



# Django编程实践



壬寅



什么是Django?

Django 是一个由 Python 编写的一个开放源代码的 Web 应用框架。

使用 Django, 只要很少的代码, Python 的程序开发人员就可以轻松地完成一个正式网站所需要的大部分内容, 并进一步开发出全功能的 Web 服务

# 目录

/contents



## 壹 • 安装Django

1. 环境准备
2. 创建项目
3. 浏览器访问
4. Django实例

## 贰 • Linux 安装Django

1. 安装mysql
2. 安装Django及发布



安装 Django

# 环境准备

1. 新建虚拟环境, 打开spyder (anaconda3) 的命令行界面, kjxy (创建一个名为kjxy的虚拟环境).

```
Anaconda Prompt (anaconda3)
(base) C:\Users\xiaopihai>conda create -n kjxy
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 4.12.0
Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\xiaopihai\anaconda3\envs\kjxy

Proceed ([y]/n)? y      这里输入y然后回车
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate kjxy
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) C:\Users\xiaopihai>_
```

# 环境准备

2. 激活环境 -----conda activate kjxy
3. 为当前环境安装Django, 安装过程中需要按y确认安装

```
(kjxy) C:\Users\xiaopihai>conda activate kjxy  
(kjxy) C:\Users\xiaopihai>
```

```
(kjxy) C:\Users\xiaopihai>conda install django  
Collecting package metadata (current_repodata.json): done  
Solving environment: done  
  
==> WARNING: A newer version of conda exists. <==  
  current version: 4.10.1  
  latest version: 4.12.0  
  
Please update conda by running  
  
  $ conda update -n base -c defaults conda  
  
# All requested packages already installed.  
  
(kjxy) C:\Users\xiaopihai>
```

# 创建项目

1. 在本地磁盘中创建 web 的目录, 进入 web, 创建django项目

```
(kjxy) C:\Users\xiaopihai>
(kjxy) C:\Users\xiaopihai>
(kjxy) C:\Users\xiaopihai>
(kjxy) C:\Users\xiaopihai>D:

(kjxy) D:\>cd stuwork\python\web

(kjxy) D:\stuwork\python\web>django-admin startproject ehr01_
```

# 创建项目

## 2. 进入项目

## 3. 迁移django到当前项目

```
(kjxy) D:\web\dw1>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK

(kjxy) D:\web\dw1>
```



# 创建项目

## 4. 启动项目

```
(kjxy) D:\web\dw1>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 23, 2022 - 15:10:50
Django version 3.2.5, using settings 'dw1.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

# 浏览器访问

在浏览器地址栏输入:<http://127.0.0.1:8000>, 弹出此页面即成功搭建 Django 环境



django

View [release notes](#) for Django 3.2

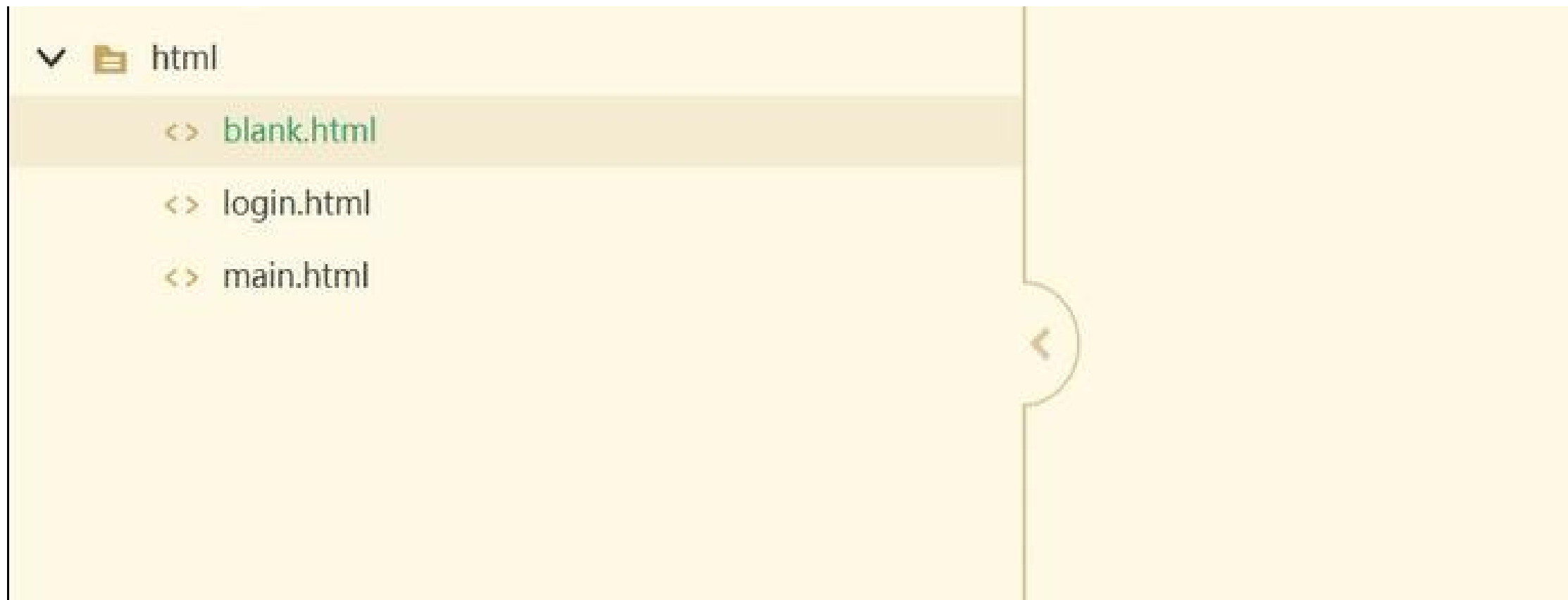


The install worked successfully! Congratulations!

You are seeing this page because [DEBUG=True](#) is in your settings file and you have not configured any URLs.

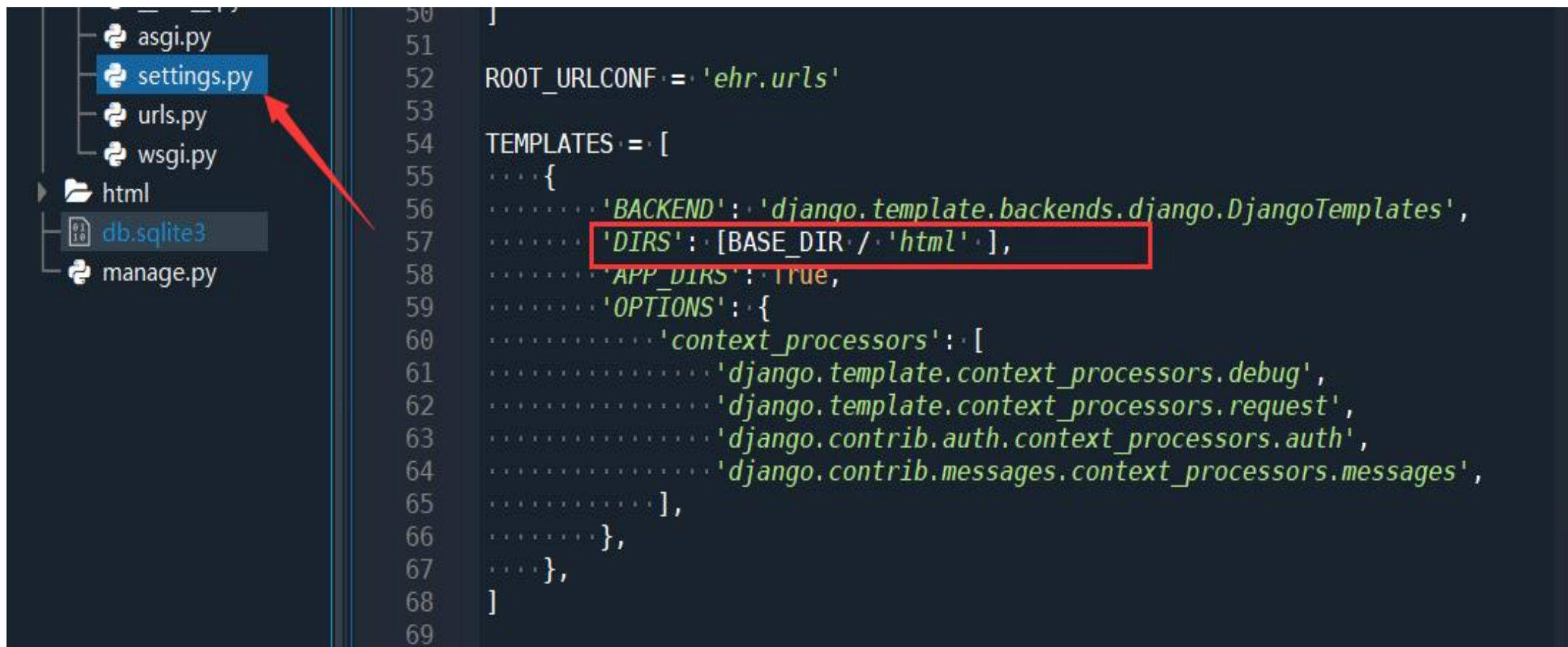
# Django 实例

1. 打开HBuider x软件复制网页套版至html中



# Django实例

打开ehr01目录下的settings.py文件找到  
”TEMPLATES”语句，  
修改DIRS为：’ DIRS’ : [BASE\_DIR / ‘html’ ],



```
50 ]
51
52 ROOT_URLCONF = 'ehr.urls'
53
54 TEMPLATES = [
55     {
56         'BACKEND': 'django.template.backends.django.DjangoTemplates',
57         'DIRS': [BASE_DIR / 'html'],
58         'APP_DIRS': True,
59         'OPTIONS': {
60             'context_processors': [
61                 'django.template.context_processors.debug',
62                 'django.template.context_processors.request',
63                 'django.contrib.auth.context_processors.auth',
64                 'django.contrib.messages.context_processors.messages',
65             ],
66         },
67     },
68 ]
69
```

# Django实例

打开LoginViews, 添加如下代码

```
10 def index(request):
11     ... #打开登良路页面
12     ... return render(request, 'login.html')
13
14 from django.views.decorators.csrf import csrf_exempt
15 from ..services import LoginServices as services
16
17 @csrf_exempt ... #装饰器,告诉django,此处无需检查
18 def login(request):
19     ... """
20     ...     校验用户名和密码是否正确,如果正确则跳转到main.html
21     ...     如果用户名或密码错误,则跳回到login.html,并提示用户名或密码错误
22     ... """
23     ... #获取页面数据
24     ... userName=request.POST['userName']
25     ... pwd=request.POST['pwd']
26     ...
27     ... #调用业务逻辑函数
28     ... ins=services.checkUser(userName, pwd)
29     ... print(ins)
30     ...
31     ... #定义变量表示跳转的目录路径
32     ... topage='login.html'
33     ... #定义变量表示为页面的数据
34     ... data={}
35     ... if userName=='admin' and pwd=='admin1234':
36     ...     topage='main.html'
37     ... else:
38     ...     data['msg']='用户名或密码错误'
39     ...
40     ... return render(request, topage, data)
41     ...
```

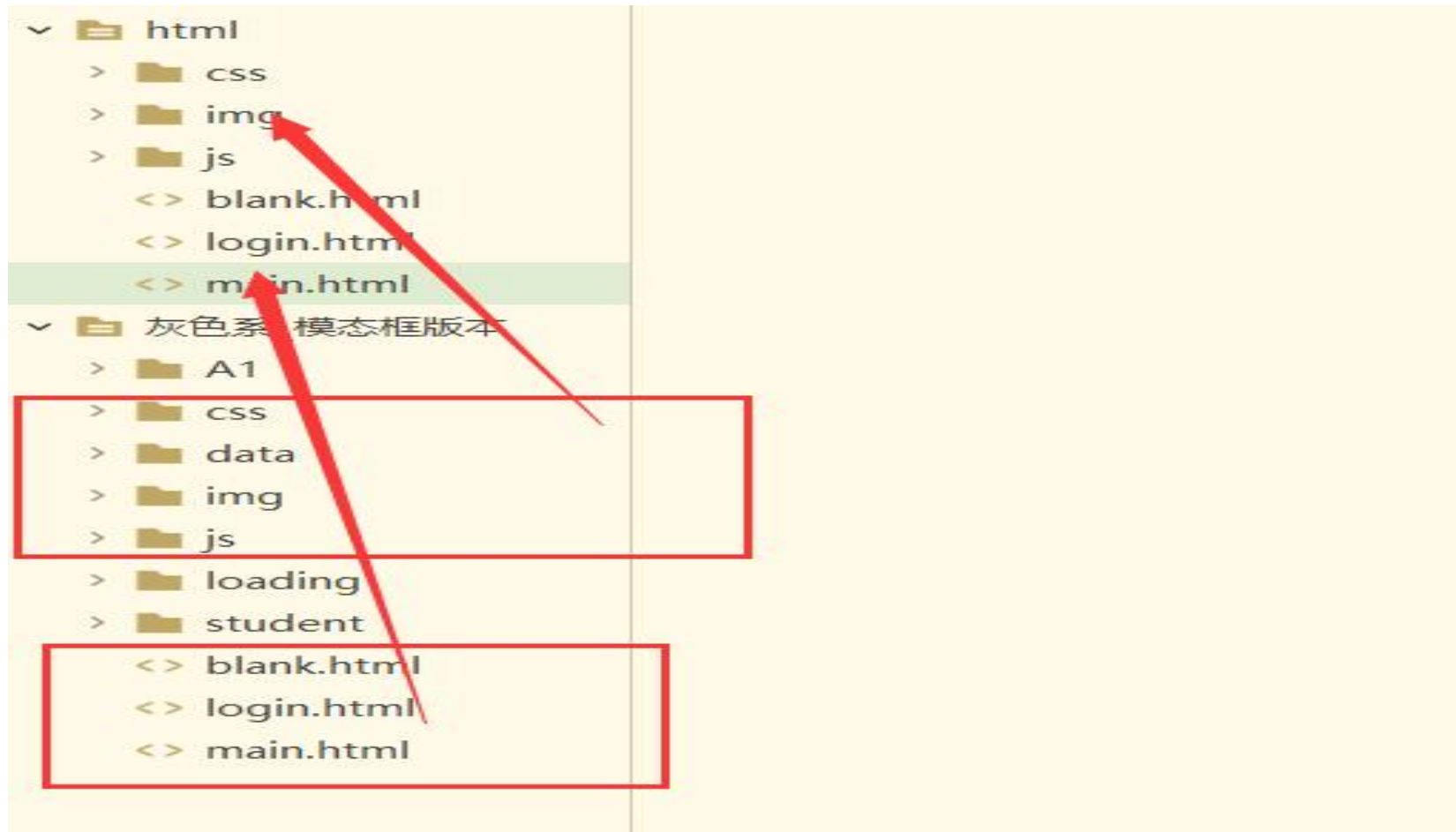
# Django 实例

打开 `urls.py`, 添加如下代码

```
13 ... 1. Import the include() function: from django.urls import include
14 ... 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path
18 from .views import LoginView
19
20 urlpatterns = [
21     # path('admin/', admin.site.urls),
22     path('', LoginView.index),
23     path('index/', LoginView.index),
24     path('login/', LoginView.login),
25 ]
26
27
```

# Django实例

打开HBui lder X将模板下的部分文件粘贴到html中



# Django实例

blank.html代码如下

```
main.html    login.html    blank.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Insert title here</title>
6
7 </head>
8 <body>
9   <table border="0" width="100%" align="center" height="550px">
10  <tr align="center">
11  <td>
12    
13  </td>
14  <td width="10%"></td>
15  </tr>
16 </table>
17 </body>
18 </html>
```



# Django实例

login.html代码如下

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <style>@import url("/static/css/login/login.css"); </style>
7   </head>
8   <body class="login">
9     <div class="login_m">
10      <div class="login_logo">
11        
12      </div>
13      <div class="login_boder">
14        <div class="login_padding">
15          <form action="/login/" method="post" >
16            <h2>用户名</h2>
17            <label>
18              <input type="text" id="username" name="userName" class="txt_input txt_input2" autofocus="true" placeholder="Your name" >
19            </label>
20            <h2>密码</h2>
21            <label>
22              <input type="password" id="userpwd" name="pwd" class="txt_input" placeholder="*****" >
23            </label>
24            <p class="forgot">
25              {{msg}}
26            </p>
27            <div class="rem_sub">
28              <div class="rem_sub_1">
29                <input type="checkbox" name="checkbox" id="save_me">
30                <label for="checkbox">记住</label>
31              </div>
32              <label>
33                <input type="submit" class="sub_button" name="button" id="button" value="登录" >
34              </label>
35            </div>
36          </form>
37        </div>
38      </div><!--login_boder end-->
39    </div>
40  </body>
41 </html>
```

