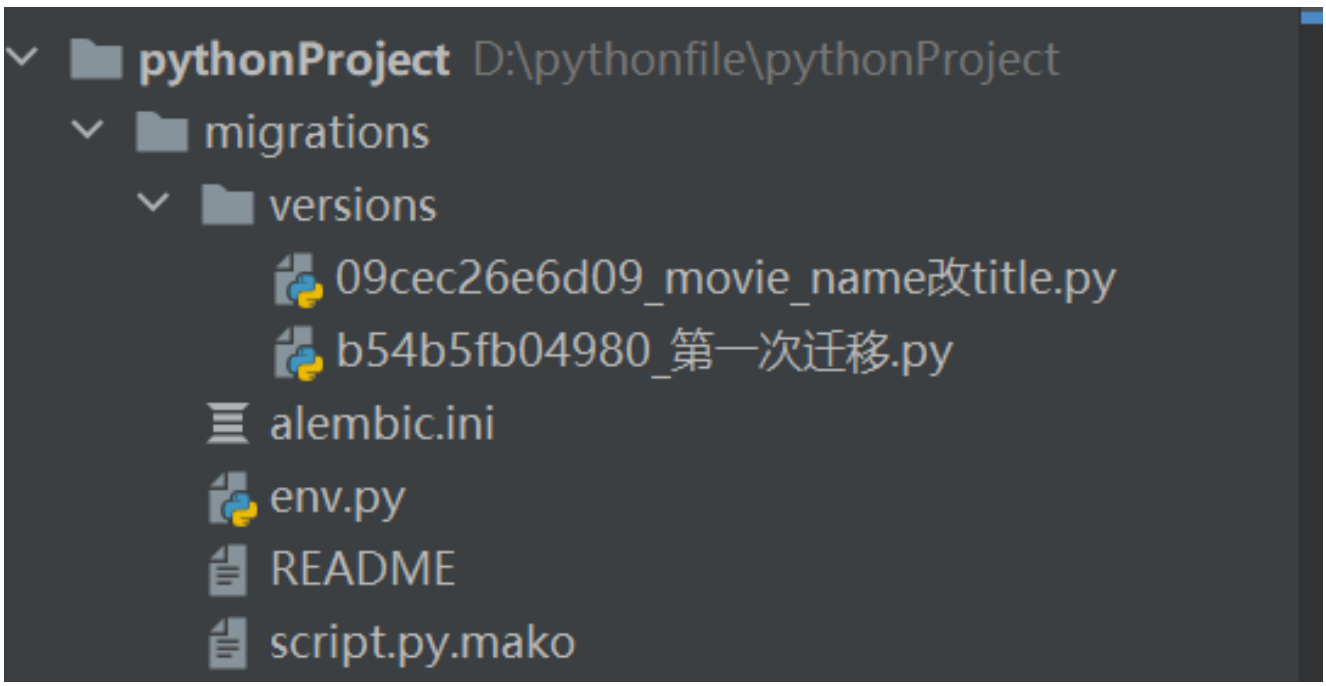
```
# 只有执行迁移之后才会真的在数据库中生成表。

python kuoza.py db upgrade
```

```
(flask_bei) D:\pythonfile\pythonProject>python kuoza.py db upgrade
INFO  [alembic.runtime.migration] Context impl MySQLImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.runtime.migration] Running upgrade  -> b54b5fb04980, '第一次迁移'

(flask_bei) D:\pythonfile\pythonProject>
```

```
mysql> show tables;
Empty set (0.00 sec)

mysql> show tables;
+-----------------+
| Tables_in_py    |
+-----------------+
| alembic_version |
| cast            |
| movie           |
+-----------------+
3 rows in set (0.00 sec)
```

修改了模型类需要重新生成迁移文件，然后执行迁移。
将Movie模型类中的name字段改成title,每次重新迁移都会生成一个文件。

```
(flask_bei) D:\pythonfile\pythonProject>python kuoza.py db migrate -m 'movie_name改title'
INFO  [alembic.runtime.migration] Context impl MySQLImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.autogenerate.compare] Detected added column 'movie.title'
INFO  [alembic.autogenerate.compare] Detected removed column 'movie.name'
Generating D:\pythonfile\pythonProject\migrations\versions\09cec26e6d09_movie_name改title.py ...  done

(flask_bei) D:\pythonfile\pythonProject>python kuoza.py db upgrade
INFO  [alembic.runtime.migration] Context impl MySQLImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.runtime.migration] Running upgrade b54b5fb04980 -> 09cec26e6d09, 'movie_name改title'
```



6.回退数据库

查看迁移历史记录

```
python kuoza.py db history
```

```
(flask_bei) D:\pythonfile\pythonProject>python kuoza.py db history
b54b5fb04980 -> 09cec26e6d09 (head), 'movie_name改title'
<base> -> b54b5fb04980, '第一次迁移'
```

查看迁移历史记录

```
python hello_wolrd.py db downgrade 版本号

迁移记录，版本号，提交备注
```

比如我们要回退到第一次迁移:

```
python kuoza.py db downgrade b54b5fb04980
```

```
(flask_bei) D:\pythonfile\pythonProject>python kuoza.py db downgrade b54b5fb04980
INFO  [alembic.runtime.migration] Context impl MySQLImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.runtime.migration] Running downgrade 09cec26e6d09 -> b54b5fb04980, 'movie_name改title'
```

```
mysql> show create table movie;
+--------+-------------------------------------------------------------------------------------------------------------------------------+
| Table  | Create Table                                                                                                                  |
+--------+-------------------------------------------------------------------------------------------------------------------------------+
| movie  | CREATE TABLE `movie` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+--------+-------------------------------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

回退前

```
mysql> show create table movie;
+--------+-------------------------------------------------------------------------------------------------------------------------------+
| Table  | Create Table                                                                                                                  |
+--------+-------------------------------------------------------------------------------------------------------------------------------+
| movie  | CREATE TABLE `movie` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`)
```

回退后