安装说明

1. 安装Python解析器

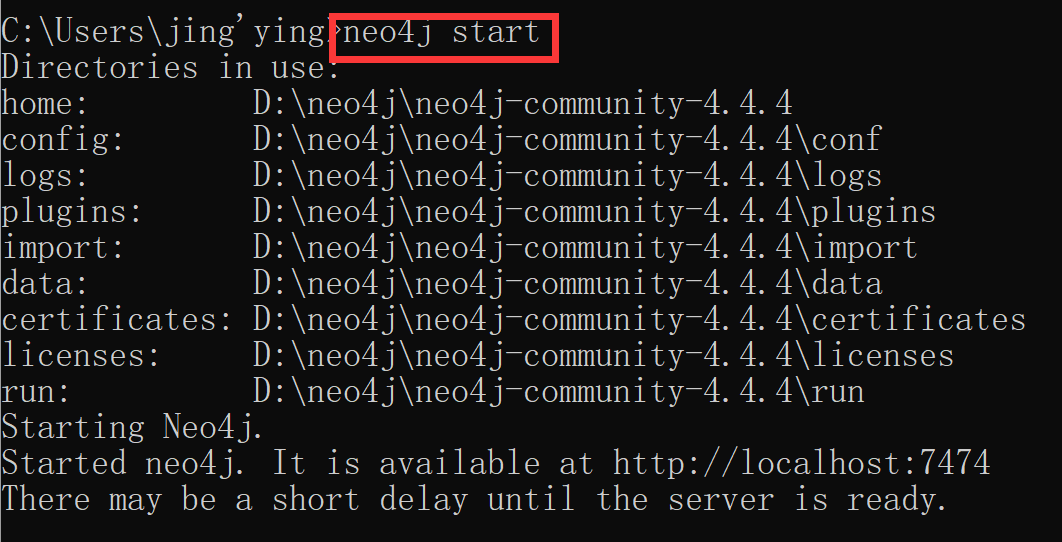
首先需要安装Python解析器来运行本软件系统。运行所需的Python解析器最低版本为Python 3.8，推荐使用最新的Python 3.11.0。

2. 安装代码编辑器与数据库

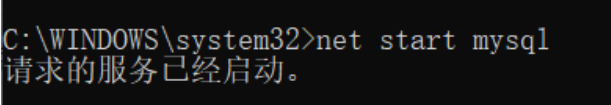
其次需要安装代码编辑器配置所需要运行的环境，推荐安装PyCharm Community 2022.1.1。数据库方面，安装MySQL与Neo4j数据库，推荐版本为MySQL 8.0.28、Neo4j 5.2.0。安装MySQL Workbench编辑器，推荐版本为MySQL Workbench 8.0.30。

