

28-用户密码管理

1、修改密码

两个页面：
1) 实际修改密码的页面
2) 修改密码成功后的页面

在此之前我们已经完成了用户登录、注册、注销等功能，接下来让我们继续为用户提供修改密码的功能。该功能 Django 的 auth 应用也已经为我们提供，过程几乎和之前的登录功能完全一样。

(1) 编写修改密码模板

视图函数也是自带的，不需要手写的。而且自动装饰需要登录才能访问

修改密码的视图函数默认渲染的模板名为 **password_change_form.html**，因此首先在 registration/ 下新建一个 password_change_form.html 文件，写入表单代码（几乎和登录页面一样）。

【templates/registration/password_change_form.html】

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
  <meta charset="utf-8">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
  <title>修改密码</title>
  <link rel="stylesheet" href="https://unpkg.com/mobi.css/dist/mobi.min.css">
  <style>
    .errorlist {
      color: red;
    }
  </style>
</head>
<body>
<div class="flex-center">
  <div class="container">
    <div class="flex-center">
      <div class="unit-1-2 unit-1-on-mobile">
        <h1><a href="{% url 'blog:index' %}">Django Auth Example</a></h1>
        <h3>修改密码</h3>
```

```

        <form class="form" action="{% url 'password_change' %}"
method="post">
            {% csrf_token %}
            {{ form.non_field_errors }}
            {% for field in form %}
                {{ field.label_tag }}
                {{ field }}
                {{ field.errors }}
                {% if field.help_text %}
                    <p class="help text-small text-
muted">{{ field.help_text|safe }}</p>
                {% endif %}
            {% endfor %}
            <button type="submit" class="btn btn-primary btn-
block">确认修改</button>
        </form>
    </div>
</div>
</div>
</div>
</body>
</html>

```

此外，在首页加一个修改密码的按钮，并且注意只对已登录用户显示：

【templates/blog/base.html】

```

{% if user.is_authenticated %}
<li class="cl-effect-11"><a href="{% url 'password_change' %}">修改密码
</a></li>
<li class="cl-effect-11"><a href="{% url 'logout' %}">Logout</a></li>
{% else %}
<li class="cl-effect-11"><a href="{% url 'login' %}">Login</a></li>
{% endif %}

```

(2) 编写密码修改成功页面模板

密码修改成功后，Django 会把用户跳转到密码修改成功页面，该页面渲染的模板为 **password_change_done.html**，因此再添加一个密码修改成功页面的模板：

【templates/registration/password_change_done.html】

```

<!DOCTYPE html>
<html lang="zh-cn">
<head>
    <meta charset="utf-8">

```

```

<meta http-equiv="x-ua-compatible" content="ie=edge">
<meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
<title>密码修改成功</title>
<link rel="stylesheet"
href="https://unpkg.com/mobi.css/dist/mobi.min.css">
</head>
<body>
<div class="flex-center">
  <div class="container">
    <div>
      <h1 class="logo"><a href="{% url 'blog:index' %}">Django
Blog</a></h1>
      <p>密码修改成功! </p>
    </div>
  </div>
</div>
</body>
</html>

```

OK，修改密码的功能就完成了。流程为已登录用户点击主页的修改密码按钮跳转到修改密码页面，修改密码成功后跳转到修改成功页面。

2、重置密码

当用户不小心忘记了密码时，网站需要提供让用户找回账户密码的功能。在示例项目中，我们将发送一封含有重置用户密码链接的邮件到用户注册时的邮箱，用户点击收到的链接就可以重置他的密码，下面是具体做法。

(0) 发送邮件设置

Django 内置了非常方便的发送邮件的功能，不过需要在 settings.py 中做一些简单配置。生产环境下通常需要使用真实的邮件发送服务器，配置步骤会比较多一点。不过 Django 为开发环境下发送邮件提供了一些方便的 Backends 来模拟真实邮件的发送，例如直接发送邮件到终端（**相当于 Flask 里的邮件默认本地发送接收**）。在 settings.py 中加入以下设置：

【settings.py】

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
```

```
EMAIL_USE_TLS = True  transtion layer security
EMAIL_HOST = 'smtp.163.com'
EMAIL_PORT = 25
EMAIL_HOST_USER = 'python7456@163.com'
EMAIL_HOST_PASSWORD = 'python'
DEFAULT_FROM_EMAIL = 'python7456@163.com'
```

这样 Django 将把邮件发送到终端。

发送邮件设置好后，接下来的步骤和之前的登录，注册，修改密码等是完全类似的了，只需添加和修改相应模板即可。

共有 4 个模板：

1. 重置密码请求的模板：password_reset_form.html 只有一个字段：要重置密码的账号的邮箱
2. 重置密码请求完成的模板：password_reset_done.html 提示发送邮件成功
3. 设置新密码页面的模板：password_reset_confirm.html 实际重置密码的表单
4. 设置新密码页面完成的模板：password_reset_complete.html 重置密码成功的页面

（1）编写重置密码模板

重置的视图函数默认渲染的模板名为 password_reset_form.html，因此首先在 registration/ 下新建一个 password_reset_form.html 文件，写入表单代码（几乎和登录页面一样）。

【registration/password_reset_form.html】

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
  <meta charset="utf-8">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
  <title>重置密码</title>
  <link rel="stylesheet"
href="https://unpkg.com/mobi.css/dist/mobi.min.css">
  <style>
    .errorlist {
      color: red;
```