# Django实例

## main.html代码如下(1)

```
4     <meta charset="utf-8">
5     <title></title>
6     <style>@import url("/static/css/main/main.css"); </style>
7     <script type="text/javascript" src="/static/js/jquery341.min.js"></script>
8     <script type="text/javascript" src="/static/js/main/tendina.min.js"></script>
9     <script type="text/javascript" src="/static/js/main/common.js"></script>
10
11 </head>
12 <body>
13 <!--顶部-->
14 <div class="layout_top_header">
15     <div style="float: left"><span style="font-size: 16px;line-height: 45px;padding-left: 20px;color: #a4caac">技术验证平台套版</span></div>
16     <div id="ad_setting" class="ad_setting">
17         <a class="ad_setting_a" href="javascript:;" >
18             <i class="icon-user glyph-icon" style="font-size: 20px"></i>
19             <span>系统管理员</span>
20             <i class="icon-chevron-down glyph-icon"></i>
21         </a>
22         <ul class="dropdown-menu-uu" style="display: none" id="ad_setting_ul">
23             <li class="ad_setting_ul_li"> <a href="javascript:;"><i class="icon-user glyph-icon"></i> 个人中心 </a> </li>
24             <li class="ad_setting_ul_li"> <a href="javascript:;"><i class="icon-cog glyph-icon"></i> 设置 </a> </li>
25             <li class="ad_setting_ul_li"> <a href="login.html"><i class="icon-signout glyph-icon"></i> <span class="font-bold">退出</span> </a> </li>
26         </ul>
27     </div>
28 </div>
29 <!--菜单-->
30 <div class="left">
31     <script type="text/javascript">
32         var myMenu;
33         window.onload = function() {
34             myMenu = new SDMenu("my_menu");
35             myMenu.init();
36         };
```

# Django实例

## main.html代码如下(2)

```
37 | </script>
38 | <div id="my_menu" class="sdmenu">
39 |   <div class="layout_left_menu">
40 |     <ul id="menu">
41 |       <li class="childUlli">
42 |         <a href="student/QueryStudent.html" target="menuFrame"> <i class="glyph-icon icon-reorder"></i>学籍管理</a>
43 |         <ul>
44 |           <li><a href="student/QueryStudent.html" target="menuFrame"><i class="glyph-icon icon-chevron-right"></i>学生管理</a></li>
45 |           <li><a href="student/AddStudent.html" target="menuFrame"><i class="glyph-icon icon-chevron-right"></i>学生添加</a></li>
46 |           <li><a href="student/ViewStudent.html" target="menuFrame"><i class="glyph-icon icon-chevron-right"></i>查看学生</a></li>
47 |           <li><a href="#"><i class="glyph-icon icon-chevron-right"></i>档案查询</a></li>
48 |           <li><a href="#"><i class="glyph-icon icon-chevron-right"></i>档案查询</a></li>
49 |         </ul>
50 |       </li>
51 |       <li class="childUlli">
52 |         <a href="" target="menuFrame"> <i class="glyph-icon icon-reorder"></i>部门管理</a>
53 |         <ul>
54 |           <li><a href="A1/A1010.html" target="menuFrame"><i class="glyph-icon icon-chevron-right"></i>档案管理</a></li>
55 |           <li><a href="A1/A1011.html" target="menuFrame"><i class="glyph-icon icon-chevron-right"></i>部门添加</a></li>
56 |           <li><a href="#"><i class="glyph-icon icon-chevron-right"></i>帮助</a></li>
57 |         </ul>
58 |       </li>
59 |       <li class="childUlli">
60 |         <a href="#"> <i class="glyph-icon icon-location-arrow"></i>菜单管理</a>
61 |         <ul>
62 |           <li><a href="meunbox.html" target="menuFrame"><i class="glyph-icon icon-chevron-right"></i>平台菜单</a></li>
63 |           <li><a href="meunbox_add.html" target="menuFrame"><i class="glyph-icon icon-chevron-right"></i>运行商菜单</a></li>
64 |           <li><a href="XXX" target="menuFrame"><i class="glyph-icon icon-chevron-right"></i>xxxx</a></li>
65 |           <li><a href="XXX" target="menuFrame"><i class="glyph-icon icon-chevron-right"></i>xxxx</a></li>
66 |         </ul>
67 |       </li>
68 |     </ul>
69 |   </div>
70 | </div>
71 | </div>
72 | <div class="Switch"></div>
```

# Django 实例

## main.html 代码如下 (3)

```
73 | <script type="text/javascript">
74 |     $(document).ready(function(e) {
75 |         $(".Switch").click(function(){
76 |             $(".left").toggle();
77 |         });
78 |     });
79 | </script>
80 |
81 | <!-- 工作区 -->
82 | <div id="layout_right_content" class="layout_right_content">
83 |     <div class="mian_content">
84 |         <div id="page_content">
85 |             <iframe id="menuFrame" name="menuFrame" src="/static/blank.html" style="overflow:visible;"
86 |                 scrolling="yes" frameborder="no" width="100%" height="100%"></iframe>
87 |         </div>
88 |     </div>
89 | </div>
90 | <!-- 底部 -->
91 | <div class="layout_footer">
92 |     <p>Copyright © copenedu - 素月轩出品</p>
93 | </div>
94 | <script src="../js/layer/layer.js"></script>
95 |
96 | <script>
97 |     function modal1(childPage){
98 |         //let page="A1/A1011.html";
99 |         layer.open({
100 |             type:2,
101 |             area:["800px","520px"],
102 |             title:"父子操作",
103 |             maxmin:true,
104 |             shadeClose:true,
105 |             content:childPage
106 |         });
107 |     }
108 | </script>
109 | </body>
110 | </html>
111 |
```

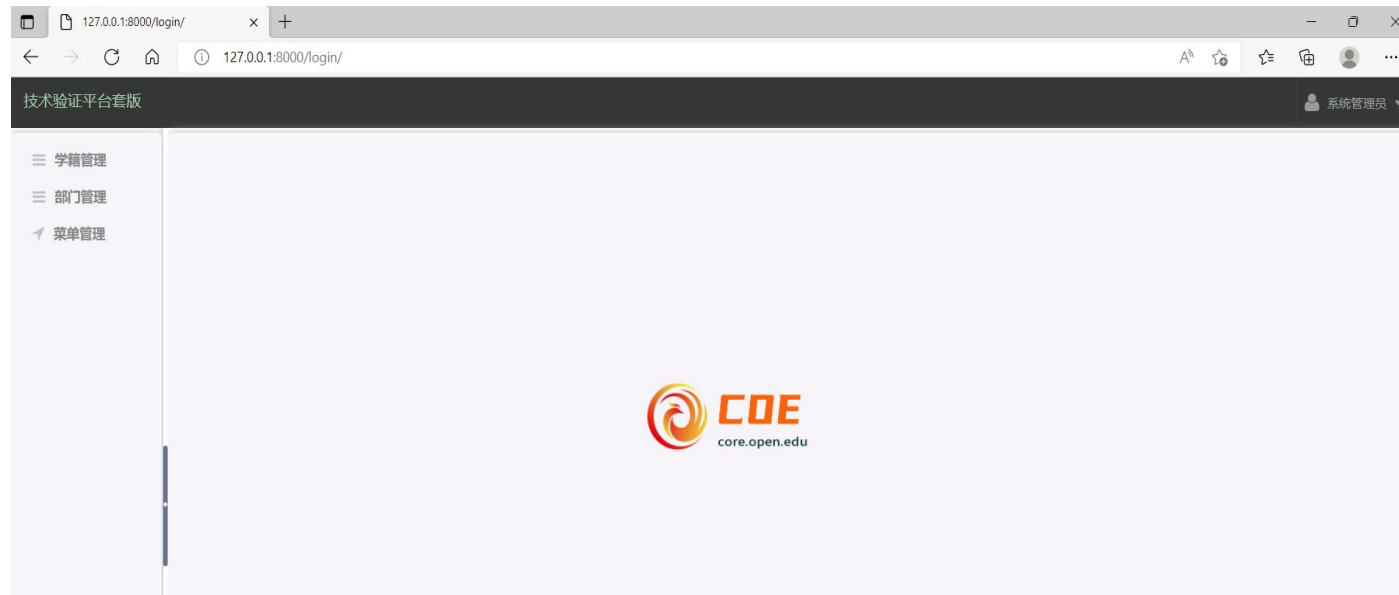
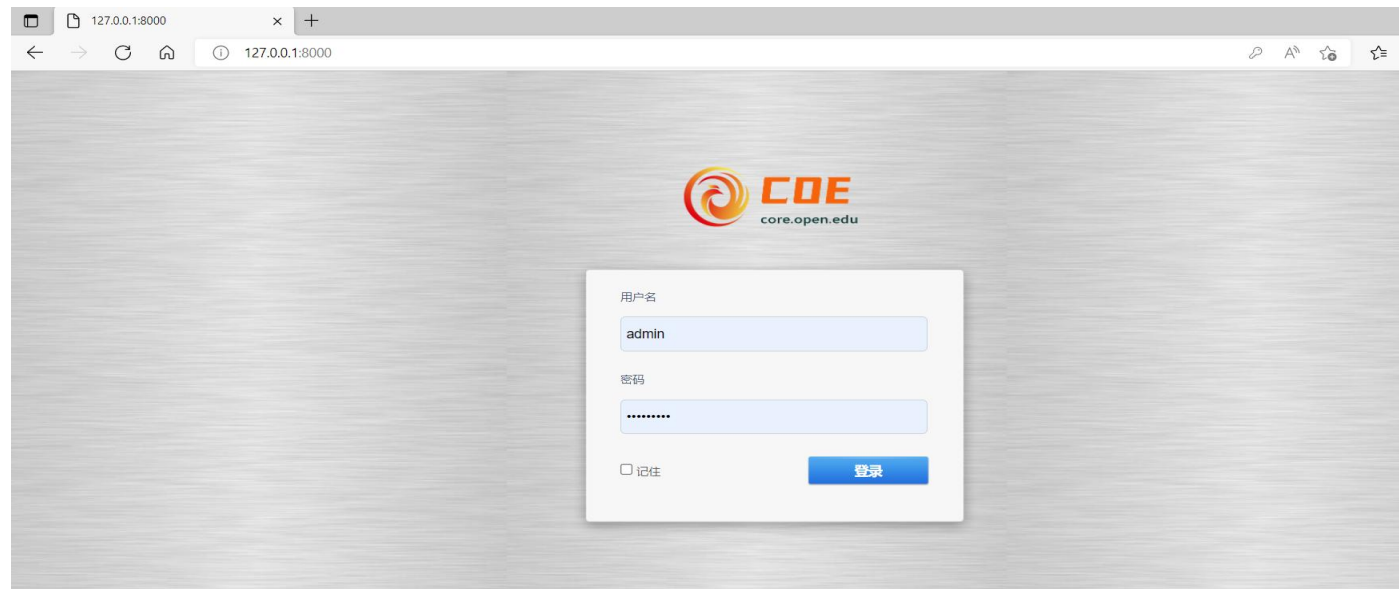
# Django 实例

打开spyder进入views目录在settings.py文件中找到STATIC\_URL = '/static/'  
添加如下代码

```
130
131 STATIC_URL = '/static/'
132 import os
133 STATICFILES_DIRS = (
134     ... # os.path.join(BASE_DIR, 'static'),
135     ...
136     ... os.path.join(BASE_DIR, 'html'), ...
137     ...
138 )
139
```

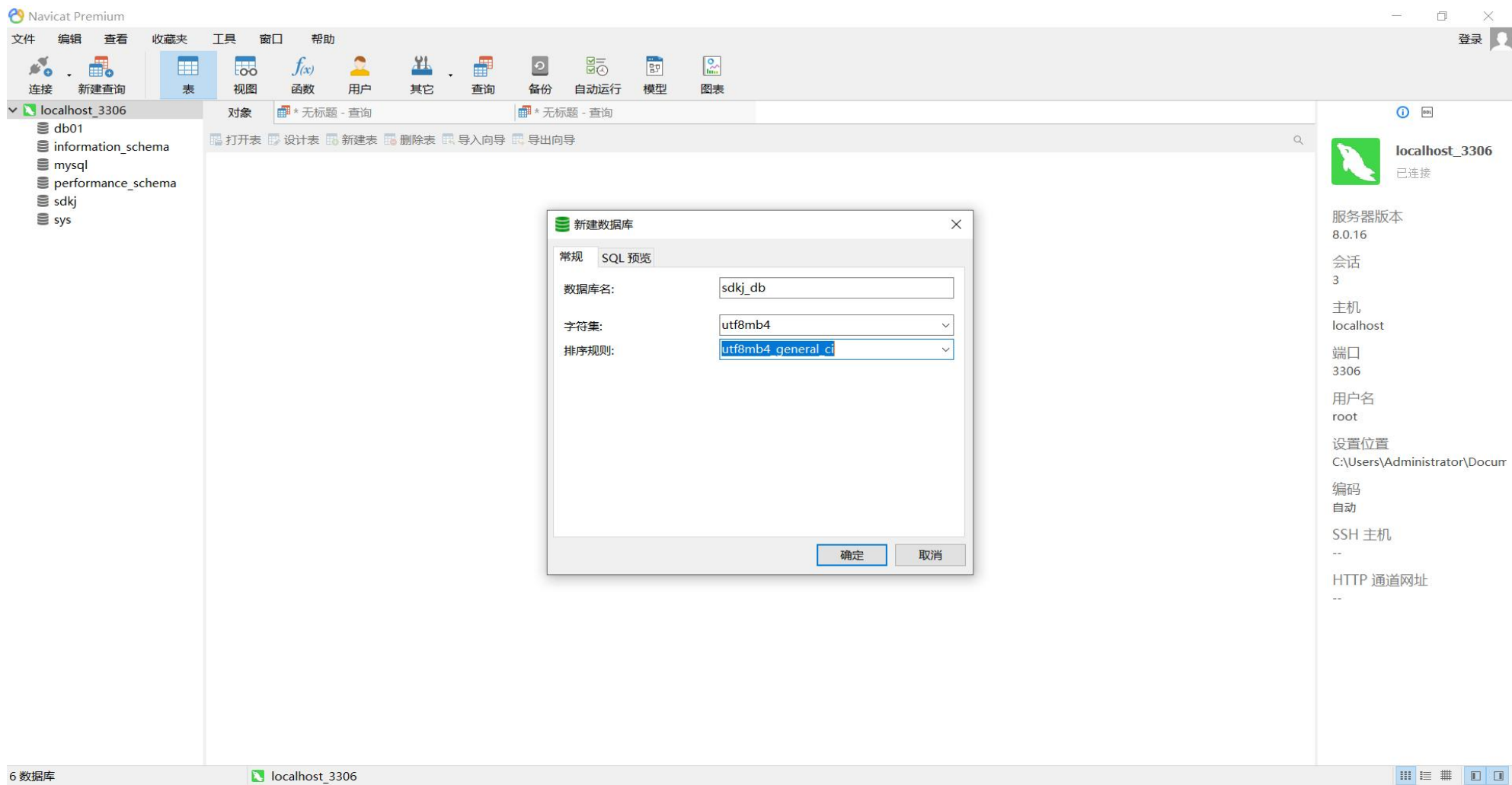
# Django实例

## 登录测试



# Django实例

## 创建数据库







# Django实例

激活容器安装mysql驱动

```
(sdkj) D:\>conda activate kjxy
(kjxy) D:\stuwork\python\web\ehr01>pip install pymysql
Collecting pymysql
  Using cached PyMySQL-1.0.2-py3-none-any.whl (43 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.0.2
```

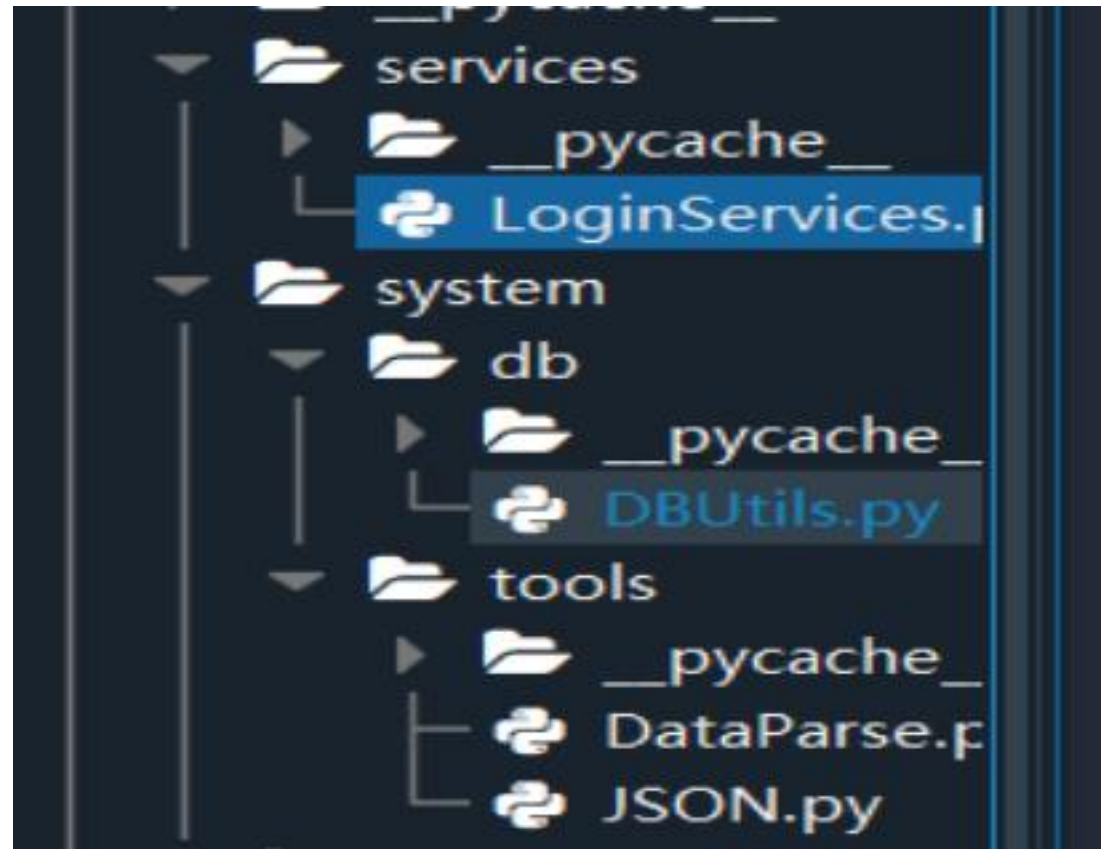
# Django实例

打开spyder找到setting.py, 添加本地数据库

```
81 # }
82
83 DATABASES = {
84     ... 'default': {
85         ... 'ENGINE': 'django.db.backends.mysql', # 默认数据库
86         ... 'NAME': 'sdkj_db', # 连接的数据库
87         ... 'HOST': '127.0.0.1', # mysql的ip地址
88         ... 'PORT': 3306, # mysql的端口
89         ... 'USER': 'root', # mysql的用户名
90         ... 'PASSWORD': '', # mysql的密码
91     ... }
92 }
93
```