3. 启动数据库服务

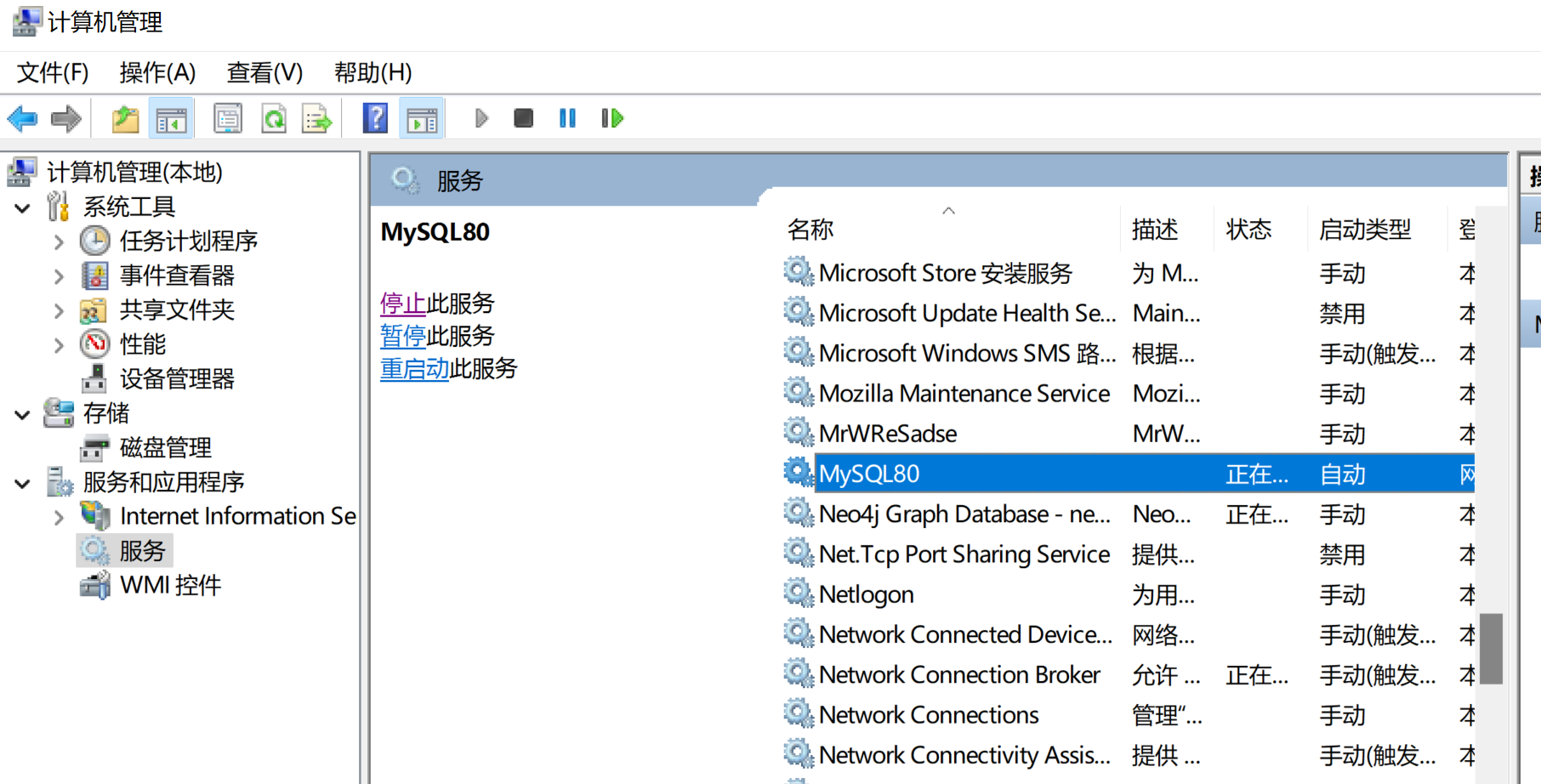
将Neo4j数据库文件的bin目录路径加入到系统的环境变量中。打开命令提示符(cmd)或者Windows PowerShell输入并执行neo4j start命令即可启动Neo4j数据库。



