
网站 综合实训

项目：手机记录管理

(*) 建立 Phones 的 Django 项目

django-admin startproject Phones



```
管理员: 命令提示符
C:\myPython>django-admin startproject Phones
C:\myPython>
```

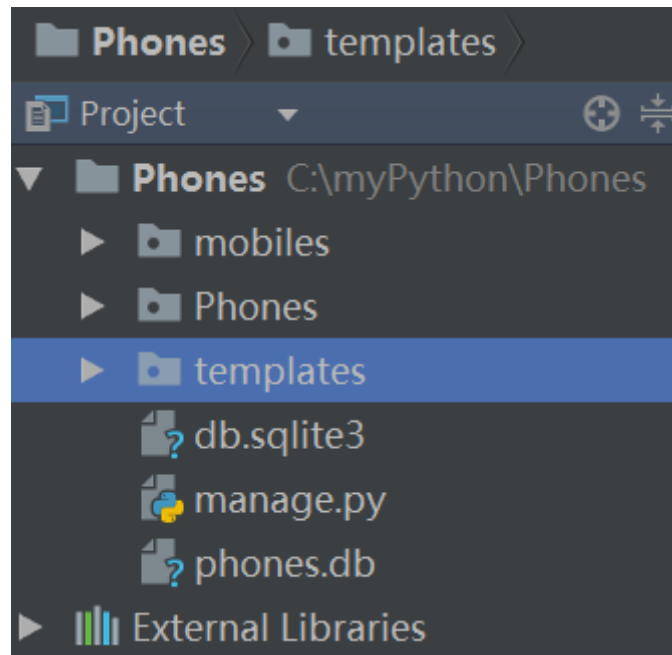
(*) 在 Phones 中建立 mobiles 的 app

python manage.py startapp mobiles



```
管理员: 命令提示符
C:\myPython>django-admin startproject Phones
C:\myPython>cd phones
C:\myPython\Phones>python manage.py startapp mobiles
C:\myPython\Phones>
```

(*) 在 Phones 中建立 templates 目录



(*) 在 `settings.py` 中安装 `mobiles` 的 app

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'mobiles'  
]
```

(*) 在 `settings.py` 中注释掉 `csrf`

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    '#django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]
```

(*) 在 `settings.py` 中设置 `templates` 目录

```
TEMPLATES = [  
    {
```

```
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [os.path.join(BASE_DIR, "templates")],
'APP_DIRS': True,
'OPTIONS': {
    'context_processors': [
        'django.template.context_processors.debug',
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
    ],
},
],
```

(*) 在 `settings.py` 中设置数据库为 `sqlite3`，名称是 `phones.db`

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'phones.db'),
    }
}
```

(*) 在 `settings.py` 中设置语言为中文，时区为上海

```
LANGUAGE_CODE = 'zh-Hans'
```

```
TIME_ZONE = 'Asia/Shanghai'
```

```
USE_I18N = True
```

```
USE_L10N = True
```

```
USE_TZ = True
```

(*) 在 `mobiles/models.py` 中建立 `Product` 类与 `User` 类

```
from django.db import models

# Create your models here.

class Product(models.Model):
    model=models.CharField(max_length=128,blank=False,verbose_name="型号")
    maker = models.CharField(max_length=128, blank=False,verbose_name="制造商")
    year=models.PositiveIntegerField(blank=False)
    price=models.PositiveIntegerField(default=0,verbose_name="价格")
```

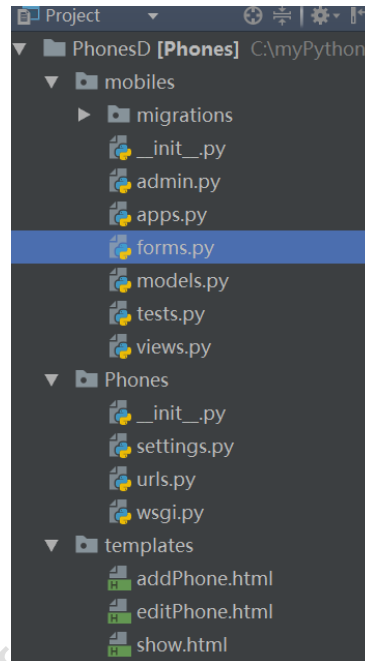
```

    qty=models.PositiveIntegerField(default=1,verbose_name="数量")
    desp = models.CharField(max_length=128, blank=False,verbose_name="简介")

class User(models.Model):
    name=models.CharField(max_length=128,primary_key=True)
    password=models.CharField(max_length=128,blank=False)

```

(*) 在 mobiles 中建立 forms.py



(*) 在 forms.py