# Django实例

在ehr01下创建两个文件夹:services和system,  
在services下创建一个python文件叫做LoginServices,  
在system下创建名为db和tools的文件夹, 在群里下载DBUtils.py,  
DataParse.py, JSON.py;并把DBUtils.py复制到db中,  
把DataParse.py和JSON.py复制到tools中



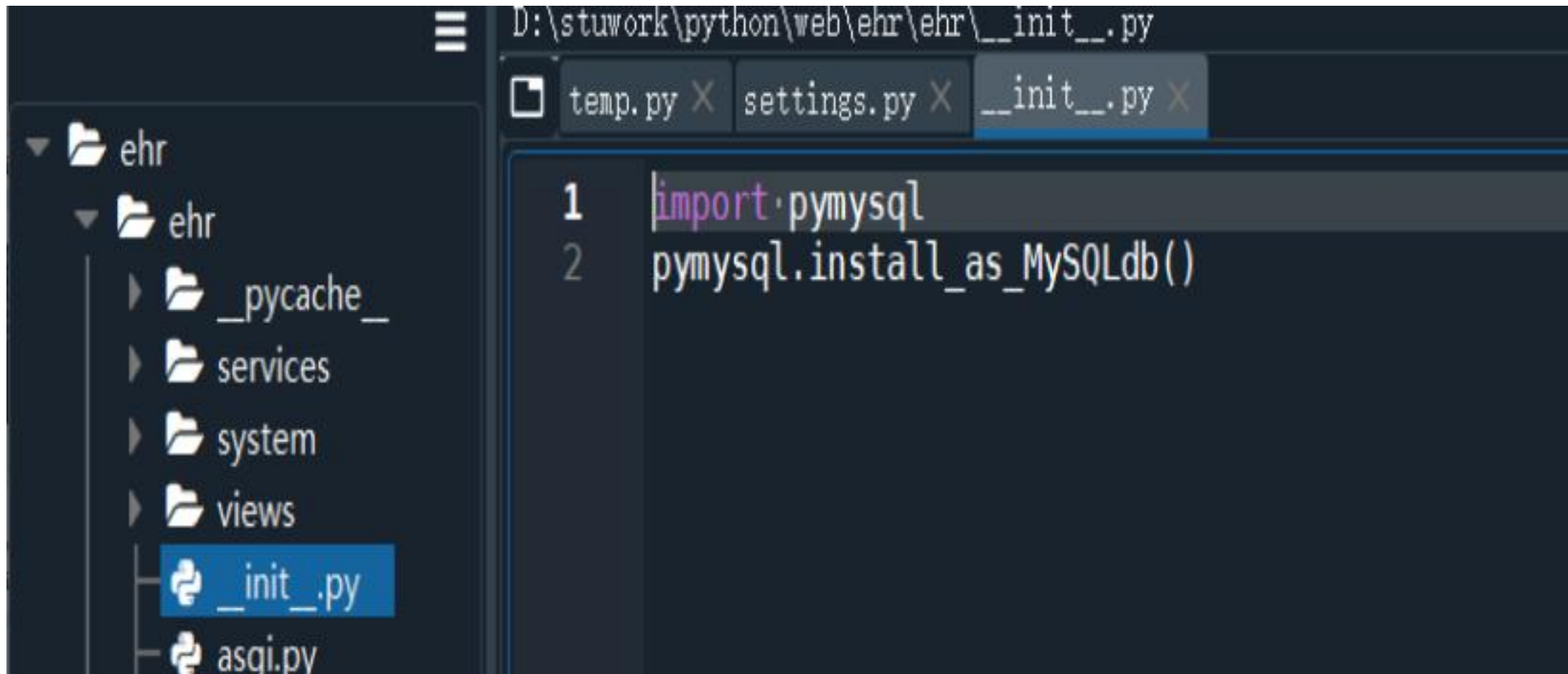
# Django实例

在LoginServices.py, 添加如下代码

```
5 @author: Administrator
6 """
7
8 #导入django的模块
9 from django.db import connection as conn
10 #导入DBUtils
11 from ..system.db import DBUtils
12
13 def checkUser(loginName:str, pwd:str):
14     """
15     .. 校验用户名和密码是否正确
16     .. """
17     .. #定义SQL语句
18     .. sql="""
19     .. select x.loginname, x.username
20     .. from sysuser x
21     .. where x.state=%s
22     .. and x.pwd=%s
23     .. and x.loginname=%s
24     .. """
25     .. params=[
26     ..     '1',
27     ..     pwd,
28     ..     loginName
29     .. ]
30     .. with conn.cursor() as cursor:
31     ..     cursor.execute(sql, params)
32     .. return DBUtils.dict_fetchone(cursor)
```

# Django 实例

找到 `__init__.py`, 添加如下代码



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the following structure:

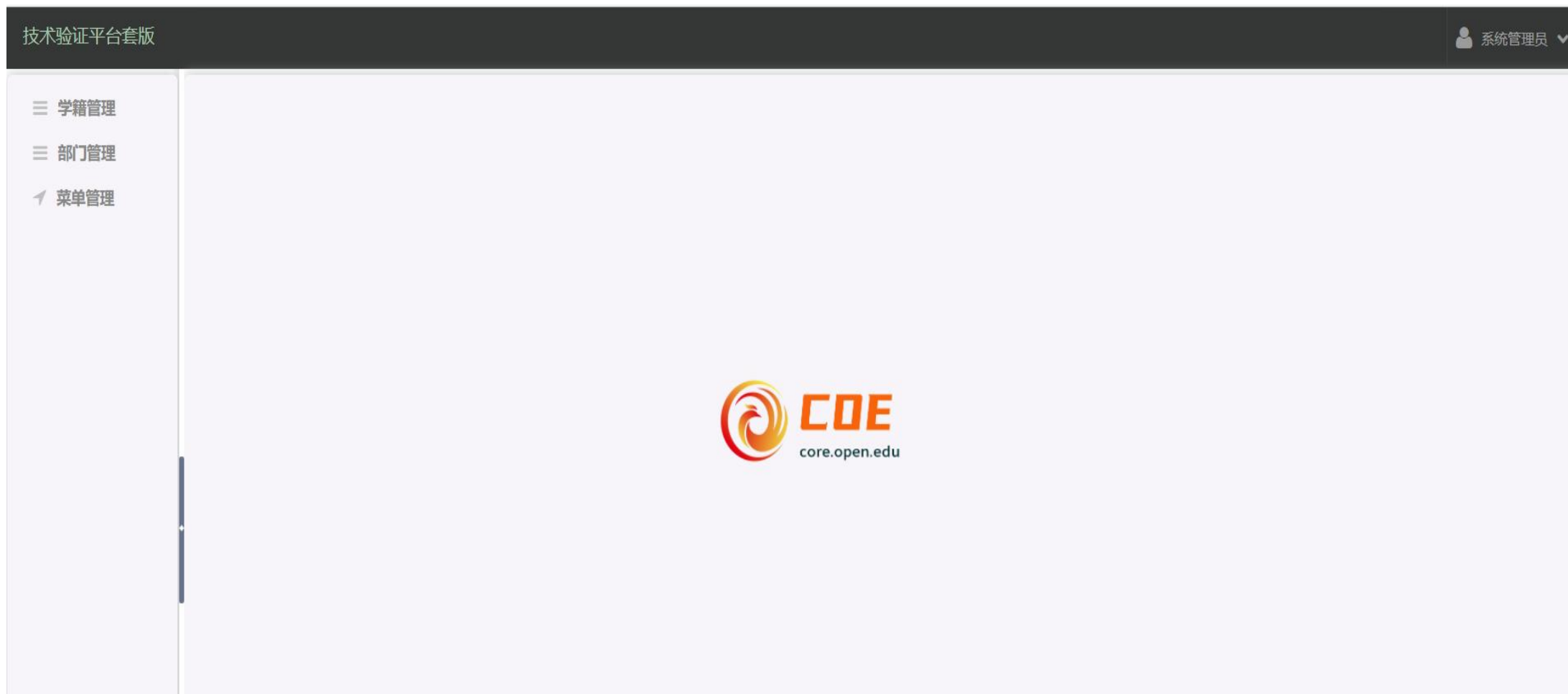
- ehr
  - ehr
    - \_\_pycache\_\_
    - services
    - system
    - views
    - \_\_init\_\_.py**
    - asqi.py

The code editor shows the following code in `__init__.py`:

```
1 import pymysql
2 pymysql.install_as_MySQLdb()
```

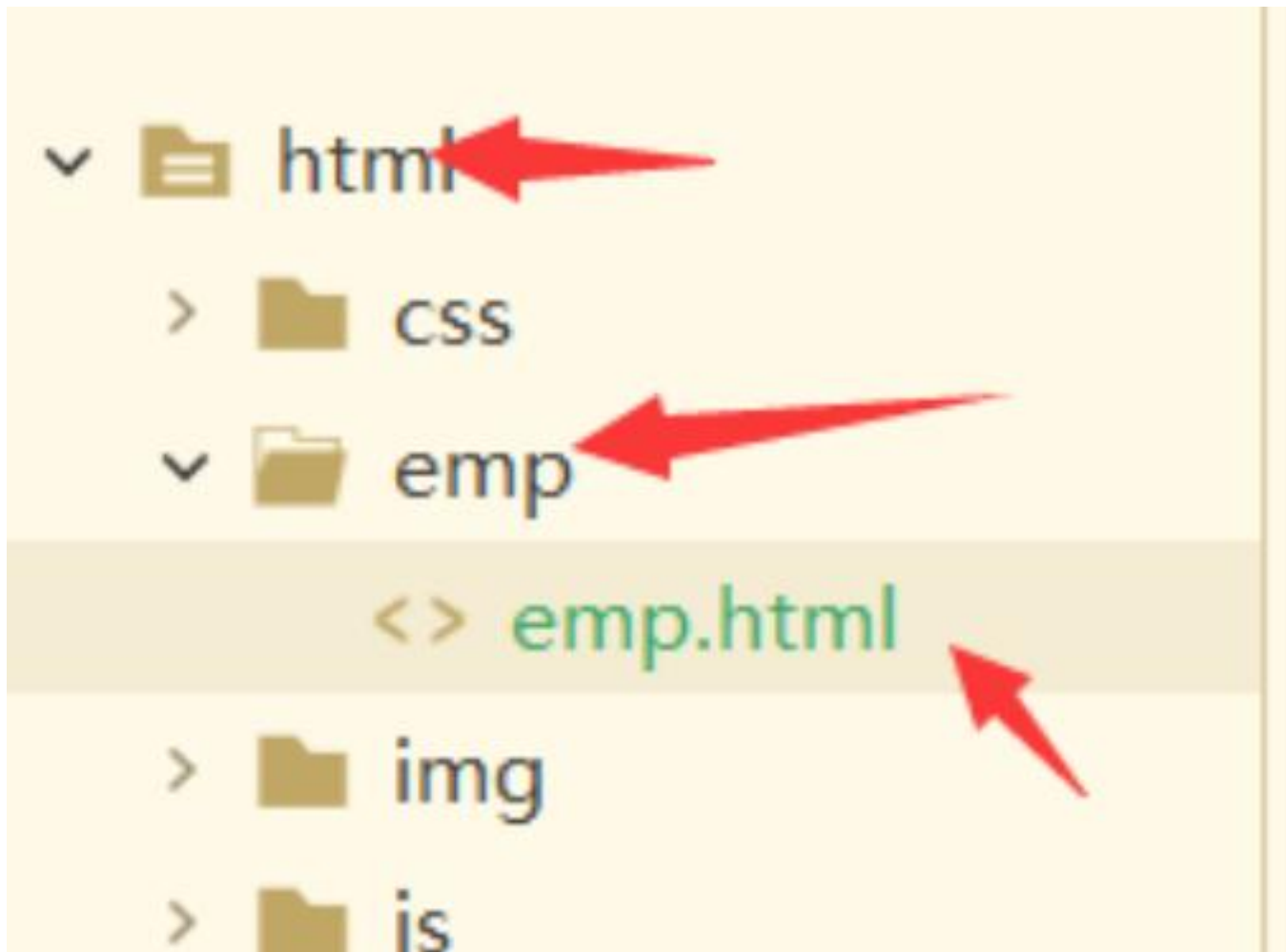
# Django实例

## 测试结果



# Django实例

进入Hbulider x在html下创建名为emp  
文件夹及其子文件emp.html



# Django实例

编写emp.html代码如下,实现查询,添加,删除等功能

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>A1010(部门管理)</title>
6 <style type="text/css">@IMPORT url("../css/style.css");</style>
7 <style>@import url("../css/loading/loading.css"); </style>
8 <script src="../js/jquery-3.4.1.min.js"></script>
9 <script src="../js/ui.js"></script>
10 </head>
11 <body>
12 <div id="qmsg" class="msg">&nbsp;</div>
13 <br>
14 <br>
15 <form id="mainform">
16 <!-- 查询条件 -->
17 <div class="query">
18 <table>
19 <caption>
20 员工管理
21 <hr>
22 </caption>
23 <tr>
24 <td colspan="8">查询条件</td>
25 </tr>
26 <tr>
27 <td>员工编号:</td>
28 <td>
29 <input type="text" id="eno" name="eno">
30 </td>
31 <td>姓名</td>
32 <td>
33 <input type="text" id="ename" name="ename">
34 </td>
35 <td>性别</td>
36 <td>
37 <input type="radio" name="sex" value="男"/>男
38 <input type="radio" name="sex" value="女"/>女
39 <input type="radio" name="sex" checked value="男"/>==不限==
40 </td>
41 <td>民族</td>
42 <td>
43 <select name="nation" id="nation">
44 <option value="">==不限==</option>
45 <option value="汉族">汉族</option>
46 <option value="满族">满族</option>
47 <option value="蒙古族">蒙古族</option>
48 <option value="藏族">藏族</option>
49 <option value="维族">维族</option>
50 </td>
51 </tr>
```



# Django实例

```
52 |         </table>
53 |     </div>
54 |     <!-- 查询结果-->
55 |     <div class="data">
56 |         <table id="dataTable"></table>
57 |         <script>
58 |             let title=["","序号","编号","姓名","性别","生日","民族","邮件","电话","职务",""];
59 |             addEmptyRow("dataTable",title,15);
60 |         </script>
61 |     </div>
62 |     <div class="pageinfo">
63 |         <table>
64 |             <tr>
65 |                 <td>
66 |                     <select name="pageCtrl" id="pageCtrl"></select>
67 |                 </td>
68 |             </tr>
69 |         </table>
70 |     </div>
71 |     <!-- 功能按钮-->
72 |     <div class="button">
73 |         <table>
74 |             <tr>
75 |                 <td>
76 |                     <input type="button" name="next" id="next_query" value="查询" />
77 |                     <input type="button" name="next" id="next_add" value="添加" />
78 |                 </td>
79 |             </tr>
80 |         </table>
81 |     </div>
82 | </form>
83 | <!-- 模态框弹窗-->
84 | <div id="modal">
85 |     <!-- 弹窗内容 -->
86 |     <div id="modal-content">
87 |         <span onclick="closeModal()" class="close">[关闭&times;]</span>
88 |         <!-- 消息提示 -->
89 |         <div id="msg" class="msg"></div>
90 |
91 |         <!-- 在此处嵌入页面代码-->
92 |         <form id="editform">
93 |             <div class="edit">
94 |                 <table>
95 |                     <caption>
96 |                         添加员工
97 |                     <hr>
98 |                     </caption>
99 |                     <tr>
100 |                         <td colspan="2">员工信息</td>
101 |                     </tr>
102 |                     <tr>
```

# Django实例

```
103 <td>员工编号</td>
104 <td>
105     <input type="text" name="eno" />
106 </td>
107 </tr>
108 <tr>
109     <td>姓名</td>
110 <td>
111     <input type="text" name="ename" />
112 </td>
113 </tr>
114 <tr>
115     <td>性别</td>
116 <td>
117     <input type="radio" name="sex" value="男" checked= />男
118     <input type="radio" name="sex" value="女" />女
119 </td>
120 </tr>
121 <tr>
122     <td>生日</td>
123 <td>
124     <input type="date" name="birthday" />
125 </td>
126 </tr>
127 <tr>
128     <td>民族</td>
129 <td>
130     <select name="nation">
131         <option value="汉族">汉族</option>
132         <option value="满族">满族</option>
133         <option value="蒙古族">蒙古族</option>
134         <option value="藏族">藏族</option>
135         <option value="维族">维族</option>
136     </select>
137 </td>
138 </tr>
139 <tr>
140     <td>邮件</td>
141 <td>
142     <input type="email" name="email" />
143 </td>
144 </tr>
145 <tr>
146     <td>电话</td>
147 <td>
148     <input type="text" name="phone" />
149 </td>
150 </tr>
151 <tr>
152     <td>入职日期</td>
153 <td>
```

# Django实例

