# 1.用pycharm创建一个day16项目,使用我们创建的虚拟环境

|  |
| --- |
|  |
|  |

# 2.需要把默认的templates文件夹删除然后在settings.py里面把设置也删除

|  |
| --- |
|  |

# 3.为项目创建app,使用命令python manage.py startapp day16app也可以在oycharm里面点击Tools->run manage.py task,就会打开一个特殊的终端

|  |
| --- |
|  |

# 4.这就相当于pycharm帮你输入了python manage.py,你只需要在后面输入startapp day16app回车,就可以创建app了

|  |
| --- |
|  |

# 5.在day16包里面的settings.py中注册我们的day16app

|  |
| --- |
|  |

# 6.在day16app里面新建一个templates文件夹,并且在上面点击右键->Mark Directory as->template folder把它标记为模板文件夹

|  |
| --- |
|  |

## 然后他的颜色就会改变,说明标记成功

|  |
| --- |
|  |

## 注意:是templates因为有很多模板,如果你写成template,django就找不到模板文件

# 7.在day16app里面新建一个static文件夹,把我们需要用到的bootstrap和jquery和图片等等放置到里面.然后我们需要在settings.py里面设置静态文件夹

|  |
| --- |
|  |

## 然后在urls.py里面做如下配置

|  |
| --- |
|  |
|  |

## 然后我们把django默认的路由注释了,我们来添加应该home路由

|  |
| --- |
|  |
|  |
|  |

## 然后我们来访问一下static里面的img文件夹里面的1.jpg文件,效果如下

|  |
| --- |
|  |

### 说明静态文件夹配置成功,最基本的设置就做好了,然后我们可以继续往下做项目.

# 8.设计表结构(前提需要你安装好MySQL服务和navicat,navicat的下载地址: <https://gitcode.com/open-source-toolkit/654d7> ,使用git clone命令来下载).我们这里使用MySQL8.4.6lts版本.我们只需要创建应该数据库,然后配置链接,然后就可以使用django的orm来创建表格了.

## 1>设计数据表,我们在day16app的models.py里面添加几个类,然后使用数据迁移来创建表格

|  |
| --- |
| **from django.db import models   *# Create your models here.* class Department(models.Model):  *"""部门表"""  # id = models.BigAutoField(verbose\_name="ID",primary\_key=True) #不需要写django会自动处理这个表字段* title = models.CharField(verbose\_name="标题", max\_length=32)   class UserInfo(models.Model):  *"""员工表"""* name = models.CharField(verbose\_name="姓名", max\_length=32)  password = models.CharField(verbose\_name="密码", max\_length=64)  age = models.IntegerField(verbose\_name="年龄", default=18) *# 需要添加默* account = models.DecimalField(verbose\_name="账户余额", max\_digits=10, decimal\_places=2, default=0)  create\_time = models.DateTimeField(verbose\_name="入职时间")  *# 这个类型不对,它是没有约束的  # dep\_id=models.BigIntegerField(verbose\_name="部门ID")  # 这个类型有约束的,也就是外键  # 只需要写dep,然后django会把它变为dep\_id  # 方式1,需要设置关联的表(类),还要设置表的关联的列,还需要设置部门被删除了,对应的员工数据会级联删除.* dep = models.ForeignKey(to="Department", to\_field="id",on\_delete=models.CASCADE)  *# 方式2.需要设置关联的表(类),还要设置表的关联的列,还需要设置可以置空,也就是部门被删除了,对应员工的部门变为空  # dep = models.ForeignKey(to="Department", to\_field="id",null=True,blank=True, on\_delete=models.SET\_NULL)* gender\_choices = (  (0,"女"),  (1,"难")  )  gender = models.SmallIntegerField(verbose\_name="性别",choices=gender\_choices) *# 注意django中性别的处理方式*** |