```

from django import forms

# Create your models here.

class ProductForm(forms.Form):
    model=forms.CharField(max_length=128,required=True,label="型号")
    maker = forms.CharField(max_length=128, required=True,label="厂商")
    year=forms.IntegerField(required=True,label="年份")
    price=forms.IntegerField(initial=0,label="价格")
    qty=forms.IntegerField(initial=1,label="数量")
    desp = forms.CharField(max_length=128, required=False,label="简介")

class LoginForm(forms.Form):
    name=forms.CharField(max_length=128,required=True,label="用户")
    password = forms.CharField(max_length=128, required=True, label="密码",widget=forms.PasswordInput)

```

(*) 在 templates/show.html 显示

```
<script>
    function delPhone(id)
    {
        if(confirm("删除该记录?")) window.location.href="/show?id="+id;
    }
</script>

<h3 align="center">手机列表</h3>
<div align="center"><a href="/">注销</a>&nbsp;&nbsp;&nbsp;<a href="/add">增加
</a></div>
<table border="1" width="800" align="center">
    <tr><th>ID</th><th>型号</th><th>厂商</th><th>年份</th><th>价格
</th><th>数量</th><th>简介</th><th>删除</th><th>编辑</th></tr>
    {% for p in phones %}
    <tr><td>{{p.id}}</td><td>{{p.model}}</td><td>{{p.maker}}</td><td>{{
p.year}}</td><td>{{p.price}}</td><td>{{p.qty}}</td><td>{{p.desp}}</td>
    <td><a href="javascript:delPhone('{{p.id}}')">删除</a></td>
    <td><a href="/edit?id={{p.id}}">编辑</a></td>
    </tr>
    {% endfor %}
</table>
```

(*) 在 templates/login.html 显示

```
<h3 align="center">用户登录</h3>
<form action="" method="post">
<table border="1" width="300" align="center">
    {{form.as_table}}
    <tr><td colspan="2" align="center"><input type="submit" value="登录
"></td></tr>
</table>
</form>
<div align="center">{{msg}}</div>
```

(*) 在 templates/index.html 显示

```
<h3 align="center">手机列表</h3>
<div align="center"><a href="/login">登录</a></div>
<table border="1" width="800" align="center">
    <tr><th>ID</th><th>型号</th><th>厂商</th><th>年份</th><th>价格
</th><th>数量</th><th>简介</th></tr>
    {% for p in phones %}
```

```

        <tr><td>{{p.id}}</td><td>{{p.model}}</td><td>{{p.maker}}</td><td>{{
p.year}}</td><td>{{p.price}}</td><td>{{p.qty}}</td><td>{{p.desp}}</td><
/tr>
        {% endfor %}
</table>

```

(*) templates/addPhone.html

```

<h3 align="center">增加手机</h3>
<form action="" method="post">
<table border="1" width="300" align="center">
    {{form.as_table}}
    <tr><td colspan="2" align="center"><input type="submit" value="提交
"></td></tr>
</table>
</form>
<div align="center">{{msg}} &nbsp; <a href="/show">【返回】</a></div>

```

(*) templates/editPhone.html

```

<h3 align="center">编辑手机</h3>
<form action="" method="post">
<table border="1" width="300" align="center">
    {{form.as_table}}
    <tr><td colspan="2" align="center"><input type="submit" value="提交
"></td></tr>
</table>
    <input type="hidden" name="id" value="{{id}}" >
</form>
<div align="center">{{msg}} &nbsp; <a href="/show">【返回】</a></div>

```

注意：我们使用

<input type="hidden" name="id" value="{{id}}" >

来存储修改记录的 id

(*) 在 urls.py 中定义 url,

```

from django.contrib import admin
from django.conf.urls import url
from mobiles import views

urlpatterns = [
    url(r'^$', views.index),
    url(r'^show$', views.show),

```

```
url(r'^add$', views.addPhone),
url(r'^edit$', views.editPhone),
url(r'^login$', views.login),
]
```

(*) 在 `mobiles/views.py` 中设计函数

```
from django.shortcuts import render
from mobiles import models
from mobiles import forms
from django.http import HttpResponseRedirect

# Create your views here.

def initUser():
    user=models.User(name="admin",password="123")
    user.save()

def index(request):
    request.session["logged"]=" "
    initUser()
    phones=models.Product.objects.all()
    return render(request,"index.html",locals())

def show(request):
    if request.session.get("logged","")==":
        return HttpResponseRedirect("/")
    id=request.GET.get("id","")
    if id!="":
        models.Product.objects.filter(id=id).delete()
    phones=models.Product.objects.all()
    return render(request,"show.html",locals())

def addPhone(request):
    if request.session.get("logged","")==":
        return HttpResponseRedirect("/")
    msg=""
    if request.method=="POST":
        form=forms.ProductForm(request.POST)
        if form.is_valid:
            model=request.POST.get("model","")
            maker = request.POST.get("maker", "")
            year = request.POST.get("year", "")
            price = request.POST.get("price", "")
```