```
154         <input type="date" name="hiredate" />
155     </td>
156 </tr>
157 <tr>
158     <td>职务</td>
159     <td>
160         <input type="text" name="job" />
161     </td>
162 </tr>
163 <tr>
164     <td>薪资</td>
165     <td>
166         <input type="number" step="0.01" name="sal" />
167     </td>
168 </tr>
169 <tr>
170     <td>备注</td>
171     <td>
172         <textarea rows="5" cols="55" name="memo"></textarea>
173     </td>
174 </tr>
175 <tr>
176     <td colspan="2">
177         <input type="button" name="next" id="next_save" value="保存" />
178     </td>
179 </tr>
180 </table>
181 </div>
182 </form>
183 <!-- 在以上嵌入页面代码-->
184
185 </div>
186 </div>
187 <script src="../../css/loading/loading.js"></script>
188 <!-- 此处插入js-->
189 <script>
190     <!-- 为添加按钮,绑定js代码,当单击next_add的时候,将执行以下代码-->
191     $("#next_add").click(function(){
192         showModal();
193     });
194
195     //分页查询即数据解析
196     let queryForPage=function(){
197         //清空当前页面表格的数据
198         clearTable("dataTable");
199         $("#pageCtrl").css('display','none'); //隐藏分页器
200
201         $.post(
202             '/queryEmp/', //提交的目标地址
203             $("#mainform").serialize(),
204             function(result){
```

# Django实例

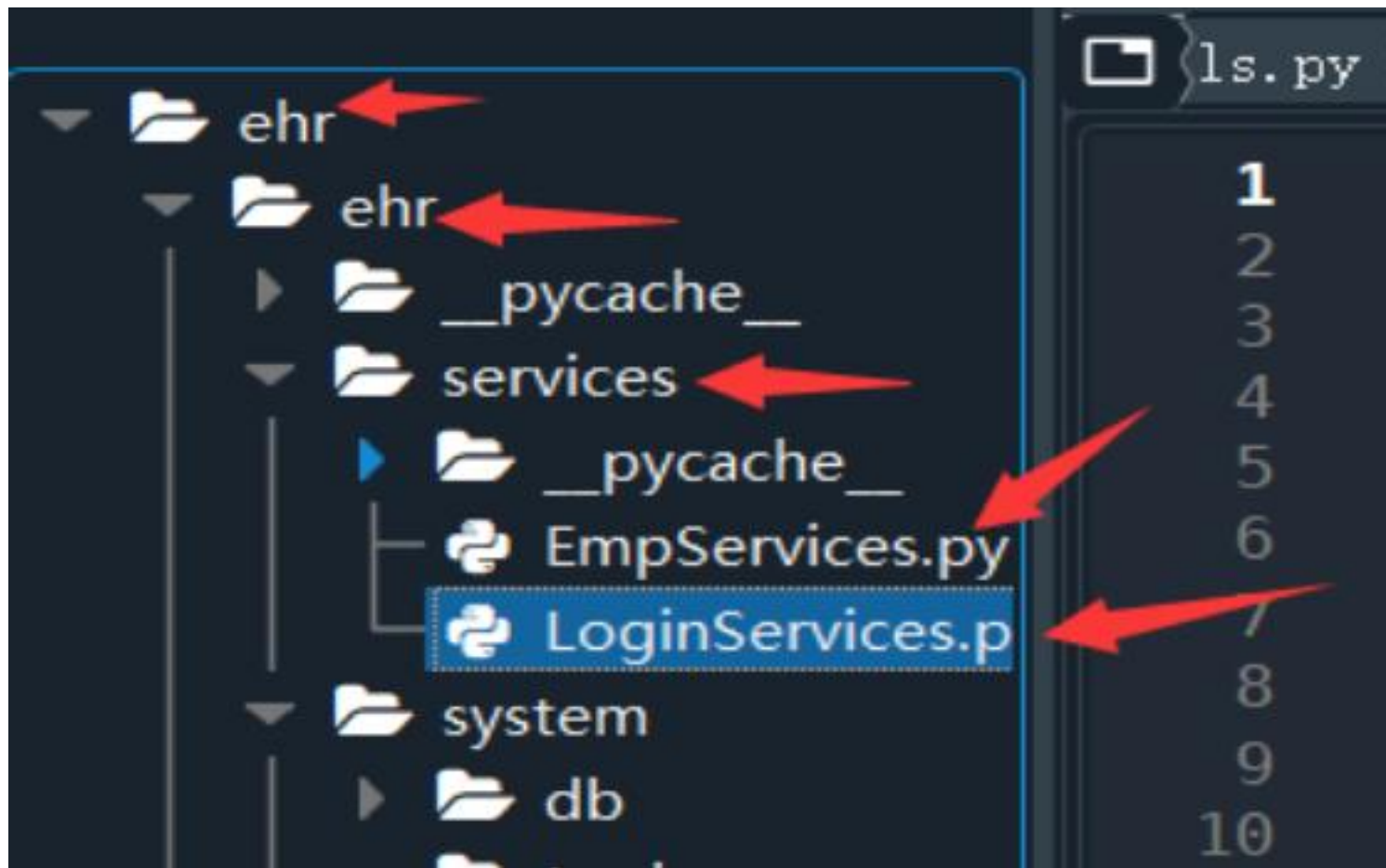
```
205 //alert(result) //将返回的字符串,解析成js的json对象
206 data=JSON.parse(result)
207 pageData=data.pageData; //读取当前页的数据
208 let rowSize=pageData.length; //计算查询到的当前页的行数
209
210 if(rowSize==0){ //空页拦截
211 //消息提示
212 $("#qmsg").html("没有符合条件的数据!");
213 setTimeout(function(){
214     $("#qmsg").html("");
215 },3000);
216 return;
217 }
218
219 $("#pageCtrl").html(data.pageCtrl); //读取分页器,填充下拉列表
220 $("#pageCtrl").css('display','block'); //显示分页器
221
222 //alert("pageData")
223 //便利id为dataTable表格,操作每一行
224 $("#dataTable tr").each(function(index){ //index指的是每行的索引
225     if(index>0 && index<=rowSize){ //实际操作的index起始值是1
226 //回去当前行数据
227 let ins=pageData[index-1];
228 //将index所标识的当前行,转换为jquery对象,以方便jquery后继操作
229 let currRow=$(this); //表格的当前行
230 //将数据放入当前行
231 currRow.find("td:eq(1)").html(index);
232 //对字典对象按名取值
233 currRow.find("td:eq(2)").html(ins.eno);
234 currRow.find("td:eq(3)").html(ins.ename);
235 currRow.find("td:eq(4)").html(ins.sex);
236 currRow.find("td:eq(5)").html(ins.birthday);
237 currRow.find("td:eq(6)").html(ins.nation);
238 currRow.find("td:eq(7)").html(ins.email);
239 currRow.find("td:eq(8)").html(ins.phone);
240 currRow.find("td:eq(9)").html(ins.job);
241 //拼接最后一列(编辑列)
242 let editCell='<a class="cellLabel" onclick="onDel(\''+ins.eid+'\')">[删除]</a>';
243 currRow.find("td:eq(10)").html(editCell);
244     }
245 });
246
247 };
248
249
250 let onDel=function(eid){
251 //alert(eid);
252 $.post(
253     '/deleteById/', //向目标地址提交请求
254     {'eid':eid}, //该请求发送的数据,数据格式为json
255     function(result){ //解析返回的函数---回调接口
```

# Django实例

```
255 |         function(result){ // 解析返回的函数---回调接口
256 |             data=JSON.parse(result);
257 |             $("#qmsg").html(data.msg);
258 |             setTimeout(function(){
259 |                 $("#qmsg").html("&nbsp;");
260 |             },1500)
261 |             queryForPage();
262 |         }
263 |     );
264 |
265 |
266 |     //查询按钮的单击事件
267 |     $("#next_query").click(queryForPage);
268 |     //为分页器绑定内容改变事件
269 |     $("#pageCtrl").change(queryForPage);
270 |
271 |     //-----分割线以下为模态框脚本-----
272 |
273 |     $("#next_save").click(function(){
274 |         load.show(); //弹出等待框
275 |         //调用jquery代码,将也i按数据想python代码提交
276 |         $.post(
277 |             '/addEmp/', //提交的目标地址
278 |             $("#editform").serialize(), //将页面数据项目标地址提交
279 |             function(result){ //该函数表示接收到了服务端的返回消息
280 |                 load.hide();
281 |                 //将服务器端返回的字符串,转换成js的json对象
282 |                 data=JSON.parse(result);
283 |                 //alert(data.msg);
284 |                 $("#msg").html(data.msg);
285 |                 setTimeout(function(){ //三秒清空
286 |                     $("#msg").html("");
287 |                 },3000);
288 |             }
289 |         );
290 |     });
291 | </script>
292 | </body>
293 | </html>
```

# Django实例

使用spyder更新python代码如下(之前ppt中有的代码也在以下内容中)  
新建services文件创建EmpServices.py和LoginServices.py的子文件



# Django实例

EmpServices.py文件代码如下

```
def deleteEmpById(dto:dict):  
    """EmpService.deleteEmpById 按主键删除数据"""  
    sql="delete from emp where eid=%s"  
    tag=0  
    try:  
        with conn.cursor() as cursor:  
            cursor.execute(sql,[dto.get('eid')])  
            tag=1  
    except Exception as ex:  
        print(ex)  
    return tag
```

#导入django的数据库模块

```
from django.db import connection as conn  
from ..system.db import DBUtils
```

# Django实例

```
28 def query(dto:dict):
29     """EmpServices数据分页查询"""
30     sql=""
31     select x.eid,x.eno,x.ename,x.sex,
32     date_format(x.birthday,'%Y-%m-%d')-birthday,
33     x.nation,x.email,x.job,x.phone
34     from emp x
35     where l=1
36     """
37     #定义参数列表
38     params=[]
39     #拼接条件
40     #1.从dto获取所有数据--将页面查询条件逐一获取
41     eno=dto.get('eno') #编号
42     ename=dto.get('ename') #姓名
43     sex=dto.get('sex') #性别
44     nation=dto.get('nation') #民族
45     #2.逐一判断条件是否输入,如果输入,就拼接一个and
46     if eno:
47         sql+=" and x.eno=%s"
48         params.append(eno)
49     if ename:
50         sql+=" and x.ename like %s"
51         params.append("%"+ename+"%")
52     if sex:
53         sql+=" and x.sex=%s"
54         params.append(sex)
55     if nation:
56         sql+=" and x.nation=%s"
57         params.append(nation)
58     print(sql)
59     print(params)
60     #def queryForPage(sql:str,args:list,pageNum:str,pageSize=10):
61     return DBUtils.queryForPage(sql,params,-,dto.get("pageCtrl"),15)
62
63
64
65
66
67
68
69
70
71
```



# Django实例

```
def newEmp(dto:dict):
    """
    .. 添加员工
    """
    #定义SQL语句
    sql="""
    insert into emp(eno,ename,sex,birthday,nation,
    → → → → → email,phone,hiredate,job,sal,
    → → → → → memo)
    → → → → → values(%s,%s,%s,%s,%s,
    → → → → → %s,%s,%s,%s,%s,
    → → → → → %s)
    """
    #编写参数数组
    params=[
    .. dto.get('eno'),
    .. dto.get('ename'),
    .. dto.get('sex'),
    .. dto.get('birthday'),...#row1
    .. dto.get('nation'),
    ..
    .. dto.get('email'),
    .. dto.get('phone'),
    .. dto.get('hiredate'),
    .. dto.get('job'),
    .. dto.get('sal'),...#row2
    ..
    .. dto.get('memo')
    .. ]
    #执行sql语句
    tag=False
    try:
    .. with conn.cursor() as cursor:
    ..     cursor.execute(sql,params)
    ..     tag=True
    .. except Exception as ex:
    ..     print(ex)
    ..
    .. return tag
```

# Django实例

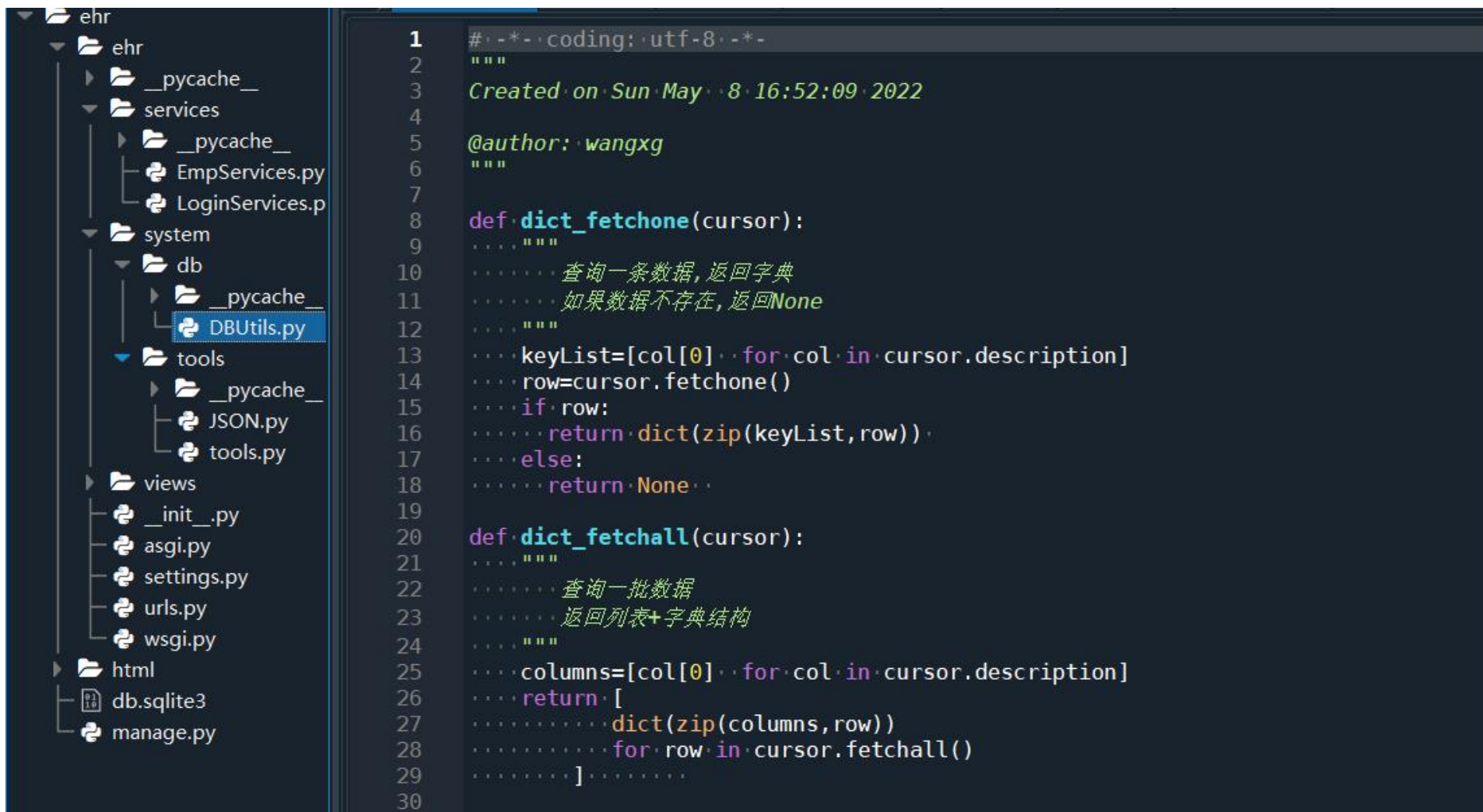
## LoginServices.py文件代码如下

```
#导入django的数据库模块
from django.db import connection as conn
#导入DBUtils
from ..system.db import DBUtils
from ..system.tools import tools

def checkUser(loginName:str,pwd:str):
    """
    .. 校验用户名和密码是否正确
    """
    """
    .. 定义SQL语句
    """
    sql="""
    .. select x.loginname,x.username
    .. from sysuser x
    .. where x.state=%s
    .. and x.pwd=%s
    .. and x.loginname=%s
    """
    params=[
    .. '1',
    .. #对密码进行加密,然后基于密文进行比较
    .. tools.md5code2(pwd),
    .. loginName
    .. ]
    print(sql)
    print(params)
    """
    #连接 数据库执行SQL
    with conn.cursor() as cursor:
    .. #基于游标执行查询
    .. cursor.execute(sql,params)
    .. return DBUtils.dict_fetchone(cursor) ..
```

# Django实例

## 编写DBUtils.py文件代码



The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a Django project structure with a file named `DBUtils.py` selected under the `tools` directory. The code editor displays the following Python code:

