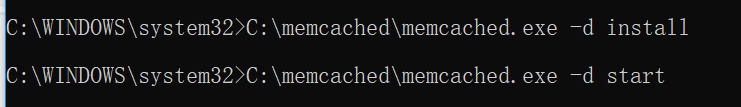
worker3：mysql以及nosql配置，设计数据库（表的属性），数据服务器

* 使用Mysql存储数据~~，将mySQL部署在一台PC作为数据服务器~~
* 使用memchache缓存

Todo

* 设计Mysql数据库
  + 角色：用户（买家卖家）、商品管理员
  + 商品：商品名，卖家账户名，底价，状态（未开卖，正拍卖，已卖出），最终成交价，开拍时间
  + 用户：账户名，密码，昵称，资金
  + 角色：账户名，角色
  + 拍卖情况：商品名，最新出价买家账户名（未卖出为NULL），最新价
* ~~学习Django访问远程数据服务器~~
* 学习如何使用memchache缓存

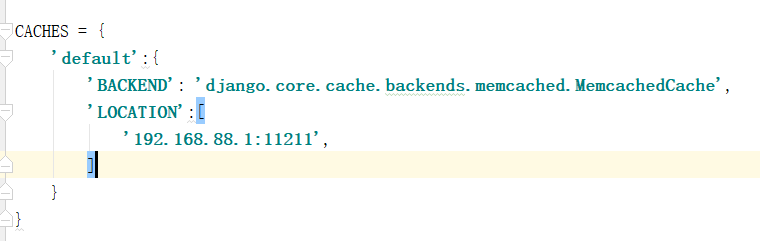
安装和启动memcached服务



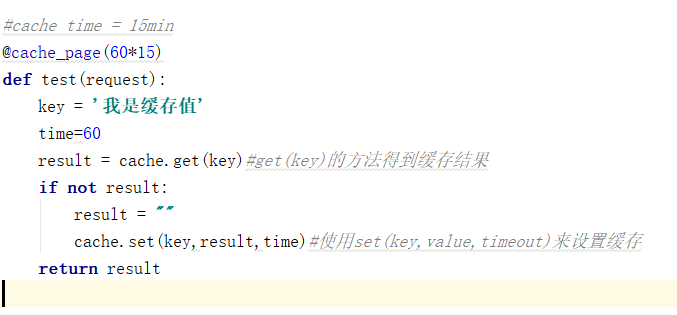
|  |
| --- |
| 启动服务：memcached -d -m 128 -p 11211 -u root  参数的具体解释为：  -p 监听的端口  -l 连接的IP地址, 默认是本机  -d start 启动memcached服务  -d restart 重起memcached服务  -d stop|shutdown 关闭正在运行的memcached服务  -d install 安装memcached服务  -d uninstall 卸载memcached服务  -u 以的身份运行 (仅在以root运行的时候有效)  -m 最大内存使用，单位MB。默认64MB  -M 内存耗尽时返回错误，而不是删除项  -c 最大同时连接数，默认是1024  -f 块大小增长因子，默认是1.25-n 最小分配空间，key+value+flags默认是48  -h 显示帮助 |