```
        qty = request.POST.get("qty", "")
        desp = request.POST.get("desp", "")
        if year=="":
            year="0"
        if price=="":
            price="0"
        if qty=="":
            qty="0"
        product=models.Product(model=model,maker=maker,year=int(year),price=int(price),qty=int(qty),desp=desp)
        product.save()
        msg="增加成功"
    else:
        msg="增加失败"
    else:
        form=forms.ProductForm()
    return render(request,"addPhone.html",locals())

def editPhone(request):
    if request.session.get("logged","")=="":
        return HttpResponseRedirect("/")
    msg=""
    if request.method=="POST":
        form=forms.ProductForm(request.POST)
        if form.is_valid:
            model=request.POST.get("model","")
            maker = request.POST.get("maker", "")
            year = request.POST.get("year", "")
            price = request.POST.get("price", "")
            qty = request.POST.get("qty", "")
            desp = request.POST.get("desp", "")
            if year=="":
                year="0"
            if price=="":
                price="0"
            if qty=="":
                qty="0"
            id=request.POST.get("id","")
            models.Product.objects.filter(id=id).update(model=model,maker=maker,year=int(year),price=int(price),qty=int(qty),desp=desp)
            msg="更新成功"
        else:
            msg="更新失败"
    else:
```



```

    try:
        id=request.GET.get("id","")
        p=models.Product.objects.get(id=id)
        d={"model":p.model,"maker":p.maker,"year":p.year,"price":p.
price,"qty":p.qty,"desp":p.desp}
        form = forms.ProductForm(d)
    except:
        form=forms.ProductForm()
    return render(request,"editPhone.html",locals())

def login(request):
    request.session["logged"]=" "
    msg=" "
    if request.method=="POST":
        form=forms.LoginForm(request.POST)
        name=request.POST.get("name","")
        password = request.POST.get("password", "")
        user=models.User.objects.filter(name=name,password=password)
        if user:
            request.session["logged"]="OK"
            return HttpResponseRedirect("/show")
        else:
            msg="登录失败"
    else:
        form=forms.LoginForm()
    return render(request,"login.html",locals())

```

(*) 执行命令

python manage.py makemigrations

python manage.py migrate

```

管理员: 命令提示符
C:\myPython\Phones>python manage.py makemigrations
Migrations for 'mobiles':
  mobiles\migrations\0001_initial.py
    - Create model Product

C:\myPython\Phones>

```

```
管理员: 命令提示符
C:\myPython\Phones>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, mobiles, 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 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 mobiles.0001_initial... OK
  Applying sessions.0001_initial... OK

C:\myPython\Phones>
```

(*) 运行服务器

```
管理员: 命令提示符 - python manage.py runserver
Superuser created successfully.

C:\myPython\Phones>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
May 26, 2018 - 11:50:26
Django version 2.0.2, using settings 'Phones.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

(*) 访问 <http://127.0.0.1:8000>



(*) 登录

127.0.0.1:8000/login

用户登录

用户: admin

密码: ...

登录

(*) 登录后可以增加、编辑、删除记录

127.0.0.1:8000/show

手机列表

[注销](#) [增加](#)

ID	型号	厂商	年份	价格	数量	简介	删除	编辑
1	P9 Plus	华为	2017	0	1		删除	编辑

127.0.0.1:8000/add

增加手机

型号: P9 Plus

厂商: 华为

年份: 2016

价格: 3000

数量: 1

简介:

提交

[【返回】](#)

编辑手机

型号:	P9 Plus
厂商:	华为
年份:	2016
价格:	3000
数量:	1
简介:	

提交

[【返回】](#)

(*) 注意在 **Product** 中没有定义 **primary key**, 那么系统自动产生一个 **ID** 的 **primary key**. 在 **users** 中定义 **name** 为 **primary key**, 因此不产生 **ID**.

SQLite Expert Personal 5.2 (x64)

Database: phones Table: mobiles_product File: C:\PhonesH\phones.db SQLite library: sqlite3.dll 3.21.0 [FTS3 FTS4 FTS5 RTREE]

rowid	id	model	maker	year	price	qty	desp
1	1	P9 Plus	华为	2017	0	2	
2	2	Mate 30	华为	2019	6000	1	