## 2>在mysql数据库中创建一个day16数据库

|  |
| --- |
|  |

## 3>然后我们在settings.py里面配置数据库

|  |
| --- |
|  |

## 4>然后我们就可以使用orm来迁移数据到数据库,打开终端,先输入:

### python manage.py makemigrations

|  |
| --- |
|  |

### 然后输入: python manage.py migrate

|  |
| --- |
|  |

### 运行成功后,数据表就生成了

|  |
| --- |
|  |

# 9.部门管理我们先学习最原始的方式,django有提供Form和ModelForm组件

## 9.1.做部门列表页面展示数据,我们的页面结构如图

|  |
| --- |
|  |

## 9.2我们在urls.py里面添加一个路由dep/list/

|  |
| --- |
|  |

## 9.3然后我们在views.py里面定义这个dep\_list函数

|  |
| --- |
|  |

## 9.4然后我们需要在templates文件夹里面创建这个dep\_list.html,然后我们到boostrap官网拷贝一个导航条过来,适当修改一下

|  |
| --- |
| **<nav class="navbar navbar-default">  <div class="container-fluid">  *<!-- Brand and toggle get grouped for better mobile display -->* <div class="navbar-header">  <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">  <span class="sr-only">Toggle navigation</span>  <span class="icon-bar"></span>  <span class="icon-bar"></span>  <span class="icon-bar"></span>  </button>  <a class="navbar-brand" href="#">用户管理系统</a>  </div>   *<!-- Collect the nav links, forms, and other content for toggling -->* <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">  <ul class="nav navbar-nav">  <li><a href="/dep/list">部门管理</a></li>  <li><a href="#">Link</a></li>  </ul>  <ul class="nav navbar-nav navbar-right">  <li><a href="#">登录</a></li>  <li class="dropdown">  <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">当前用户名称 <span class="caret"></span></a>  <ul class="dropdown-menu">  <li><a href="#">个人资料</a></li>  <li><a href="#">我的信息</a></li>  <li role="separator" class="divider"></li>  <li><a href="#">注销</a></li>  </ul>  </li>  </ul>  </div>*<!-- /.navbar-collapse -->* </div>*<!-- /.container-fluid -->* </nav>** |

### 这个dep\_list.html的内容如下

|  |
| --- |
| {% **load static** %} <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>部门列表</**title**>  <**link rel="stylesheet" href="**{% **static 'plugins/bootstrap-3.4.1/css/bootstrap.min.css'** %}**"**>  <**style**>  .**table**{  **width**: 100%;  }  .**navbar**{  **border-radius**: 0;  }  </**style**> </**head**> <**body**>  <**nav class="navbar navbar-default"**>  <**div class="container"**>  *<!-- Brand and toggle get grouped for better mobile display -->* <**div class="navbar-header"**>  <**button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false"**>  <**span class="sr-only"**>Toggle navigation</**span**>  <**span class="icon-bar"**></**span**>  <**span class="icon-bar"**></**span**>  <**span class="icon-bar"**></**span**>  </**button**>  <**a class="navbar-brand" href="#"**>用户管理系统</**a**>  </**div**>   *<!-- Collect the nav links, forms, and other content for toggling -->* <**div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1"**>  <**ul class="nav navbar-nav"**>  <**li**><**a href="/dep/list"**>部门管理</**a**></**li**>  <**li**><**a href="#"**>Link</**a**></**li**>  </**ul**>  <**ul class="nav navbar-nav navbar-right"**>  <**li**><**a href="#"**>登录</**a**></**li**>  <**li class="dropdown"**>  <**a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false"**>当前用户名称 <**span class="caret"**></**span**></**a**>  <**ul class="dropdown-menu"**>  <**li**><**a href="#"**>个人资料</**a**></**li**>  <**li**><**a href="#"**>我的信息</**a**></**li**>  <**li role="separator" class="divider"**></**li**>  <**li**><**a href="#"**>注销</**a**></**li**>  </**ul**>  </**li**>  </**ul**>  </**div**>*<!-- /.navbar-collapse -->* </**div**>*<!-- /.container-fluid -->* </**nav**>  <**div**>  <**div class="container"**>   <**div style="margin-bottom**: 5**px**;**"**>  <**a href="/dep/add/" class="btn btn-primary"**>  <**span class="glyphicon glyphicon-plus-sign" aria-hidden="true"**></**span**>  新增部门  </**a**>  </**div**>   <**div class="panel panel-default"**>  *<!-- Default panel contents -->* <**div class="panel-heading"**>  <**span class="glyphicon glyphicon-th-list" aria-hidden="true"**></**span**>  部门列表  </**div**>   <**table class="table table-bordered table-responsive"**>  <**thead**>  <**tr**>  <**th**>ID</**th**>  <**th**>部门名称</**th**>  <**th**>操作</**th**>  </**tr**>  </**thead**>  <**tbody**>  {% **for dep in deps** %}  <**tr**>  <**td**>{{ **dep**.**id** }}</**td**>  <**td**>{{ **dep**.**title** }}</**td**>  <**td**>  <**a href="" class="btn btn-success btn-xs"**>编辑</**a**>  <**a href="/dep/del/?nid=**{{ **dep**.**id** }}**" class="btn btn-danger btn-xs"**>删除</**a**>  </**td**>  </**tr**>  {% **endfor** %}  </**tbody**>  </**table**>  </**div**>    </**div**>  </**div**>  <**script src="**{% **static 'js/jquery3.7.1.min.js'** %}**"**></**script**>  <**script src="**{% **static 'plugins/bootstrap-3.4.1/js/bootstrap.min.js'** %}**"**></**script**> </**body**> </**html**> |