以管理员身份打开命令提示符(cmd) 或者Windows PowerShell输入并执行net start mysql命令即可启动MySQL数据库。



或者是右键我的电脑，点击管理，在页面中点击服务和应用程序下拉菜单，点击服务，找到MySQL，右键点击启动即可启动MySQL数据库。



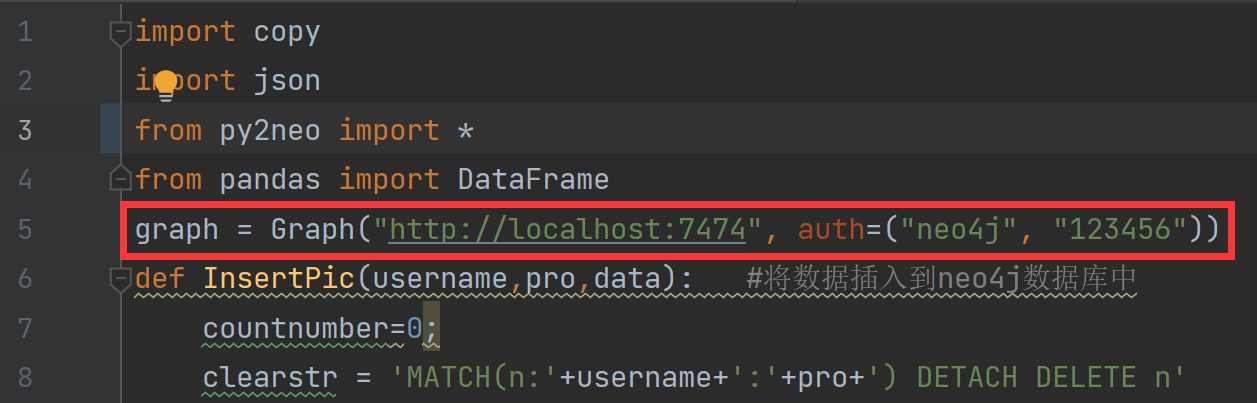
4．在MySQL中创建数据库与表格

在MySQL Workbench界面创建数据库，执行项目工程文件夹中的CreateTable.sql文件或者执行下列SQL语句创建存储数据所需的关系表格。

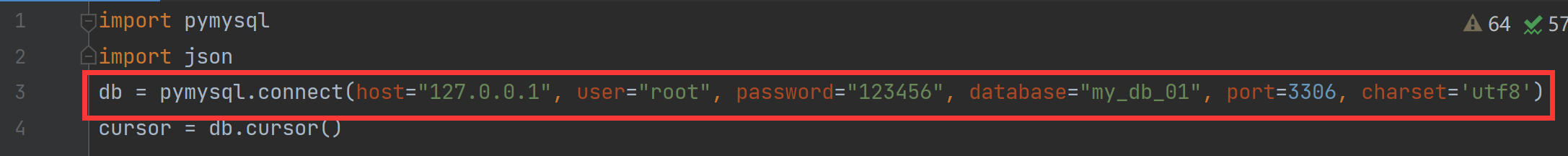
|  |
| --- |
| -- 创建推荐布局数据表格  CREATE TABLE `laypicdata` (  `layname` char(20) NOT NULL,  `laydata` text,  PRIMARY KEY (`layname`)  ) ;  -- 创建用户知识图谱工程数据表格  CREATE TABLE `picdata` (  `Userpro` char(20) NOT NULL,  `PicData` text,  PRIMARY KEY (`Userpro`)  ) ;  -- 创建用户名密码数据表格  CREATE TABLE `usermessage` (  `Username` varchar(45) NOT NULL,  `Password` varchar(45) NOT NULL,  PRIMARY KEY (`Username`),  UNIQUE KEY `Username\_UNIQUE` (`Username`)  ); |

5. 修改代码文件中的数据库信息

使用代码编辑器例如PyCharm以工程方式打开系统代码文件夹。将Neo4j.py中的数据库信息设置为运行此系统电脑中Neo4j的数据库信息。其中第一个参数是Neo4j数据库的本地网页端的URL地址。Auth中的两个参数分别是Neo4j数据库的用户名与密码。Neo4jRecLayout.py中数据库参数的修改同理。