```
1  #-*-coding:utf-8-*-
2  """
3  Created on Sun May 8 16:52:09 2022
4
5  @author: wangxg
6  """
7
8  def dict_fetchone(cursor):
9      """
10     ... 查询一条数据, 返回字典
11     ... 如果数据不存在, 返回None
12     """
13     ... keyList=[col[0] for col in cursor.description]
14     ... row=cursor.fetchone()
15     ... if row:
16     ...     return dict(zip(keyList, row))
17     ... else:
18     ...     return None
19
20  def dict_fetchall(cursor):
21     """
22     ... 查询一批数据
23     ... 返回列表+字典结构
24     """
25     ... columns=[col[0] for col in cursor.description]
26     ... return [
27     ...     dict(zip(columns, row))
28     ...     for row in cursor.fetchall()
29     ... ]
30
```

# Django实例

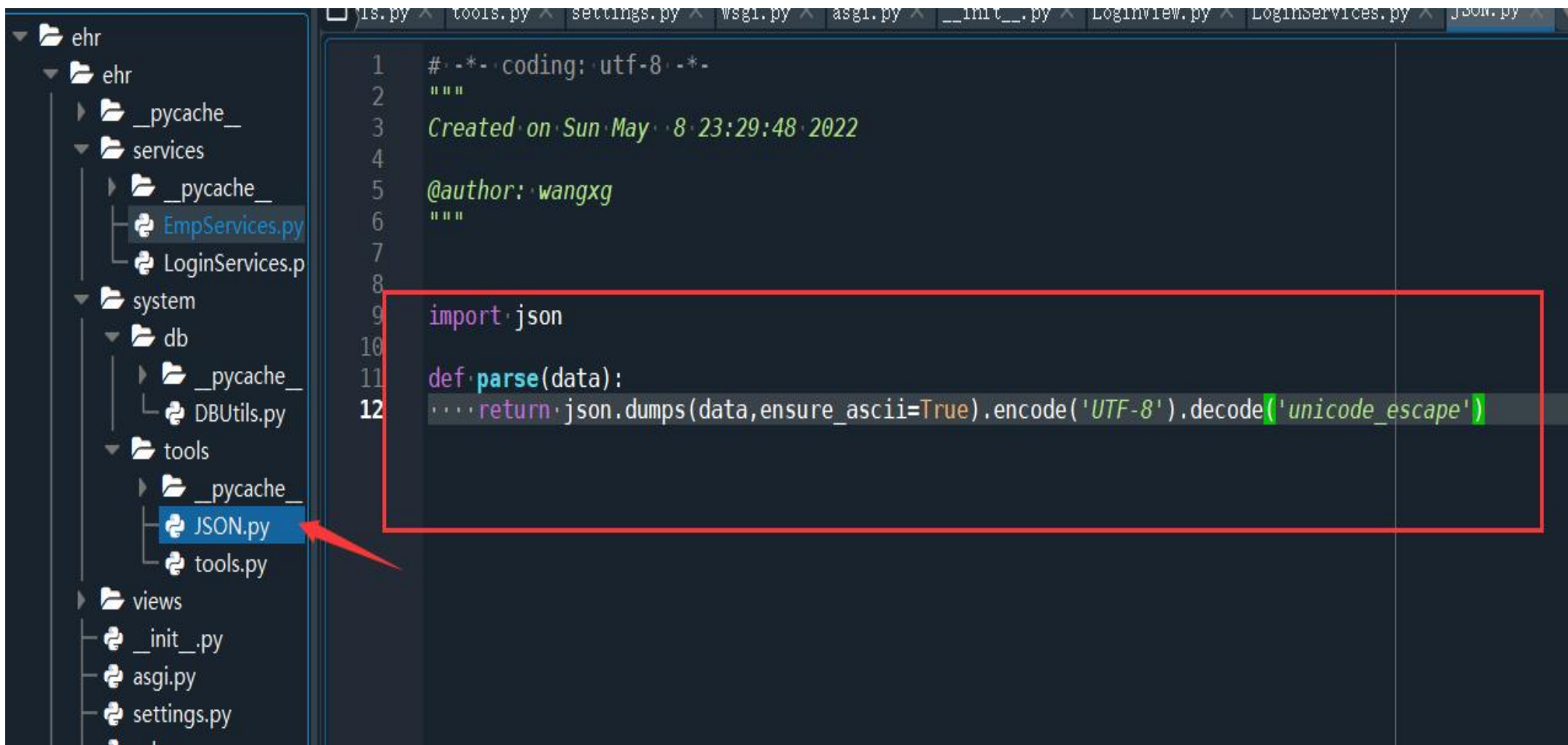
```
31
32 from django.db import connection as conn
33 # from dweb.system.tools import JSON
34 from ..tools import JSON
35 ....
36 def queryForPage(sql:str,args:list,pageNum:str,pageSize=10):
37     .... """
38     .... 数据分页查询
39     .... 返回json对象:
40     .... result is {
41     ....     'pageCtrl': '分页器字符串',
42     ....     'pageData': '查询结果list[dict{str:str}]'
43     .... } ....
44     .... """
45     ....
46     .... #截取SQL
47     .... fromIndex=sql.find('from')
48     .... orderByIndex=sql.find('order')
49     .... orderByIndex=orderByIndex if orderByIndex>=0 else len(sql)
50     .... rowCountSql="select count(1) "+sql[fromIndex:orderByIndex]
51     ....
52     .... # print(f"args is {args}, rowCountSql is {rowCountSql}")
53     .... #计算总行数
54     .... with conn.cursor() as cursor:
55     ....     cursor.execute(rowCountSql,args)
56     ....     rowSize=cursor.fetchone()[0]
57     ....
58     .... #如果没有符合条件的数据,返回空列表
59     .... if rowSize==0:
60     ....     reuslt={
61     ....         'pageCtrl':'',
62     ....         'pageData':list()
63     ....     }
64     ....     return JSON.parse(reuslt)
```

# Django 实例

```
.....
.....#计算总页码数.....
.....pages=(rowSize+pageSize-1)//pageSize..
.....#清洗用户请求的页码数:
.....currentPageNo=1.....#默认可以获取的页码数
.....if pageNum:.....#用户指定了页码数
.....    currentPageNo=int(pageNum).....#以客户指定的页码为当前页码
.....if currentPageNo> pages:.....#如果用户请求的页码大于总的页码,返回最后一页
.....    currentPageNo=pages
.....
.....#拼接分页器
.....pageCtrl=''
.....for pageNo in range(1,pages+1):
.....    if pageNo==currentPageNo:
.....        pageCtrl+=f"<option value={pageNo} selected>&nbsp;第[{pageNo}]页&nbsp;</option>"
.....    else:
.....        pageCtrl+=f"<option value={pageNo}>&nbsp;第[{pageNo}]页&nbsp;</option><br>"..
.....
.....#计算当前页的起始行数
.....startRow=(currentPageNo-1)*pageSize
.....
.....#重构查询语句
.....limitSql=sql+f".limit.{startRow},{pageSize}."
.....print(f"limitSql is {limitSql}")
.....#执行重构的sql语句
.....with conn.cursor() as cursor:
.....    cursor.execute(limitSql,args)
.....    rows=dict_fetchall(cursor)
.....
.....#构建返回值字典
.....result={
.....    'pageCtrl':pageCtrl,
.....    'pageData':rows
.....}.....
.....#返回json对象
.....return JSON.parse(result)
```

# Django实例

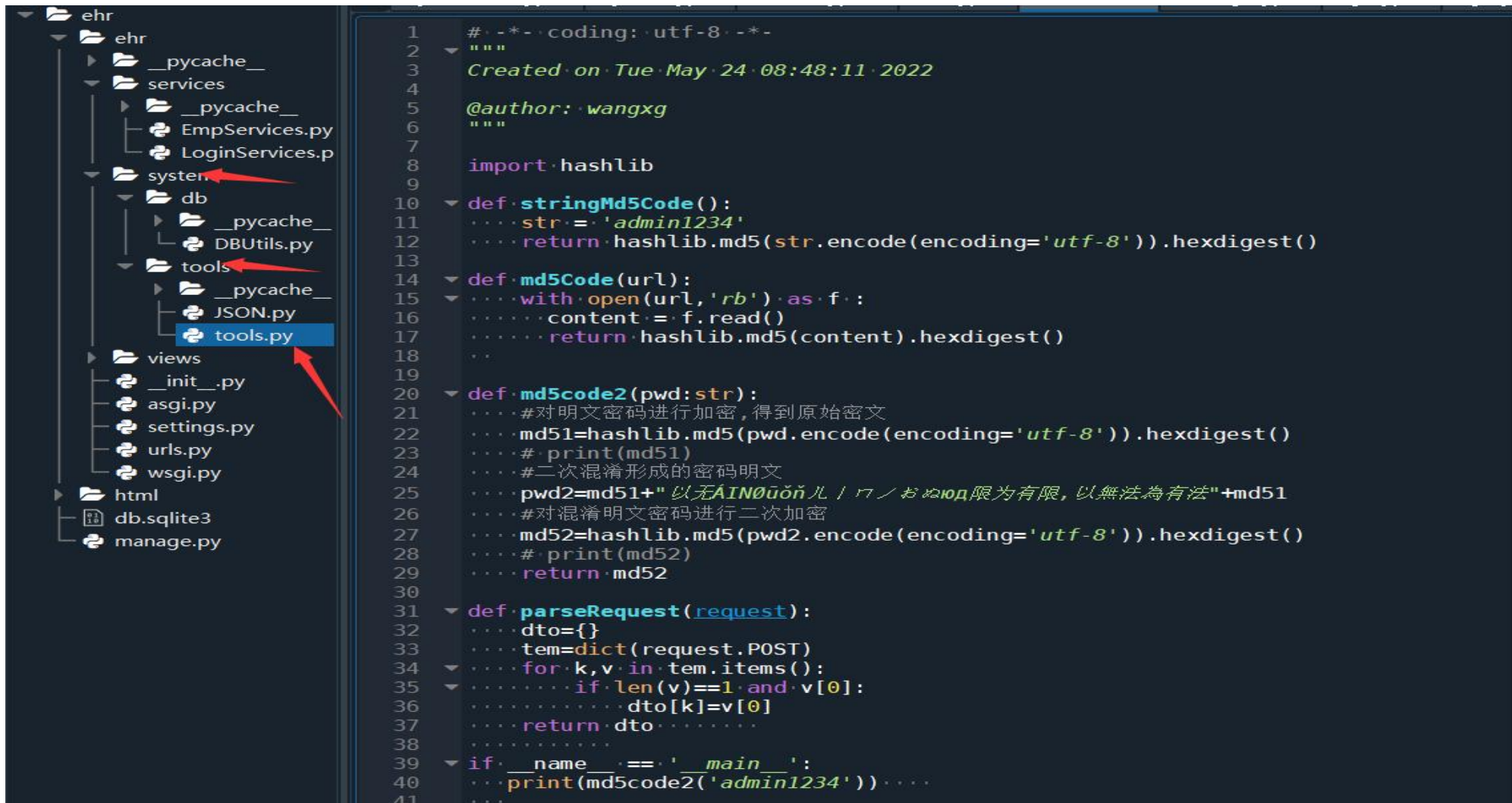
## 编写JSON文件



```
1  #-*- coding: utf-8 -*-
2  """
3  Created on Sun May  8 23:29:48 2022
4
5  @author: wangxg
6  """
7
8
9  import json
10
11  def parse(data):
12  ... return json.dumps(data,ensure_ascii=True).encode('UTF-8').decode('unicode_escape')
```

# Django实例

## 编写tools.py文件

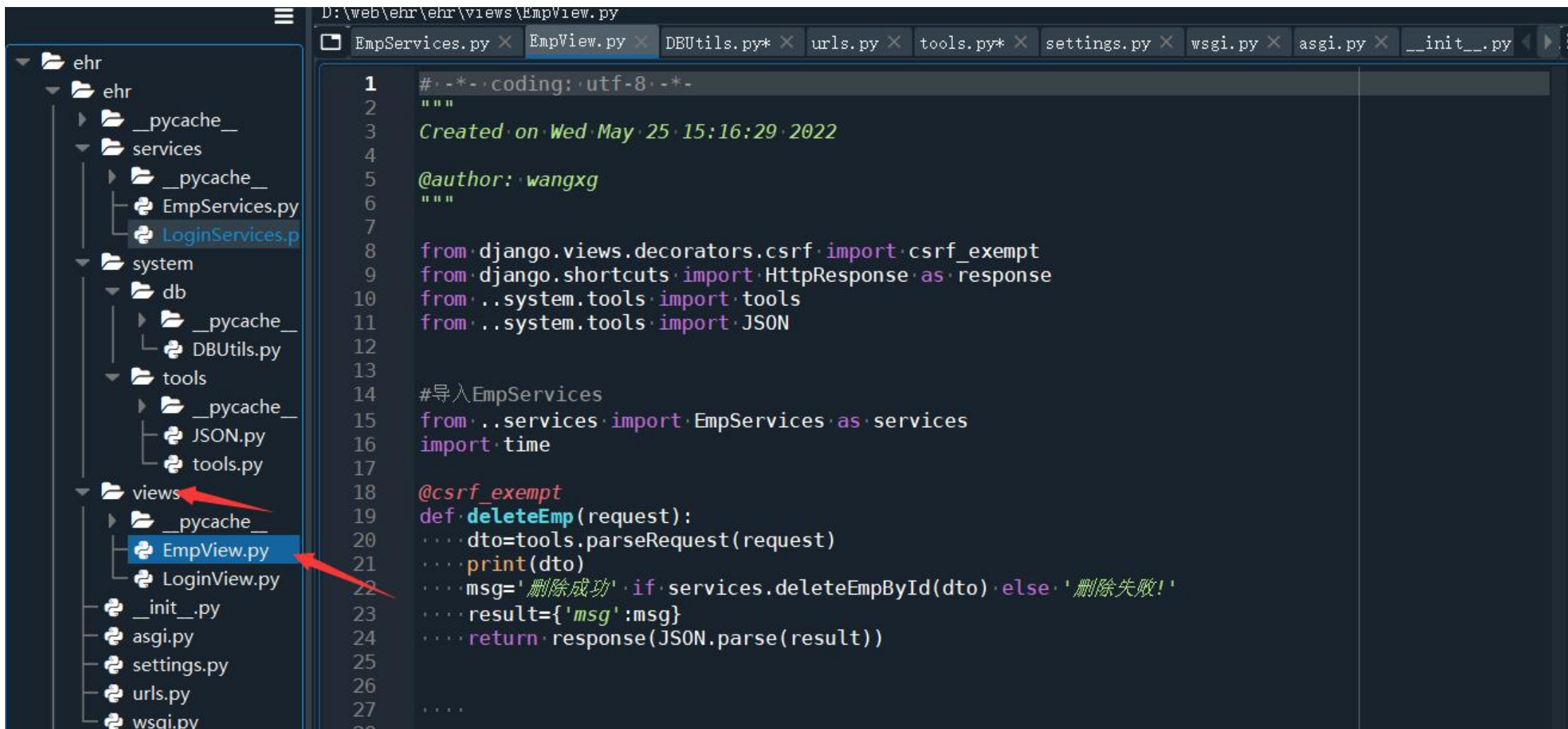


The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'tools' folder containing 'tools.py'. The code editor shows the content of 'tools.py'.

```
1  #-*-coding:utf-8-*-
2  """
3  Created on Tue May 24 08:48:11 2022
4
5  @author: wangxg
6  """
7
8  import hashlib
9
10 def stringMd5Code():
11     str='admin1234'
12     return hashlib.md5(str.encode(encoding='utf-8')).hexdigest()
13
14 def md5Code(url):
15     with open(url,'rb') as f:
16         content=f.read()
17         return hashlib.md5(content).hexdigest()
18
19
20 def md5code2(pwd:str):
21     #对明文密码进行加密,得到原始密文
22     md51=hashlib.md5(pwd.encode(encoding='utf-8')).hexdigest()
23     # print(md51)
24     #二次混淆形成的密码明文
25     pwd2=md51+"以无AIN0ūōñ儿 / 7 / 书ぬюд限为有限,以無法為有法"+md51
26     #对混淆明文密码进行二次加密
27     md52=hashlib.md5(pwd2.encode(encoding='utf-8')).hexdigest()
28     # print(md52)
29     return md52
30
31 def parseRequest(request):
32     dto={}
33     tem=dict(request.POST)
34     for k,v in tem.items():
35         if len(v)==1 and v[0]:
36             dto[k]=v[0]
37     return dto
38
39 if __name__=='__main__':
40     print(md5code2('admin1234'))
41
```

# Django实例

## 编写EmpView.py文件