## 9.5.我们先通过navicat网数据库添加2条数据,

|  |
| --- |
|  |

### 然后刷新页面,效果如下

|  |
| --- |
|  |

# 10.然后我们来实现新增功能

## 10.1先在urls.py里面添加一个dep/add/路由,把它和views.dep\_add函数对应起来

|  |
| --- |
|  |

## 10.2然后我们在views.py里面编写一个dep\_app函数,先渲染一个叫做dep\_add.html的页面返回给用户

|  |
| --- |
|  |

## 10.3然后我们在templates文件夹里面新建一个dep\_add.html模板文件,注意我们这里需要统一风格,所以把列表页面里面的导航条拷贝粘贴进来.然后我们做一下修改,这个页面的内容如下

|  |
| --- |
| {% **load static** %} <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>新增部门</**title**>  <**link rel="stylesheet" href="**{% **static 'plugins/bootstrap-3.4.1/css/bootstrap.min.css'** %}**"**>  <**style**>  .**navbar**{  **border-radius**: 0;  }  </**style**> </**head**> <**body**> <**nav class="navbar navbar-default"**>  <**div class="container"**>  *<!-- Brand and toggle get grouped for better mobile display -->* <**div class="navbar-header"**>  <**button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false"**>  <**span class="sr-only"**>Toggle navigation</**span**>  <**span class="icon-bar"**></**span**>  <**span class="icon-bar"**></**span**>  <**span class="icon-bar"**></**span**>  </**button**>  <**a class="navbar-brand" href="#"**>用户管理系统</**a**>  </**div**>   *<!-- Collect the nav links, forms, and other content for toggling -->* <**div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1"**>  <**ul class="nav navbar-nav"**>  <**li**><**a href="/dep/list"**>部门管理</**a**></**li**>  <**li**><**a href="#"**>Link</**a**></**li**>  </**ul**>  <**ul class="nav navbar-nav navbar-right"**>  <**li**><**a href="#"**>登录</**a**></**li**>  <**li class="dropdown"**>  <**a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false"**>当前用户名称 <**span class="caret"**></**span**></**a**>  <**ul class="dropdown-menu"**>  <**li**><**a href="#"**>个人资料</**a**></**li**>  <**li**><**a href="#"**>我的信息</**a**></**li**>  <**li role="separator" class="divider"**></**li**>  <**li**><**a href="#"**>注销</**a**></**li**>  </**ul**>  </**li**>  </**ul**>  </**div**>*<!-- /.navbar-collapse -->* </**div**>*<!-- /.container-fluid -->* </**nav**> <**div**>  <**div class="container"**>  <**div class="panel panel-default"**>  *<!-- Default panel contents -->* <**div class="panel-heading"**>  <**h3 class="panel-title"**>新建部门</**h3**>  </**div**>  <**div class="panel-body"**>  *{# 表单 #}* <**form action="/dep/add/" method="post"**>  <**div class="form-group"**>  <**label**>部门名称</**label**>  <**div**>  <**input type="text" class="form-control" name="title" placeholder="输入部门"**>  </**div**>  </**div**>  <**button type="submit" class="btn btn-success"**>保 存</**button**>  </**form**>  </**div**>  </**div**>   </**div**> </**div**>    <**script src="**{% **static 'js/jquery3.7.1.min.js'** %}**"**></**script**>  <**script src="**{% **static 'plugins/bootstrap-3.4.1/js/bootstrap.min.js'** %}**"**></**script**> </**body**> </**html**> |