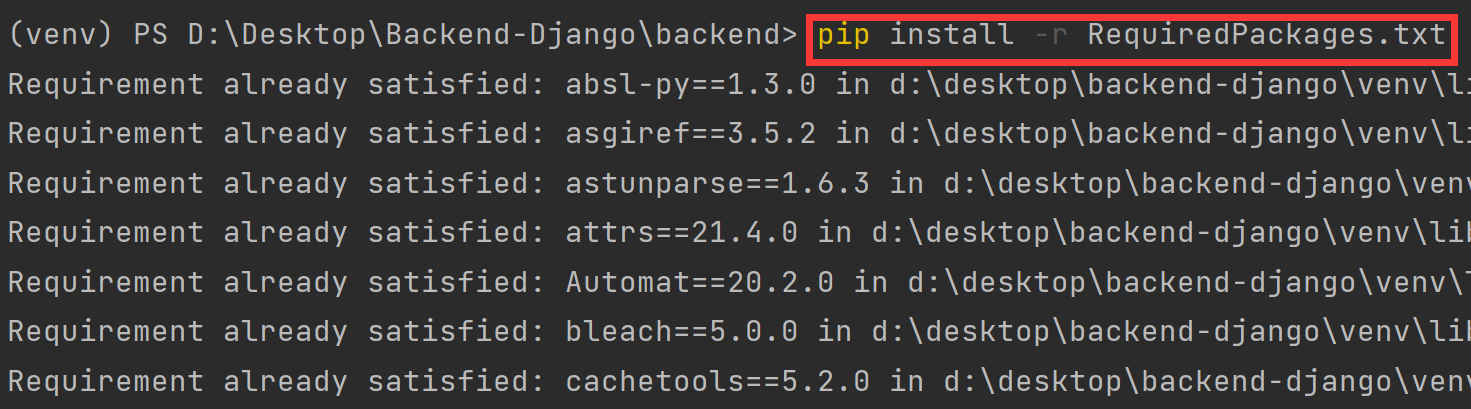


将MySQL.py中的数据库信息设置为运行此系统电脑中的MySQL数据库信息。其中Host参数为连接的MySQL数据库的地址。User参数是MySQL数据库的用户名，Password表示的是对应用户名的密码。Database参数是要操作的数据库，Port为MySQL指定的端口(默认为3306)，Charset参数为编码方式。