```
1  #-*-coding:utf-8-*-
2  """
3  Created on Wed May 25 15:16:29 2022
4
5  @author: wangxg
6  """
7
8  from django.views.decorators.csrf import csrf_exempt
9  from django.shortcuts import HttpResponseRedirect as response
10 from ..system.tools import tools
11 from ..system.tools import JSON
12
13 #导入EmpServices
14 from ..services import EmpServices as services
15 import time
16
17 @csrf_exempt
18 def deleteEmp(request):
19     ...dto=tools.parseRequest(request)
20     ...print(dto)
21     ...msg='删除成功' if services.deleteEmpById(dto) else '删除失败!'
22     ...result={'msg':msg}
23     ...return response(JSON.parse(result))
24
25
26
27
28
```

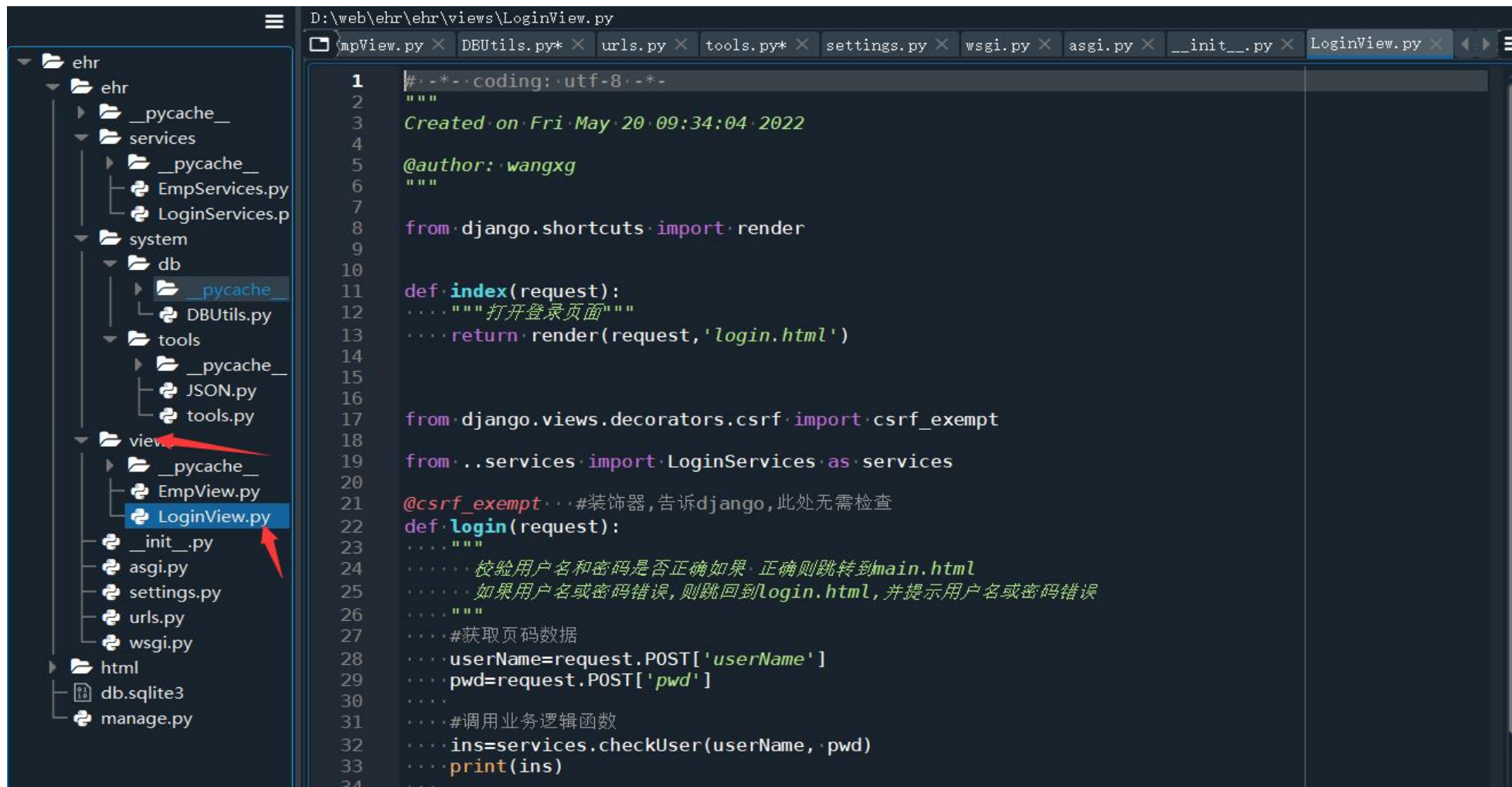


# Django实例

```
27     ....
28     ....
29     @csrf_exempt
30     def query(request):
31         .... """EmpView. 对emp表的数据, 分页查询"""
32         .... #获取dto
33         .... dto=tools.parseRequest(request)
34         .... #获取分页的数据
35         .... result=services.query(dto)
36         .... #向客户端输出查询结果
37         .... return response(result)
38
39
40     @csrf_exempt
41     def addEmp(request):
42         .... """ 添加员工 """
43         .... #调用tools中函数, 获取页面数据
44         .... dto=tools.parseRequest(request)
45         .... print(f"dto is {dto}")
46         ....
47         .... time.sleep(5)
48         ....
49         ....
50         .... #调用函数, 完成数据添加
51         .... msg=" 添加成功!" if services.newEmp(dto) else " 添加失败!"
52         ....
53         ....
54         .... # 定义返回值----字典
55         .... result={'msg':msg}
56         .... # 将字典转换成JSON对象
57         .... # jobj=JSON.parse(result)
58         .... # print(f"jobj is {jobj}")
59         .... # print(type(jobj))
60         ....
61         .... #将转换后的json对象, 返回给客户端的浏览器
62         .... return response(JSON.parse(result))
63
```

# Django实例

## 编写LoginView.py文件



The image shows a code editor window with a file explorer on the left and a code editor on the right. The file explorer shows the project structure, with the 'views' directory expanded and 'LoginView.py' selected. The code editor shows the content of 'LoginView.py'.

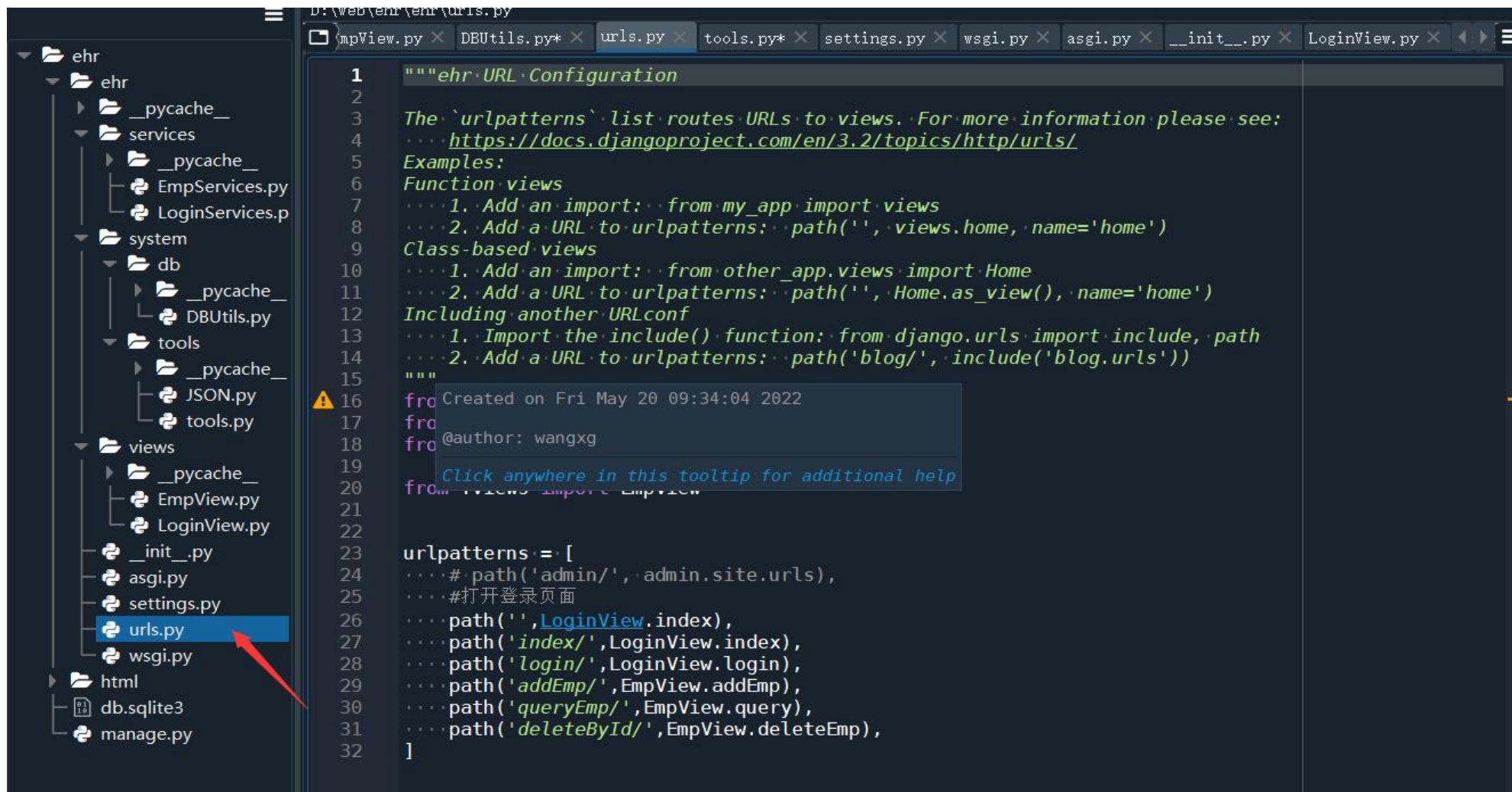
```
1  #-*-coding:utf-8-*-
2  """
3  Created on Fri May 20 09:34:04 2022
4
5  @author: wangxg
6  """
7
8  from django.shortcuts import render
9
10
11 def index(request):
12     """打开登录页面"""
13     return render(request, 'login.html')
14
15
16
17 from django.views.decorators.csrf import csrf_exempt
18
19 from ..services import LoginServices as services
20
21 @csrf_exempt #装饰器,告诉django,此处无需检查
22 def login(request):
23     """
24     校验用户名和密码是否正确如果正确则跳转到main.html
25     如果用户名或密码错误,则跳回到login.html,并提示用户名或密码错误
26     """
27     #获取页码数据
28     userName=request.POST['userName']
29     pwd=request.POST['pwd']
30
31     #调用业务逻辑函数
32     ins=services.checkUser(userName, pwd)
33     print(ins)
34
```

# Django实例

```
35     ....
36     .... #定义变量表示跳转的目标路径
37     .... topage='login.html'
38     .... #定义变量表示为页面提供的数据
39     .... # data={}
40     .... # if userName=='admin' and pwd=='admin1234':
41     .... #     .... topage='main.html'
42     .... #     .... data['username']='系统管理员'
43     .... # else:
44     .... #     .... data['msg']='用户名或密码错误'
45     .... # return render(request, topage, data) ....
46     ....
47     .... if ins:
48     ....     .... topage='main.html'
49     ....     .... return render(request, topage, ins)
50     .... else:
51     ....     .... return render(request, topage, {'msg':'用户名或密码错误'})
52     ....
```

# Django实例

## 最后编写路由文件urls.py




```
1 """ehr URL Configuration
2
3 The `urlpatterns` list routes URLs to views. For more information please see:
4 ... https://docs.djangoproject.com/en/3.2/topics/http/urls/
5 Examples:
6 Function views
7 ... 1. Add an import: from my_app import views
8 ... 2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10 ... 1. Add an import: from other_app.views import Home
11 ... 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13 ... 1. Import the include() function: from django.urls import include, path
14 ... 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.urls import path
17 from django.contrib.auth import views as auth_views
18 from . import views
19
20 urlpatterns = [
21     path('', LoginView.index),
22     path('index/', LoginView.index),
23     path('login/', LoginView.login),
24     path('addEmp/', EmpView.addEmp),
25     path('queryEmp/', EmpView.query),
26     path('deleteById/', EmpView.deleteEmp),
27 ]
```

# Django实例

## 登录测试添加功能

添加员工

员工信息	
员工编号	E1001
姓名	张三
性别	<input checked="" type="radio"/> 男 <input type="radio"/> 女
生日	2022/05/29 <input type="text"/>
民族	维族 <input type="text"/>
邮件	12
电话	1151
入职日期	2022/05/29 <input type="text"/>
职务	程序员
薪资	
备注	<div style="text-align: center;"> Loading... <input type="button" value="保存"/></div>

添加成功!

添加员工

员工信息	
员工编号	E1001
姓名	张三
性别	<input checked="" type="radio"/> 男 <input type="radio"/> 女
生日	2022/05/29 <input type="text"/>
民族	维族 <input type="text"/>
邮件	12
电话	1151
入职日期	2022/05/29 <input type="text"/>
职务	程序员
薪资	
备注	<input type="text"/>

# Django实例

## 测试查询功能

查询条件

员工编号:  姓名  性别  男  女  ==不限== 民族 ==不限==

序号	编号	姓名	性别	生日	民族	邮件	电话	职务	
1	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
2	E1001	刘备	男	2022-05-29	汉族	12	1151	程序员	[删除]
3	E1001	赵三	男	2022-05-29	维族	12	1151	程序员	[删除]
4	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
5	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
6	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
7	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
8	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
9	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
10	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
11	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
12	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
13	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
14	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
15	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]

第[1]页

# Django实例

## 测试删除功能(删除赵三)

删除成功

员工管理

查询条件

员工编号:  姓名:  性别:  男  女  ==不限== 民族: ==不限==

序号	编号	姓名	性别	生日	民族	邮件	电话	职务	
1	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
2	E1001	刘备	男	2022-05-29	汉族	12	1151	程序员	[删除]
3	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
4	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
5	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
6	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
7	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
8	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
9	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
10	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
11	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
12	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
13	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
14	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
15	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]

第[1]页

查询 添加



Linux 安装 Django



# Linux 系统安装 Django 实例

## 安装 mysql

检查是否有mysql以及MariaDB, 如果存在用yum remove命令移除, 安装我们新上传的版本

```
[admin@localhost ~]$ rpm -qa|grep mysql  
[admin@localhost ~]$ rpm -qa|grep mariadb  
[admin@localhost ~]$
```

# Linux系统安装Django实例

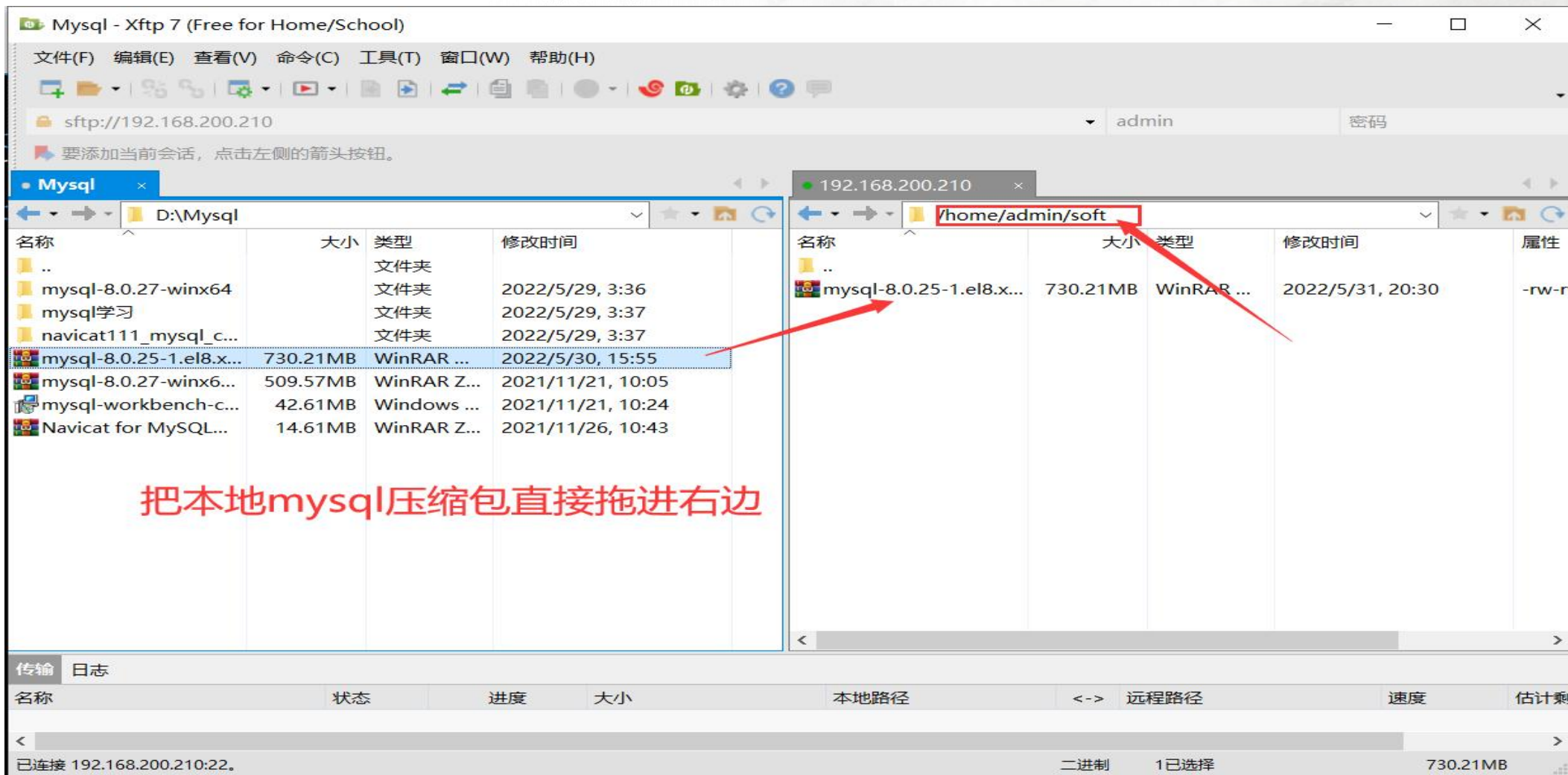
## 安装mysql

打开虚拟机, 连接xshell和xftp, 创建soft目录把mysql压缩包上传到soft目录

```
[admin@localhost ~]$ cd
[admin@localhost ~]$ ls
公共 模板 视频 图片 文档 下载 音乐 桌面
[admin@localhost ~]$ mkdir soft
[admin@localhost ~]$ ls
公共 模板 视频 图片 文档 下载 音乐 桌面 soft
[admin@localhost ~]$ █
```

# Linux系统安装Django实例

## 安装mysql



# Linux系统安装Django实例

## 安装mysql

检查是否上传成功

```
[admin@localhost ~]$ ls
公共 模板 视频 图片 文档 下载 音乐 桌面 soft
[admin@localhost ~]$ cd soft/
[admin@localhost soft]$ ls
mysql-8.0.25-1.el8.x86_64.rpm-bundle.tar
[admin@localhost soft]$
```

# Linux系统安装Django实例

## 安装mysql

使用以下命令,安装依赖包

```
sudo yum install -y openssl-devel.x86_64  
openssl.x86_64
```

```
sudo yum install -y perl.x86_64 perl-devel.x86_64
```

```
sudo yum install -y perl-JSON.noarch
```

```
sudo yum install -y autoconf
```

# Linux系统安装Django实例

## 安装mysql

解压上传上来的mysql压缩包

