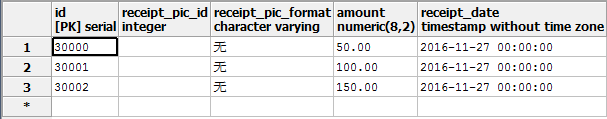
使用数据库:Postgresql 使用ORM:SqlAlchemy

1. 首先要连接数据库

网上有答案,注意数据库的用户名,密码，和连接数据库的名字要根据自己的数据库的设置进行更改

搜索关键词:python postgresql

1. 进行数据库相关的操作



以上面那张表为例:

创建表的结构 Receipt.py # 定义模型 收据

Class Receipt(db.Model):

\_\_tablename\_\_ = ‘receipts’ //数据表的名称

id = db.Column(db.Integer,primary\_key=True) # 主键

receipt\_pic\_id = db.Column(db.integer)

下面代码相似。

首先要导入sqlalchemy包 和 Receipt.py

生成对象 例如: db=SQLalchemy()

然后根据主键也就是本表中ID进行操作:

添加操作：

def add\_receipt(self, id, receipt\_pic\_id)： # 类属性使用构造方法赋值

receipt = Receipt(

id = id

receipt\_pic\_id = receipt\_pic\_id

)

db.session.add(receipt) # 会话

db.session.commit() # 向数据库提交会话

查询操作:

通过主键进行查询

Info = db.session.query(Receipt).filter(Receipt.id = id ).first()

返回的是一个对象 获得receipt\_pic\_id 只需要 info. receipt\_pic\_id即可获得

Infos = db.session.query(Receipt).filter(Receipt. receipt\_pic\_id.is\_(None)).all()

返回的是一个数组对象,.receipt\_pic\_id不为空的所有数据表 利用for循环加上“点”操作

获得相应的数据

以上为常用的方法，详细

http://blog.chedushi.com/archives/2120

删除操作:

def delete\_receipt(self,id, receipt\_pic\_id):

receipt = db.session.query(Receipt).filter(Receipt.id == id).first()

db.session.delete(receipt) # receipt为一Recepit实例

db.session.commit()

更新操作:

def update\_receipt(self,id, new\_receipt\_pic\_id):

db.session.query(Receipt).filter(Receipt.id==id).update({‘receipt\_pic\_id’, new\_receipt\_pic\_id }) # query出的数据为对象

db.session.commit()

基础的有关数据库的操作就是这些，应该够用了，如果还有不会的上网稍微搜索下就可以。

关于在web上数据的获取 # p12 request请求上下文，请求对象，http请求

框架:Flask

对于表单(form)的获取数据:

request.form['abc'] 是获取表单中的数据。 建议使用request.form.get('abc','default value'). 理由：如果表单中没有'abc', form.get('abc','default value') 会返回’default value'. 如果用 form.get('abc'）则返回None; 如果用 form['abc'], 则抛出 400 异常，整个http request 都不可用。

对于勾选(checkbox)的获取数据:

requst.values.getlist(“\*”) //\*表示为在HTML中checkbox的name

以上的情况为主要常用的情况，详细

http://blog.csdn.net/wangqing008/article/details/39183153

将参数写到URL上的方法，

构造 URL 参数

你也许经常想为 URL 的查询字符串(query string)传递某种数据。如果你是手工构建 URL，那么数据会以键/值对的形式置于 URL 中，跟在一个问号的后面。例如， httpbin.org/get?key=val。 Requests 允许你使用 params 关键字参数，以一个字典来提供这些参数。举例来说，如果你想传递key1=value1 和 key2=value2 到 httpbin.org/get ，那么你可以使用如下代码：

>>> payload = **{**'key1'**:** 'value1'**,** 'key2'**:** 'value2'**}**

>>> r = requests.get**(**"http://httpbin.org/get"**,** params=payload**)**

通过打印输出该 URL，你能看到 URL 已被正确编码：

>>> **print(**r.url**)**

http://httpbin.org/get?key2=value2&key1=value1

获取URL 上的参数、

当你构造的路由为这种格式的时候,利用这种方法获取路由上的参数

@app.route**(**'/user/<username>'**)**

**def** show\_user\_profile**(**username**):**

*# show the user profile for that user*

**return** 'User %s' % username