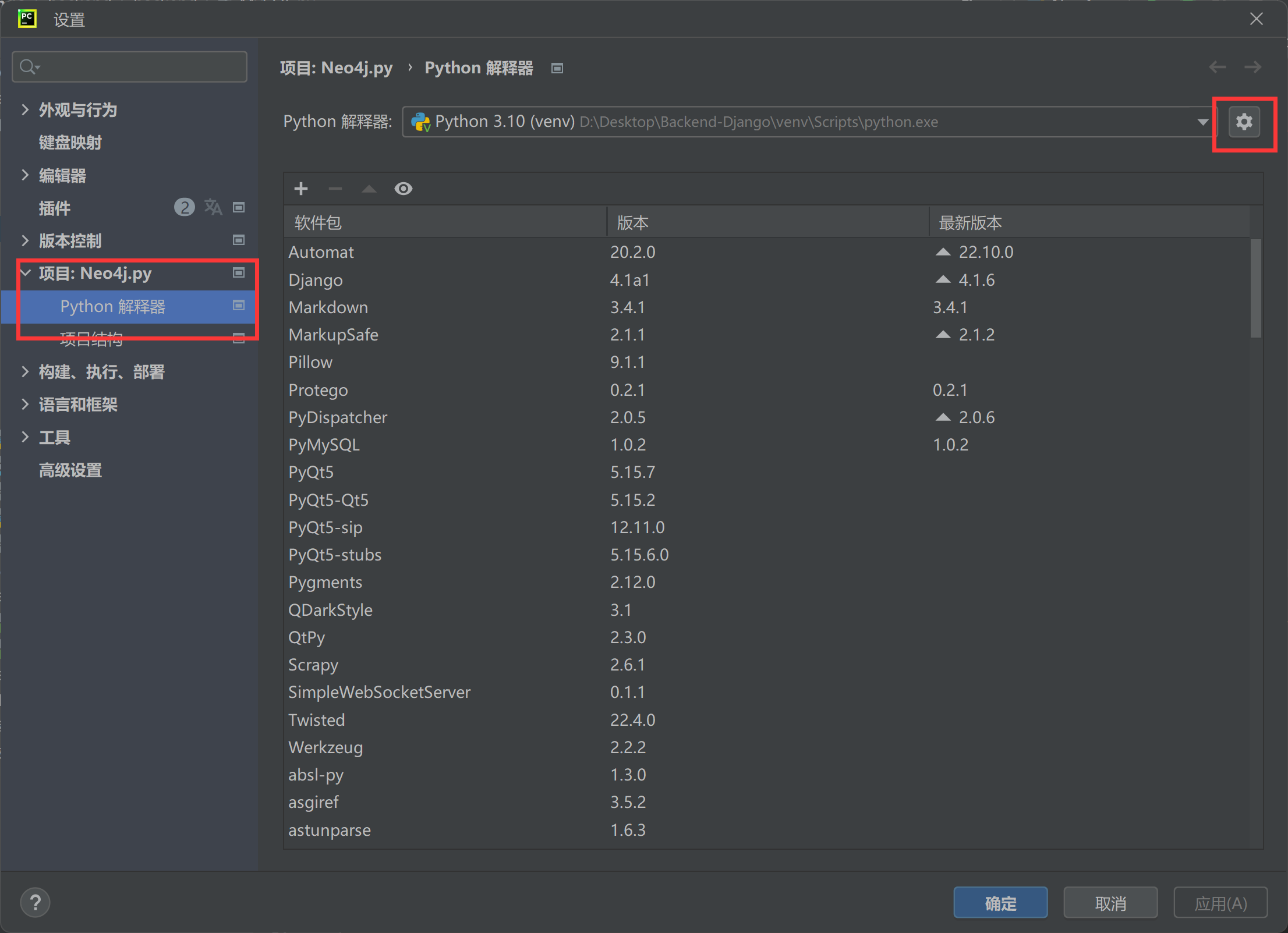
6．配置系统运行所依赖的第三方开发包

在代码编辑器界面中（以PyCharm为例），找到backend文件夹，右键打开于终端。在下方的终端界面中输入并执行指令pip install -r RequiredPackages.txt即可安装系统所需要的第三方依赖包。



注意，这里的第三方的开发包是默认安装在你现在编辑器选中的虚拟环境中的，如果需要一个独立开发环境，请手动在编辑器中进行设置。这里仍然以PyCharm编辑器为例说明。

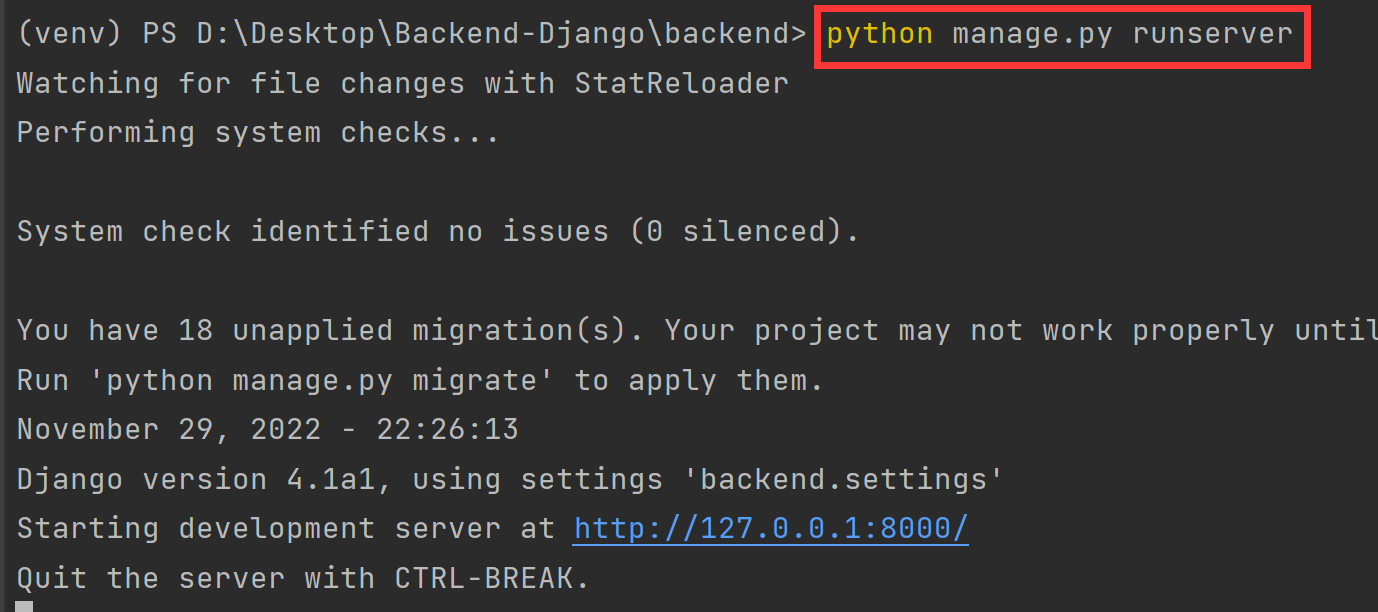
先在工具栏中的文件-设置部分中，找到python解析器。在此界面中可以看到自己所选中的虚拟开发环境以及在此环境中所安装的所有第三方开发包。