```
[admin@localhost ~]$ cd soft/  
[admin@localhost soft]$ tar -xvf mysql-8.0.25-1.el8.x86_64.rpm-bundle.tar  
mysql-community-client-8.0.25-1.el8.x86_64.rpm  
mysql-community-client-debuginfo-8.0.25-1.el8.x86_64.rpm  
mysql-community-client-plugins-8.0.25-1.el8.x86_64.rpm  
mysql-community-client-plugins-debuginfo-8.0.25-1.el8.x86_64.rpm  
mysql-community-common-8.0.25-1.el8.x86_64.rpm  
mysql-community-debuginfo-8.0.25-1.el8.x86_64.rpm  
mysql-community-debugsource-8.0.25-1.el8.x86_64.rpm  
mysql-community-devel-8.0.25-1.el8.x86_64.rpm  
mysql-community-libs-8.0.25-1.el8.x86_64.rpm  
mysql-community-libs-debuginfo-8.0.25-1.el8.x86_64.rpm  
mysql-community-server-8.0.25-1.el8.x86_64.rpm  
mysql-community-server-debug-8.0.25-1.el8.x86_64.rpm  
mysql-community-server-debug-debuginfo-8.0.25-1.el8.x86_64.rpm  
mysql-community-server-debuginfo-8.0.25-1.el8.x86_64.rpm  
mysql-community-test-8.0.25-1.el8.x86_64.rpm  
mysql-community-test-debuginfo-8.0.25-1.el8.x86_64.rpm  
[admin@localhost soft]$
```

# Linux系统安装Django实例

## 安装mysql

按顺序安装如下软件包

```
sudo rpm -ivh mysql-community-common-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-client-plugins-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-libs-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-client-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-debuginfo-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-client-debuginfo-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-client-plugins-debuginfo-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-debugsource-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-server-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-server-debug-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-server-debuginfo-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-server-debug-debuginfo-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-libs-debuginfo-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-devel-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-test-8.0.25-1.el8.x86_64.rpm
sudo rpm -ivh mysql-community-test-debuginfo-8.0.25-1.el8.x86_64.rpm
```

# Linux系统安装Django实例

## 安装mysql

新建一个叫mysql的目录，mv 这些rpm到mysql目录下，便于以后文件的管理

```
admin@localhost ~]$ mkdir mysql
admin@localhost ~]$ ls
公共 模板 视频 图片 文档 下载 音乐 桌面 mysql soft
admin@localhost ~]$ cd mysql/
admin@localhost mysql]$ mv /home/admin/soft/* ./
admin@localhost mysql]$ ls
mysql-8.0.25-1.el8.x86_64.rpm-bundle.tar
mysql-community-client-8.0.25-1.el8.x86_64.rpm
mysql-community-client-debuginfo-8.0.25-1.el8.x86_64.rpm
mysql-community-client-plugins-8.0.25-1.el8.x86_64.rpm
mysql-community-client-plugins-debuginfo-8.0.25-1.el8.x86_64.rpm
mysql-community-common-8.0.25-1.el8.x86_64.rpm
mysql-community-debuginfo-8.0.25-1.el8.x86_64.rpm
mysql-community-debugsource-8.0.25-1.el8.x86_64.rpm
mysql-community-devel-8.0.25-1.el8.x86_64.rpm
mysql-community-libs-8.0.25-1.el8.x86_64.rpm
mysql-community-libs-debuginfo-8.0.25-1.el8.x86_64.rpm
mysql-community-server-8.0.25-1.el8.x86_64.rpm
mysql-community-server-debug-8.0.25-1.el8.x86_64.rpm
mysql-community-server-debug-debuginfo-8.0.25-1.el8.x86_64.rpm
mysql-community-server-debuginfo-8.0.25-1.el8.x86_64.rpm
mysql-community-test-8.0.25-1.el8.x86_64.rpm
mysql-community-test-debuginfo-8.0.25-1.el8.x86_64.rpm
admin@localhost mysql]$
```



# Linux系统安装Django实例

## 安装mysql

启动mysql 数据库

```
[admin@localhost mysql]$ sudo systemctl start mysqld
[admin@localhost mysql]$ systemctl status mysqld
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor p
   Active: active (running) since Tue 2022-05-31 09:00:13 EDT; 8s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
   Process: 44830 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, stat
 Main PID: 44910 (mysqld)
    Status: "Server is operational"
     Tasks: 38 (limit: 24624)
    Memory: 545.7M
    CGroup: /system.slice/mysqld.service
           └─44910 /usr/sbin/mysqld
```

# Linux 系统安装 Django 实例

## 安装 mysql

### 本地登录密码设置

```
[admin@localhost mysql]$ sudo cat /var/log/mysqld.log | grep password
2022-05-31T13:00:09.772283Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: x*d;#z_23:%F
[admin@localhost mysql]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.25

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

输入临时密码进行登录

# Linux系统安装Django实例

## 安装mysql

修改root密码

```
mysql> set global validate_password.policy=0; 密码等级降为0级
Query OK, 0 rows affected (0.00 sec)

mysql> set global validate_password.length=6; 密码最短长度为6
Query OK, 0 rows affected (0.00 sec)

mysql> set global validate_password.mixed_case_count=0; 大小写字母个数
Query OK, 0 rows affected (0.00 sec)

mysql> alter user 'root'@'localhost' identified by 'root1234'; 修改root密码为
Query OK, 0 rows affected (0.01 sec)
root1234

mysql> █
```

# Linux系统安装Django实例

## 安装mysql

验证新密码

```
mysql> exit
Bye
[admin@localhost mysql]$ mysql -u root -p
Enter password: ← 此时输入新的密码root1234
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.25 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement
.
mysql>
```

# Linux系统安装Django实例

## 安装mysql

### 远程登录设置



```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select host,user from user; 查看用户信息
+-----+-----+
| host | user |
+-----+-----+
| localhost | mysql.infoschema |
| localhost | mysql.session |
| localhost | mysql.sys |
| localhost | root |
+-----+-----+
4 rows in set (0.00 sec)

mysql> create user 'root'@'%' identified by 'root1234'; 创建远程登录用户
Query OK, 0 rows affected (0.01 sec)

mysql> grant all privileges on *.* to 'root'@'%';
```

```
mysql> grant all privileges on *.* to 'root'@'%'; 为用户添加权限
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select host,user,plugin from user; 查看信息
```

```
+-----+-----+-----+
| host | user | plugin |
+-----+-----+-----+
| % | root | caching_sha2_password |
| localhost | mysql.infoschema | caching_sha2_password |
| localhost | mysql.session | caching_sha2_password |
| localhost | mysql.sys | caching_sha2_password |
| localhost | root | caching_sha2_password |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> █
```

# Linux系统安装Django实例

## 安装mysql

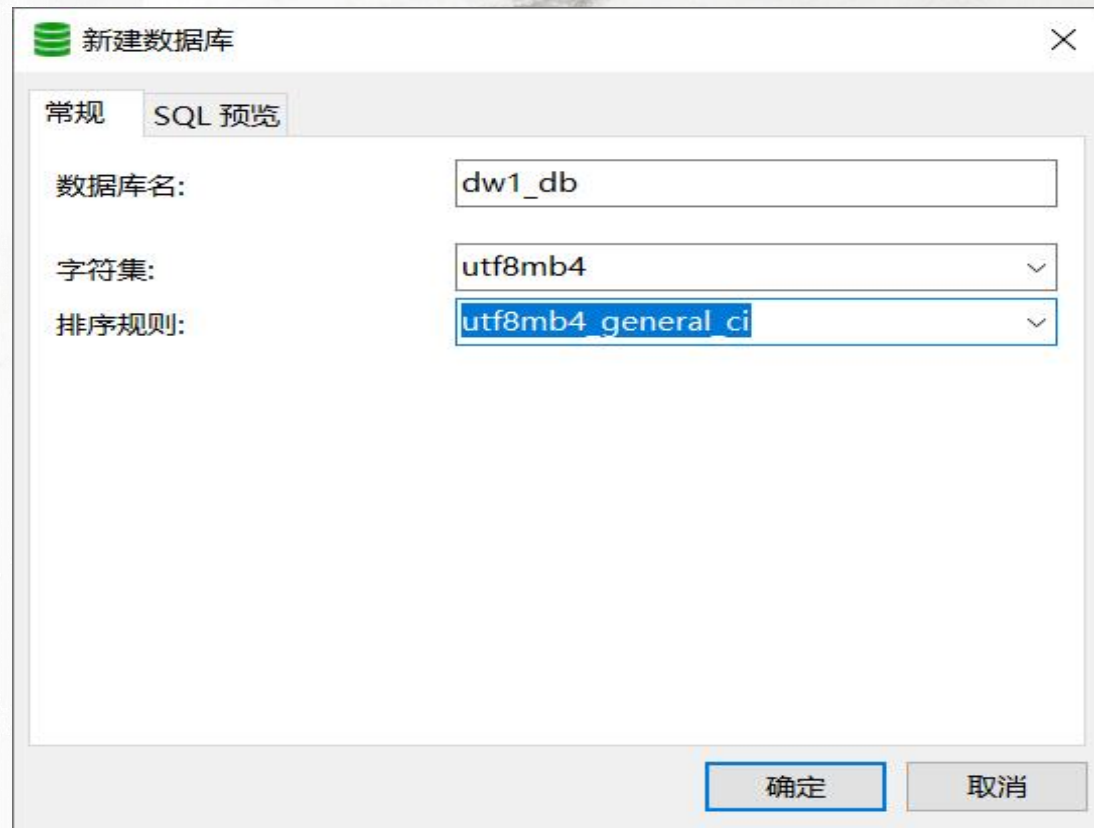
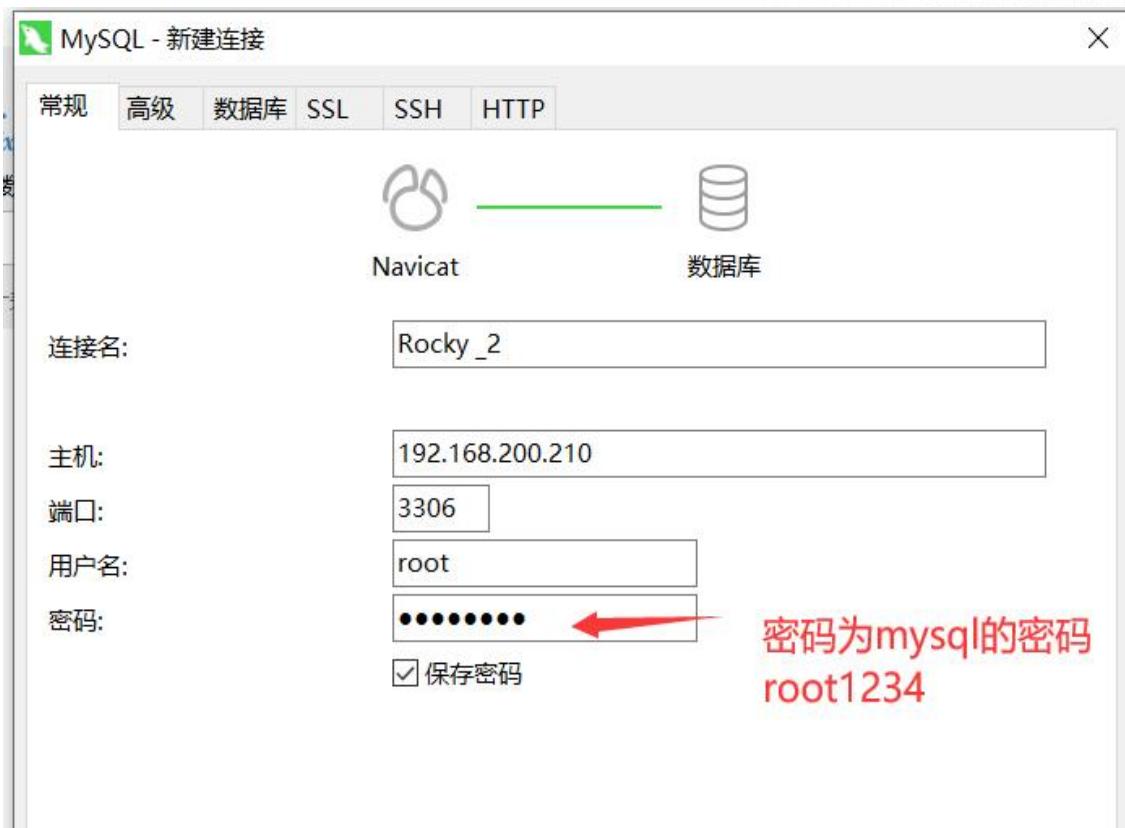
配置防火墙,配置完防火墙mysql的安装就已完成.

```
[admin@localhost mysql]$ sudo firewall-cmd --zone=public --add-port=3306/tcp --permanent
success
[admin@localhost mysql]$ sudo firewall-cmd --reload 重新加载防火墙 开放3306端口
success
[admin@localhost mysql]$ sudo firewall-cmd --zone=public --list-ports 查看所有开放的端口
3306/tcp
[admin@localhost mysql]$ sudo firewall-cmd --permanent --list-all 查看防火墙规则
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 3306/tcp
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[admin@localhost mysql]$
```

# Linux系统安装Django实例

## 安装Django及发布

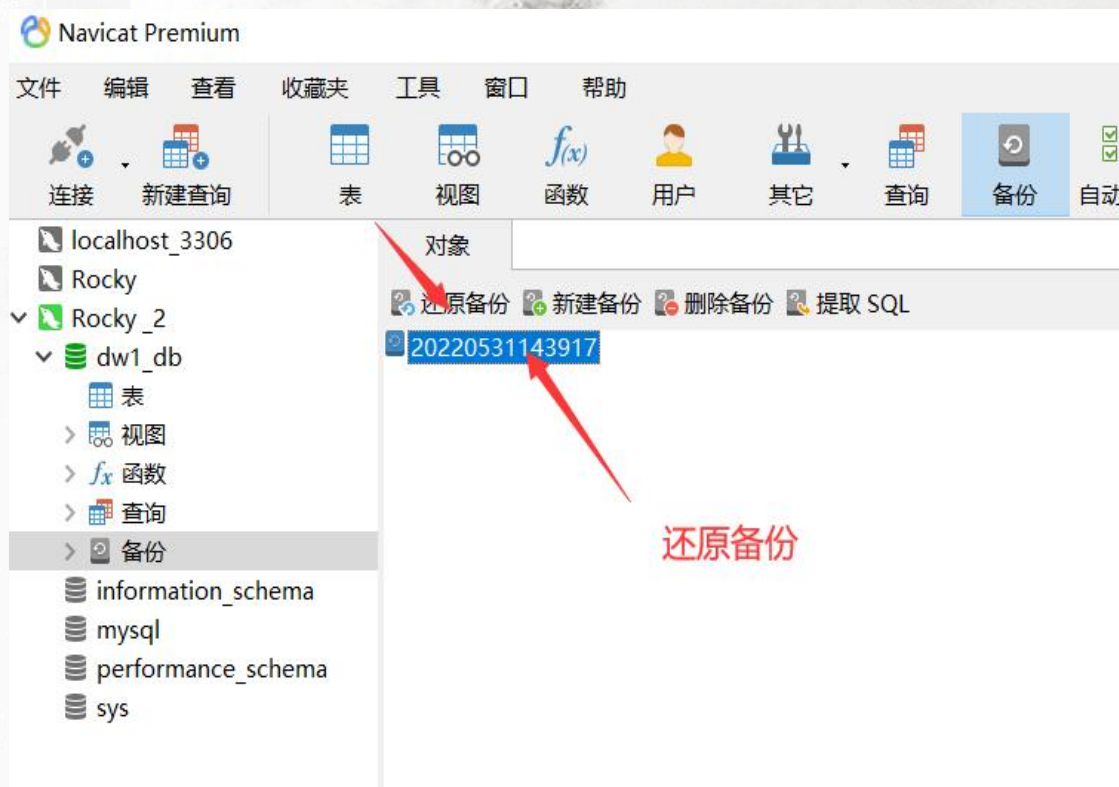
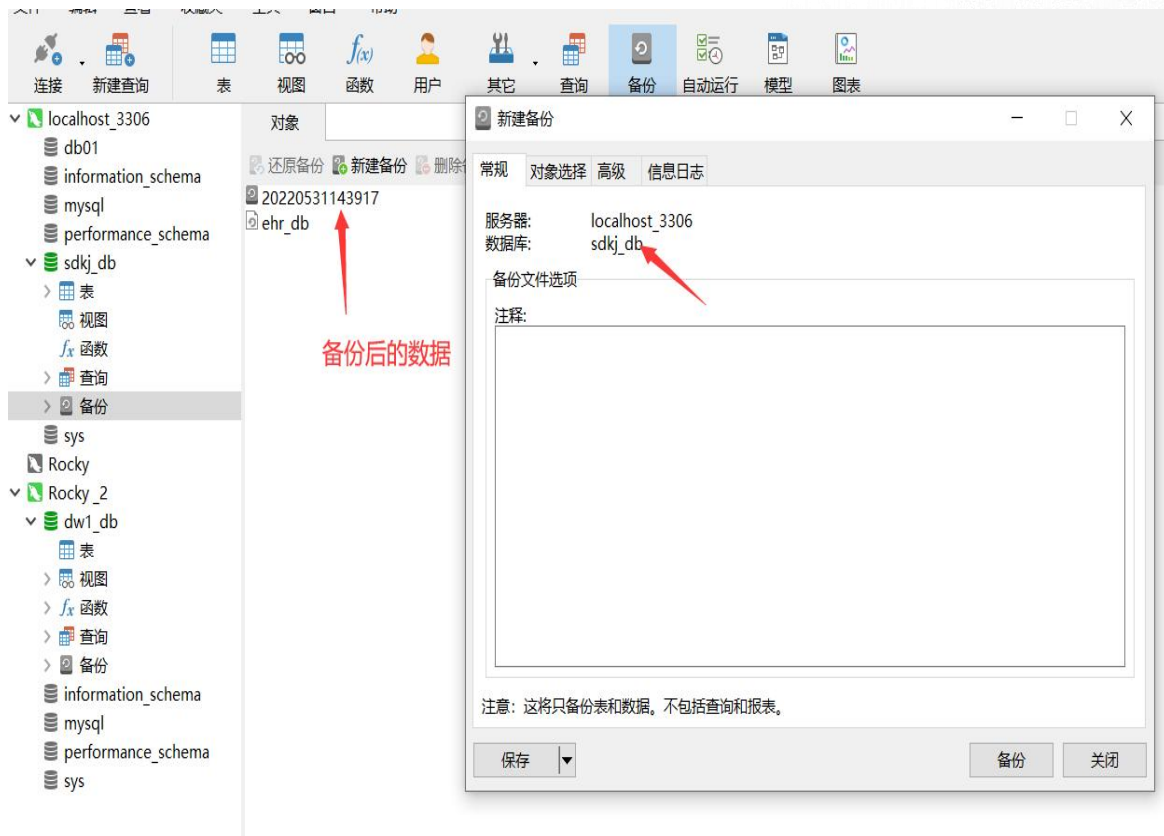
在Navicat中创建一个新连接并创建一个名为dw1\_db的数据库