### 运行一下,效果如下

|  |
| --- |
|  |

## 10.4然后我们再回到views.py里面的dep\_add函数,我们来添加post请求的处理代码

|  |
| --- |
|  |

### 运行程序,发现添加成功了

|  |
| --- |
|  |

# 11.然后我们来实现删除功能

## 11.1先在urls.py里面添加一条删除路由dep/del/,把它和views.py里面的dep\_del函数对应起来

|  |
| --- |
|  |

## 11.2然后我们在views.py里面编写一个dep\_del函数来实现删除功能

|  |
| --- |
|  |

### 运行程序,找到需要删除的数据,比如最后一个,点击删除

|  |
| --- |
|  |

### 该部门就被删除了,而且还跳转回列表页面

|  |
| --- |
|  |

# 12.我们来做做修改功能

## 12.1.在urls.py里面添加一个修改路由dep/edit/,把它和view.py里面的dep\_edit函数对应起来

|  |
| --- |
|  |

## 12.2在dep\_list.html里面给编辑超链接添加链接,传递一个部的id

|  |
| --- |
|  |

## 12.3在views.py里面编写一个dep\_edit函数,在函数里面先处理get请求,获取需要修改的部门的id,然后根据这个id来查询部门,渲染模板文件把找到的数据传递给他

|  |
| --- |
|  |

## 12.4.然后把dep\_add.html复制一份改名dep\_edit.html,然后我们来修改他的内容,修改后内容如下:

|  |
| --- |
| {% **load static** %} <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>新增部门</**title**>  <**link rel="stylesheet" href="**{% **static 'plugins/bootstrap-3.4.1/css/bootstrap.min.css'** %}**"**>  <**style**>  .**navbar**{  **border-radius**: 0;  }  </**style**> </**head**> <**body**> <**nav class="navbar navbar-default"**>  <**div class="container"**>  *<!-- Brand and toggle get grouped for better mobile display -->* <**div class="navbar-header"**>  <**button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false"**>  <**span class="sr-only"**>Toggle navigation</**span**>  <**span class="icon-bar"**></**span**>  <**span class="icon-bar"**></**span**>  <**span class="icon-bar"**></**span**>  </**button**>  <**a class="navbar-brand" href="#"**>用户管理系统</**a**>  </**div**>   *<!-- Collect the nav links, forms, and other content for toggling -->* <**div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1"**>  <**ul class="nav navbar-nav"**>  <**li**><**a href="/dep/list"**>部门管理</**a**></**li**>  <**li**><**a href="#"**>Link</**a**></**li**>  </**ul**>  <**ul class="nav navbar-nav navbar-right"**>  <**li**><**a href="#"**>登录</**a**></**li**>  <**li class="dropdown"**>  <**a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false"**>当前用户名称 <**span class="caret"**></**span**></**a**>  <**ul class="dropdown-menu"**>  <**li**><**a href="#"**>个人资料</**a**></**li**>  <**li**><**a href="#"**>我的信息</**a**></**li**>  <**li role="separator" class="divider"**></**li**>  <**li**><**a href="#"**>注销</**a**></**li**>  </**ul**>  </**li**>  </**ul**>  </**div**>*<!-- /.navbar-collapse -->* </**div**>*<!-- /.container-fluid -->* </**nav**> <**div**>  <**div class="container"**>  <**div class="panel panel-default"**>  *<!-- Default panel contents -->* <**div class="panel-heading"**>  <**h3 class="panel-title"**>编辑部门</**h3**>  </**div**>  <**div class="panel-body"**>  *{# 表单 #}* <**form action="/dep/edit/" method="post"**>  {% **csrf\_token** %}  <**div class="form-group"**>  <**label**>部门名称</**label**>  <**div**>  <**input type="hidden" class="form-control" name="id"value="**{{ **dep**.**id** }}**"**>  <**input type="text" class="form-control" name="title"value="**{{ **dep**.**title** }}**"**>  </**div**>  </**div**>  <**button type="submit" class="btn btn-success"**>更 新</**button**>  </**form**>  </**div**>  </**div**>   </**div**> </**div**>    <**script src="**{% **static 'js/jquery3.7.1.min.js'** %}**"**></**script**>  <**script src="**{% **static 'plugins/bootstrap-3.4.1/js/bootstrap.min.js'** %}**"**></**script**> </**body**> </**html**> |

### 测试一下,是可以获取到数据填充到表单的

|  |
| --- |
|  |

## 12.5然后我们回到views.py里面的dep\_edit函数,我们来添加修改数据的代码

|  |
| --- |
|  |

### 测试一下,在销售部上面点击编辑,就会进入编辑页面并且会弹出数据

|  |
| --- |
|  |
|  |

### 我们把它修改为人事部,点击更新

|  |
| --- |
|  |
|  |

### 数据修改成功,

## 13.成功是成功了,不过django还提供了另外一种路由,这个我们也需要学习一下.我们把代码修改一下

|  |
| --- |
|  |
|  |
|  |

### 然后我们来测试一下

|  |
| --- |
|  |
|  |
|  |

# 扩展,数据库的一些操作习惯

# 1.员工表的部门信息是存部门名称还是部门id?

|  |
| --- |
|  |

## 我们不是大公司,一般存储部门id

|  |
| --- |
|  |

# 2.部门id需要约束吗?

## (应该是作为外键)它只能够是在部门表里面存在的id

|  |
| --- |
|  |

# 3.部门被删除了,和它关联的员工怎么处理

## 有2中处理方法

|  |
| --- |
|  |