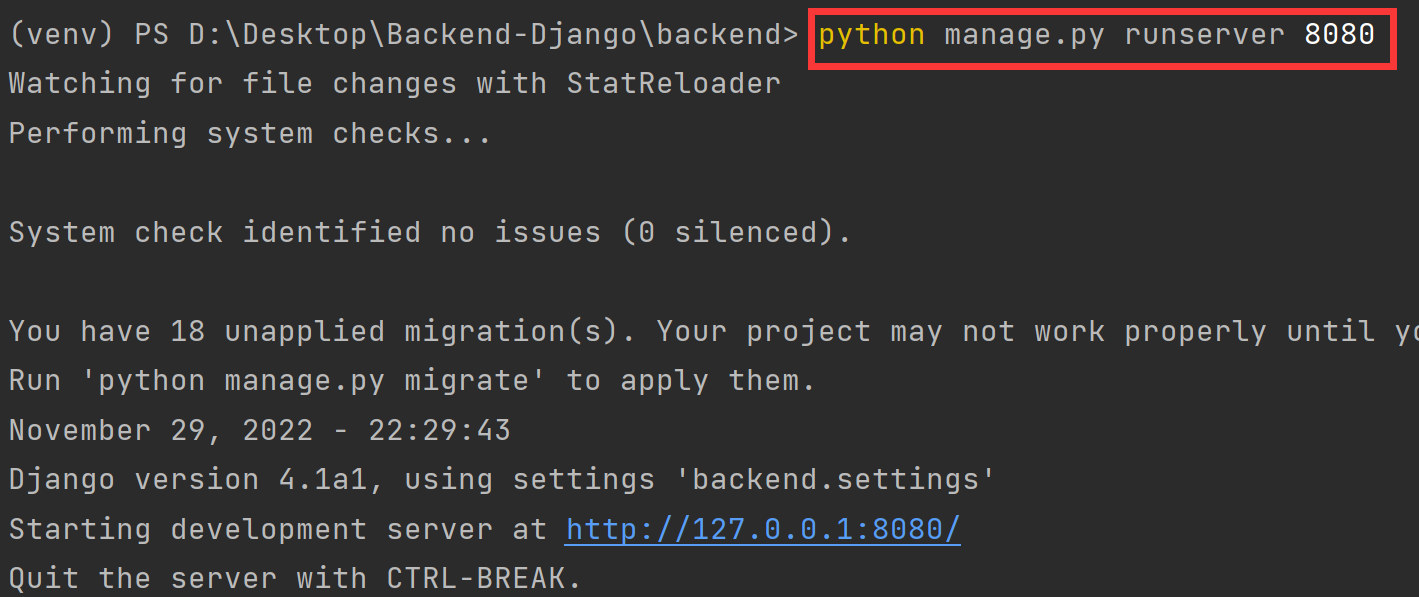
如若需要新建一个虚拟开发环境。点击上图中右侧的齿轮按钮，选择添加，在弹出的设置界面中进行配置即可。

7. 启动服务进程

同样，在刚在执行pip install -r RequiredPackages.txt命令的终端目录下，输入并执行python manage.py runserver命令，系统的服务进行便开始运行。



注意：在端口配置上，默认运行是在8000端口，如果需要更改相应进程的端口号，请在python manage.py runserver后加上需要的端口号。至此，本系统从安装到实际运行便完成了。