# Linux 系统安装 Django 实例

## 安装 Django 及发布

创建之前实例的备份, 导入到 dw1\_db 数据库中



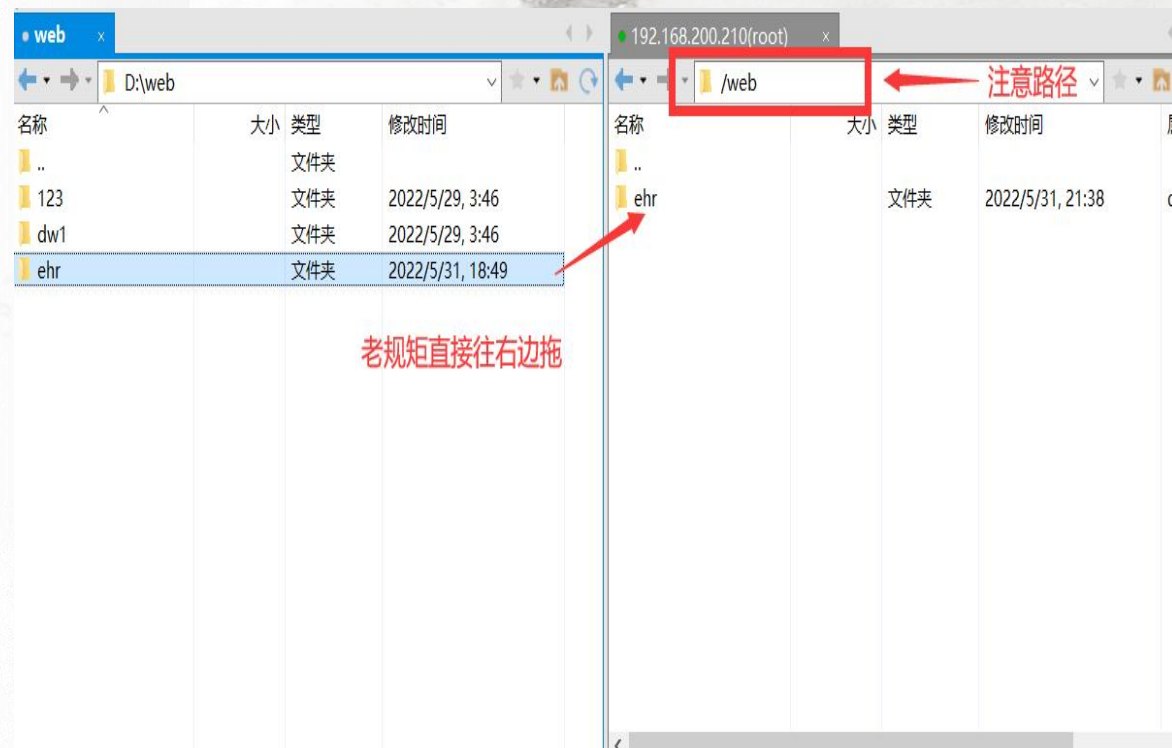


# Linux系统安装Django实例

## 安装Django及发布

通过xftp上传之前实例中的文件到web目录,使用root用户在根下创建web目录

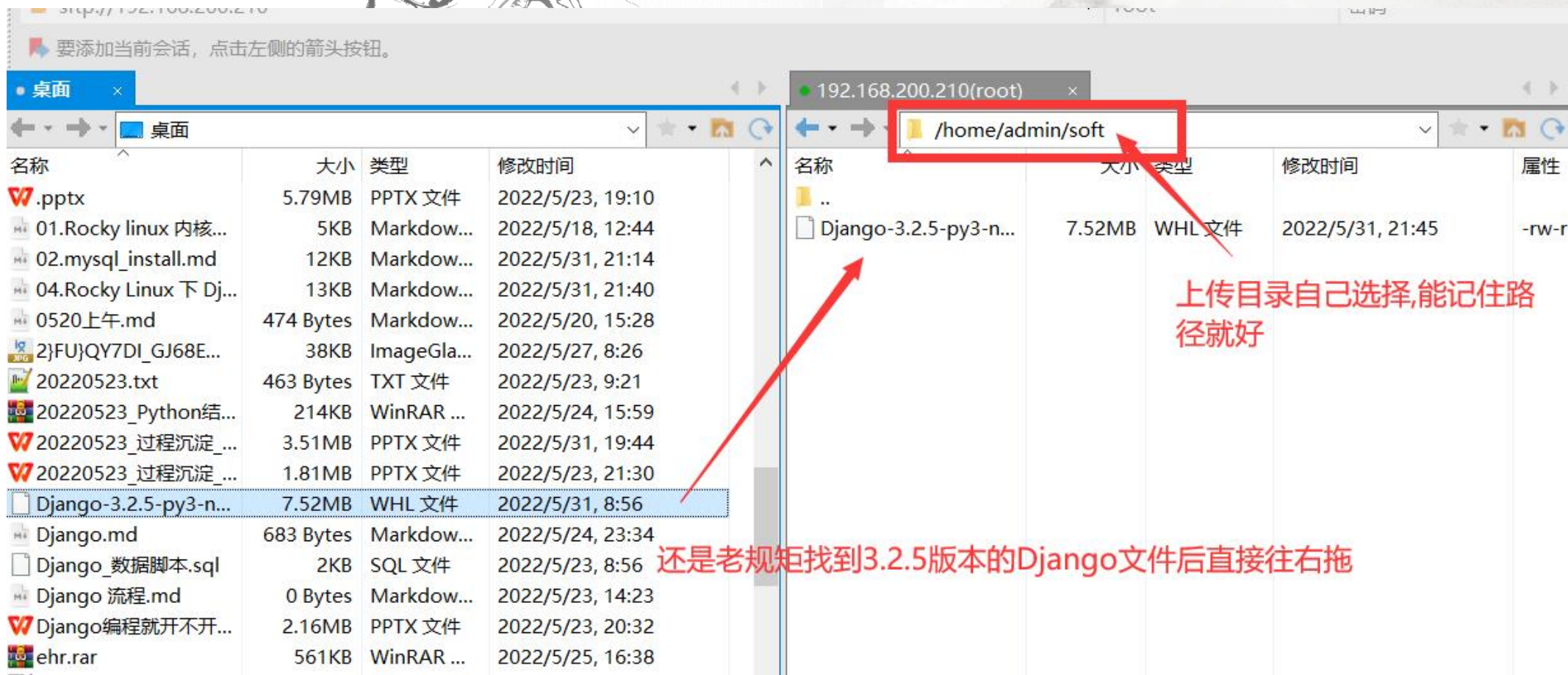
```
[root@localhost ~]# cd /  
[root@localhost /]# mkdir web  
[root@localhost /]# ls  
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var  
boot  etc  lib  media  opt  root  sbin  sys  usr  web  
[root@localhost /]#
```



# Linux系统安装Django实例

## 安装Django及发布

将3.2.5版本的Django通过xftp上传到虚拟机



# Linux系统安装Django实例

## 安装Django及发布

进入上传Django的目录进行离线安装并查看版本号

```
[root@localhost soft]# pip3 install Django-3.2.5-py3-none-any.whl
WARNING: Running pip install with root privileges is generally not a good idea.
Try `pip3 install --user` instead.
Processing ./Django-3.2.5-py3-none-any.whl
Collecting asgiref<4,>=3.3.2 (from Django==3.2.5)
  Downloading https://files.pythonhosted.org/packages/fe/66/577f32b54c50dcd8dec38447258e82ed327ecb86820d67ae7b3dea784f13/asgiref-3.4.1-py3-none-any.whl
Requirement already satisfied: pytz in /usr/lib/python3.6/site-packages (from Django==3.2.5)
Collecting sqlparse>=0.2.2 (from Django==3.2.5)
  Downloading https://files.pythonhosted.org/packages/05/40/d836d55fb3f467243ee839ab7b814822fda522cd395fa41e282684e71ee5/sqlparse-0.4.2-py3-none-any.whl (42kB)
    100% |#####| 51kB 118kB/s
Collecting typing-extensions; python_version < "3.8" (from asgiref<4,>=3.3.2->Django==3.2.5)
  Downloading https://files.pythonhosted.org/packages/45/6b/44f7f8f1e110027cf88956b59f2fad776cca7e1704396d043f89effd3a0e/typing_extensions-4.1.1-py3-none-any.whl
Installing collected packages: typing-extensions, asgiref, sqlparse, Django
Successfully installed Django-3.2.5 asgiref-3.4.1 sqlparse-0.4.2 typing-extensions-4.1.1
[root@localhost soft]# django-admin --version
3.2.5
[root@localhost soft]#
```

# Linux系统安装Django实例

## 安装Django及发布

进入/web/ehr/ehr下, 修改setting.py文件

```
[root@localhost ~]# cd /web//ehr/ehr/  
[root@localhost ehr]# ls  
asgi.py      __pycache__  settings.py  urls.py     wsgi.py  
__init__.py services     system      views  
[root@localhost ehr]#
```

# Linux系统安装Django实例

## 安装Django及发布

```
ALLOWED_HOSTS = ['192.168.200.210', '0.0.0.0:8000', '127.0.0.1', 'localhost']
```

```
# Application definition
```

添加信息,并把地址改为虚拟机的IP地址

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql', # 默认数据库  
        'NAME': 'dw1_db', # 修改之前创建的数据库名称 # 连接的数据库  
        'HOST': 'localhost', # 把名称换成localhost # mysql的ip地址  
        'PORT': 3306, # mysql的端口  
        'USER': 'root', # mysql的用户名  
        'PASSWORD': 'root1234', # 添加数据库密码 # mysql的密码  
    }  
}  
  
# Password validation  
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators  
  
AUTH_PASSWORD_VALIDATORS = [  
    {  
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',  
    },  
    {  
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',  
    },  
    # ...  
]  
  
-- 插入 --
```

93,1

66%

# Linux系统安装Django实例

## 安装Django及发布

修改防火墙规则, 开放80和8000端口

```
[root@localhost ~]# firewall-cmd --zone=public --add-port=8000/tcp --permanent
success
[root@localhost ~]# sudo firewall-cmd --zone=public --add-port=80/tcp --permanent
success
[root@localhost ~]# sudo firewall-cmd --reload
success
[root@localhost ~]# sudo firewall-cmd --permanent --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 3306/tcp 8000/tcp 80/tcp
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost ~]#
```

# Linux系统安装Django实例

## 安装Django及发布

### 启动服务器访问Django

```
[root@localhost ehr]# pwd
/web/ehr/ehr
[root@localhost ehr]# ls
asgi.py __init__.py __pycache__ services settings.py system urls.py views wsgi.py
[root@localhost ehr]# cd ..
[root@localhost ehr]# pwd
/web/ehr
[root@localhost ehr]# ls
db.sqlite3 ehr html manage.py
[root@localhost ehr]# python3 manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...
```

首先用pwd命令查看当前目录,是否有manage.py这个文件,  
如果没有cd ..返回上一级目录,找到manage.py文件后在执行命令

不要忘了加上0.0.0.0:8000,否则远程登录会失败

```
System check identified no issues (0 silenced).
```

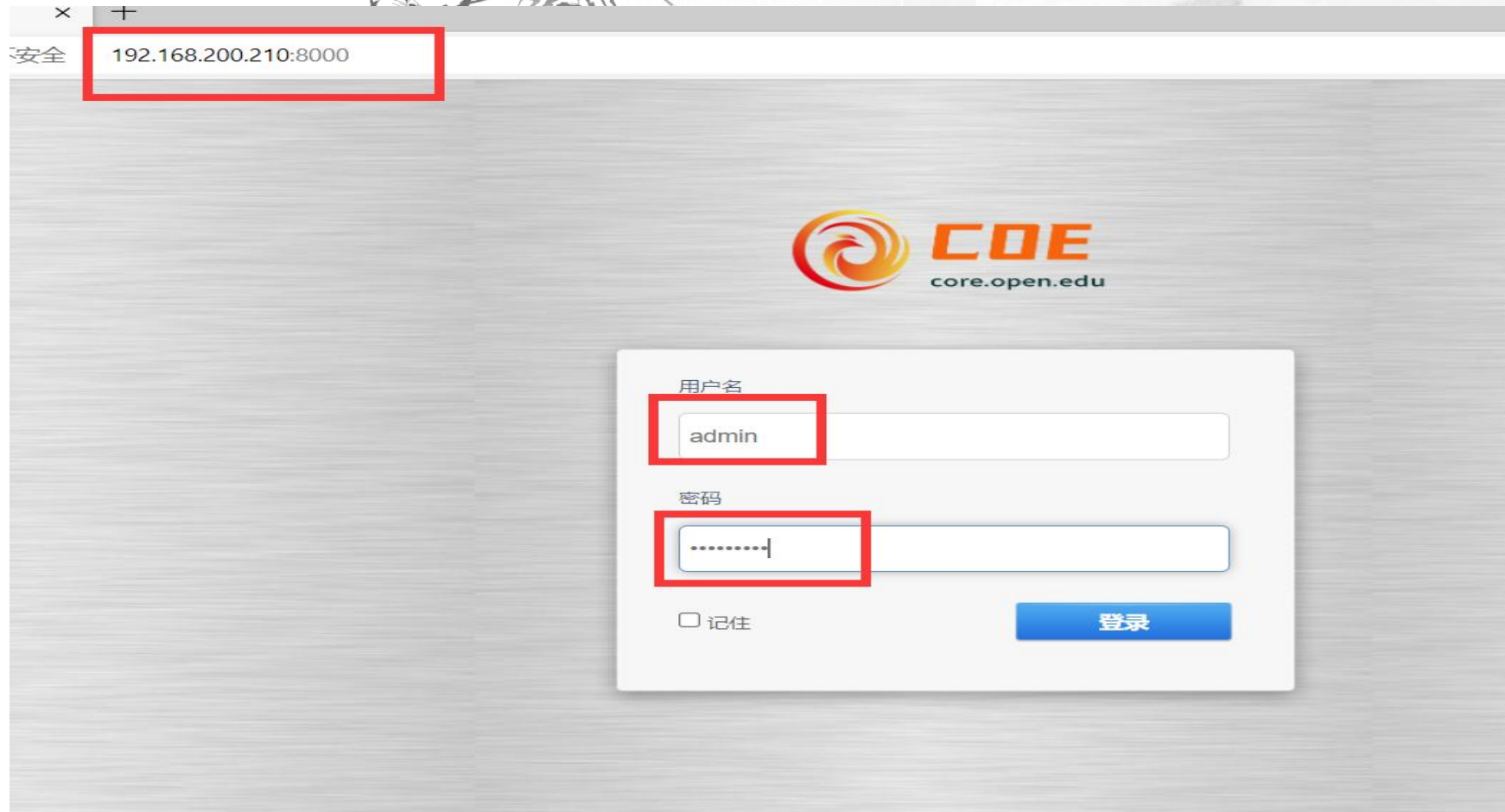
```
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
```

红色信息是指是否需要迁移Django中的表,可以忽略,此处我们选择忽略

```
May 31, 2022 - 14:29:06
Django version 3.2.5, using settings 'ehr.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

# Linux系统安装Django实例

安装Django及发布  
登录浏览器测试





# Linux系统安装Django实例

## 安装Django及发布

经过测试,自此在Linux系统中部署Django服务的实例就已完成

技术验证平台套版

系统管理员

### 员工管理

查询条件

员工编号:  姓名:  性别:  男  女  ==不限== 民族:  ==不限==

序号	编号	姓名	性别	生日	民族	邮件	电话	职务	
1	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
2	E1001	刘备	男	2022-05-29	汉族	12	1151	程序员	[删除]
3	E1001	赵三	男	2022-05-29	维族	12	1151	程序员	[删除]
4	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
5	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
6	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
7	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
8	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
9	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
10	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
11	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
12	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
13	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
14	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]
15	E1001	员工	男	2022-05-29	汉族	12	1151	程序员	[删除]

第[1]页

查询 添加



感谢您的聆听！