在django中配置：

Settings.py中加入：



缓存的使用：

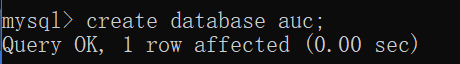


* Django连接mySQL

1.安装mysql server 5.6

Root password:yueyue

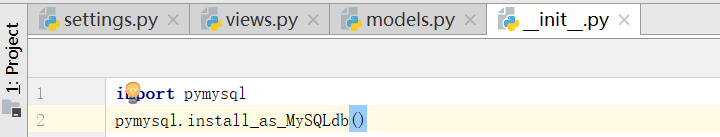
2.创建数据库auc



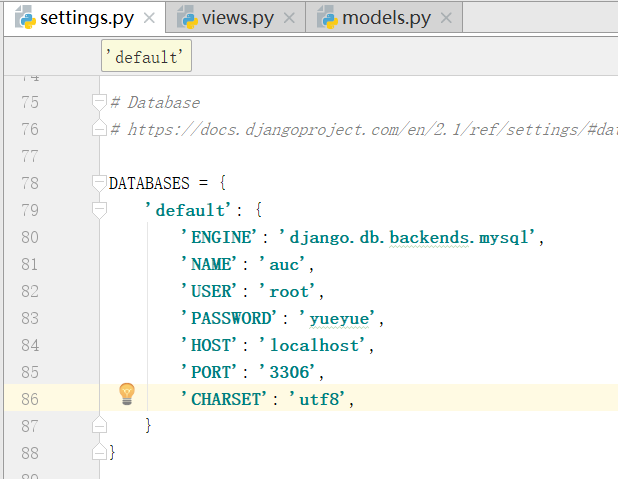
3.在pycharm添加数据库源

4.pip install pymysql

5.init文件：



6.settings.py

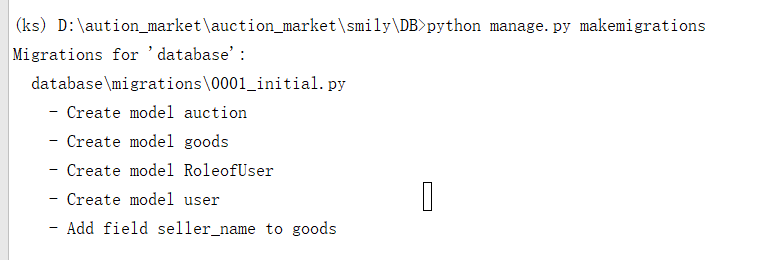


7.models.py中编写数据库逻辑，一个类对应一个表

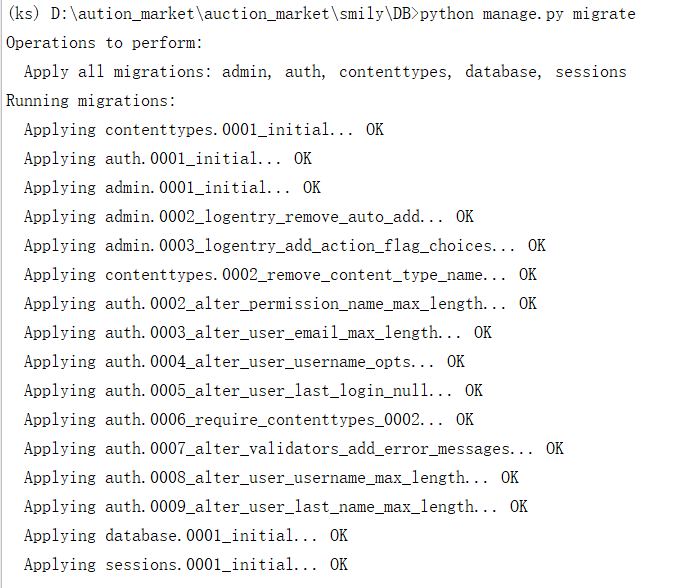
* + 商品：商品名，卖家账户名，底价，状态（未开卖，正拍卖，已卖出），开拍时间，最终成交价
  + 用户：账户名，密码，昵称，资金，状态（正常0、封禁1）
  + 角色：账户名，角色（普通用户、商品管理-下架商品and封禁账号）
  + 拍卖情况：商品名，最新出价买家账户名（未卖出为NULL），最后出价时间，最新价

8.命令行构建表结构：

Python manage.py makemigrations



Python manage.py migrate



表操作：

|  |
| --- |
| # 增  #  # models.Tb1.objects.create(c1='xx', c2='oo') 增加一条数据，可以接受字典类型数据 \*\*kwargs    # obj = models.Tb1(c1='xx', c2='oo')  # obj.save()    # 查  #  # models.Tb1.objects.get(id=123) # 获取单条数据，不存在则报错（不建议）  # models.Tb1.objects.all() # 获取全部  # models.Tb1.objects.filter(name='seven') # 获取指定条件的数据    # 删  #  # models.Tb1.objects.filter(name='seven').delete() # 删除指定条件的数据    # 改  # models.Tb1.objects.filter(name='seven').update(gender='0') # 将指定条件的数据更新，均支持 \*\*kwargs  # obj = models.Tb1.objects.get(id=1)  # obj.c1 = '111'  # obj.save() # 修改单条数据 |