8. 接口说明

在urls.py文件中规定了相应函数的接口名称，本系统的接口函数名称与View模块中的函数名称一样，方便用户理解与使用。具体的接口名称与功能如下所示。

|  |
| --- |
| urlpatterns = [  path('InsertPic/', view.InsertPic), #插入图谱工程数据  path('InsertOneNode/', view.InsertOneNode), #插入单个结点  path('InsertOneRel/', view.InsertOneRel), #插入单个关系  path('InsertUser/',view.InsertUser), #插入注册用户信息  path('InsertRecLay/',view.InsertRecLay), #插入推荐布局信息  #插入推荐布局信息（结点属性）  path('InsertRecLayByNode/',view.InsertRecLayByNode),  #插入推荐布局信息（结点关系图）  path('InsertRecLayByDis/',view.InsertRecLayByDis),  #向MySQL数据库中插入推荐布局信息  path('InsertRecLayMy/',view.InsertRecLayMy),  path('SelectPic/', view.SelectPic), #查询知识图谱工程信息  path('SelectPass/', view.SelectPass), #查询用户密码  path('SelectRecLay/', view.SelectRecLay), #查询推荐布局信息  path('SelectOnePro/',view.SelectOnePro), #查询单个工程信息  path('SelectOneRel/',view.SelectOneRel), #查询单个关系  #查询单个结点没有与其有关系的结点  path('SelectOneNodeNoRel/',view.SelectOneNodeNoRel),  #查询单个结点与其有关系的结点（此结点作为关系的入度结点）  path('SelectOneNodeAndRelIn/',view.SelectOneNodeAndRelIn),  #查询单个结点与其有关系的结点（此结点作为关系的出度结点）  path('SelectOneNodeAndRelOut/',view.SelectOneNodeAndRelOut),  #查询单个结点信息  path('SelectOneNodeByName/', view.SelectOneNodeByName),  #查询两个结点之间的所有通路  path('SelectTwoNodePath/',view.SelectTwoNodePath),  #查询两个结点之间的最短路径  path('SelectTwoNodeShortPath/', view.SelectTwoNodeShortPath),  #查询两个结点之间的所有最短路径  path('SelectTwoNodeAllShortPath/', view.SelectTwoNodeAllShortPath),  #查询经过一个结点的回路  path('SelectCircuit/', view.SelectCircuit),  path('SelectDeepNode/',view.SelectDeepNode),#查询指定深度的所有结点信息  #查询一个类型的结点与其有关系的结点  path('SelectOneTypeAndRel/',view.SelectOneTypeAndRel),  #查询一个类型的结点与其有关系的结点（所有结点都是这个类型）  path('SelectOneTypeNoRel/',view.SelectOneTypeNoRel),  #查询一个颜色的结点与其有关系的结点  path('SelectOneClourAndRel/',view.SelectOneClourAndRel),  #查询一个颜色的结点与其有关系的结点（所有结点都是这个颜色）  path('SelectOneClourNoRel/',view.SelectOneClourNoRel),  #查询推荐布局信息  path('SelectRecLayByNode/',view.SelectRecLayByNode),  path('SelectRecLayByDis/',view.SelectRecLayByDis),  path('DeletLay/',view.DeleteLay), #删除一个推荐布局信息  path('DeletePro/',view.DeletePro), #删除用户一个工程图谱信息  path('DeleteAllPro/',view.DeleteAllPro), #删除用户所有工程图谱信息  path('DeleteOneRel/',view.DeleteOneRel), #删除结点间的单个关系  path('DeleteAllRel/',view.DeleteAllRel), #删除两个结点之间所有关系  #删除单个结点（强制）  path('DeleteOneNodeCom/',view.DeleteOneNodeCom),  #删除单个结点（非强制）  path('DeleteOneNodeByName/',view.DeleteOneNodeByName),  #删除结点的一个属性  path('DeleteNodeAtt/',view.DeleteNodeAtt),  #删除结点的一个标签  path('DeleteNodeLab/',view.DeleteNodeLab),  #删除一个颜色的所有结点  path('DeleteOneClourNode/',view.DeleteOneClourNode),  #删除一个类型的所有结点  path('DeleteOneTypeNode/',view.DeleteOneTypeNode),  path('AddNodeAtt/', view.AddNodeAtt), #给结点增加一个属性  path('AddNodeLab/', view.AddNodeLab), #给结点增加一个标签  path('SetNodeAtt/', view.SetNodeAtt), #重置结点一个属性  path('SetRelName/', view.SetRelName), #重置两个结点之间关系  #根据时间更新工程图谱数据信息  path('UpdataProByTime/',view.UpdataProByTime),  #根据时间更新推荐布局图谱数据信息  path('UpdataLayByTime/',view.UpdataLayByTime),] |

9. 几点补充

在系统的文件夹中有Request test.html文件，此文件是为测试系统中各种功能所编写的。请求发送到的端口为8000端口，返回值为进行布局解析的JSON数据（以字符串形式返回）。用户可以根据需要进行参考。