```

    }
</style>
</head>
<body>
<div class="flex-center">
    <div class="container">
        <div class="flex-center">
            <div class="unit-1-2 unit-1-on-mobile">
                <h1><a href="{% url 'blog:index' %}">Django
Blog</a></h1>
                <h3>重置密码</h3>
                <form class="form" action="{% url 'password_reset' %}"
method="post">
                    {% csrf_token %}
                    {{ form.non_field_errors }}
                    {% for field in form %}
                        {{ field.label_tag }}
                        {{ field }}
                        {{ field.errors }}
                        {% if field.help_text %}
                            <p class="help text-small text-
muted">{{ field.help_text|safe }}</p>
                        {% endif %}
                    {% endfor %}
                    <button type="submit" class="btn btn-primary btn-
block">提交</button>
                </form>
            </div>
        </div>
    </div>
</div>
</body>
</html>

```

此外，修改一下重置密码按钮的超链接属性：

【registration/login.html】

```

<div class="flex-left top-gap text-small">
    <div class="unit-2-3"><span>没有账号? <a href="{% url
'users:register' %}">立即注册</a></span></div>
    <div class="unit-1-3 flex-right"><span><a href="{% url
'password_reset' %}">忘记密码? </a></span></div>
</div>

```

(3) 编写邮件发送成功页面模板

用户在重置密码页面输入注册时的邮箱后，Django 会把用户跳转到邮件发送成功页面，该页面渲染的模板为 `password_reset_done.html`，因此再添加一个密码修改成功页面的模板：

【`registration/password_reset_done.html`】

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
    <title>密码重置链接已经发送</title>
    <link rel="stylesheet"
href="https://unpkg.com/mobi.css/dist/mobi.min.css">
</head>
<body>
<div class="flex-center">
    <div class="container">
        <div>
            <h1 class="logo"><a href="{% url 'blog:index' %}">Django
Blog</a></h1>
            <h3>密码重置链接已经发送</h3>
            <p>
                如果您输入的邮件地址所对应的账户存在，设置密码的提示已经发送
                邮件给您，您将很快收到。
            </p>
            <p>
                如果您没有收到邮件，请确保您所输入的地址是正确的，并检查您的
                垃圾邮件文件夹。
            </p>
        </div>
    </div>
</div>
</body>
</html>
```

(4) 编写设置新密码页面模板

在接收到的重置密码邮件中有一个设置新密码的链接，点击该链接就会跳转到给账户设置新密码的页面，以使用户给已忘记密码的账户设置一个全新的密码。该页面渲染

的模板为 `password_reset_confirm.html`，因此再添加一个设置新密码页面的模板。首先在 `registration/` 下新建一个 `password_reset_confirm.html` 文件，写入表单代码（几乎和登录页面一样）。

【registration/password_reset_confirm.html】

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
  <meta charset="utf-8">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
  <title>设置新密码</title>
  <link rel="stylesheet"
href="https://unpkg.com/mobi.css/dist/mobi.min.css">
  <style>
    .errorlist {
      color: red;
    }
  </style>
</head>
<body>
<div class="flex-center">
  <div class="container">
    <div class="flex-center">
      <div class="unit-1-2 unit-1-on-mobile">
        <h1><a href="{% url 'blog:index' %}">Django
Blog</a></h1>
        <h3>设置新密码</h3>
        <form class="form" method="post">
          {% csrf_token %}
          {{ form.non_field_errors }}
          {% for field in form %}
            {{ field.label_tag }}
            {{ field }}
            {{ field.errors }}
            {% if field.help_text %}
              <p class="help text-small text-
muted">{{ field.help_text|safe }}</p>
            {% endif %}
          {% endfor %}
          <button type="submit" class="btn btn-primary btn-
block">提交</button>
        </form>
      </div>
    </div>
  </div>
</div>
</body>
```

```
</html>
```

(5) 编写设置新密码成功页面模板

用户在设置新密码页面输入新密码后，Django 会把用户跳转到设置新密码成功页面，该页面渲染的模板为 password_reset_complete.html，因此再添加一个设置新密码成功页面的模板：

【registration/password_reset_complete.html】

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
  <meta charset="utf-8">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
  <title>新密码设置成功</title>
  <link rel="stylesheet"
href="https://unpkg.com/mobi.css/dist/mobi.min.css">
</head>
<body>
<div class="flex-center">
  <div class="container">
    <div>
      <h1 class="logo"><a href="{% url 'blog:index' %}">Django
Blog</a></h1>
      <h3>新密码设置成功</h3>
      <p>
        你的口令已经设置。现在你可以继续进行登录。
      </p>
    </div>
  </div>
</div>
</body>
</html>
```

(6) 测试整个重置密码流程

重置密码的流程略微复杂一点，其整个过程为：**用户输入注册时邮箱，跳转到发送成功页面 → 系统发送激活链接邮件到用户邮箱 → 用户进入邮箱，点击激活链接跳转到设置新密码页面 → 用户设置新密码，跳转到设置成功页面。**下面就来看看整个过程。

1) 输入注册时邮箱

在登录页面点击找回密码的按钮，跳转到输入注册邮箱页面：

Django Auth Example

重置密码

电子邮件:

提交

2) 邮件发送成功

输入正确的邮箱地址后，系统将发送重置密码的邮件到终端：

Django Auth Example

密码重置链接已经发送

如果您输入的邮件地址所对应的账户存在，设置密码的提示已经发送邮件给您，您将很快收到。

如果你没有收到邮件，请确保您所输入的地址是正确的，并检查您的垃圾邮件文件夹。

在终端可以接收到如下的邮件内容：

你收到这封邮件是因为你请求重置你在网站 127.0.0.1:8000 上的用户账户密码。

请访问该页面并选择一个新密码：

<http://127.0.0.1:8000/users/reset/NA/4n8-64ab7ff92254d18c6b15/>

你的用户名，如果已忘记的话：zmrenwu

感谢使用我们的站点！

127.0.0.1:8000 团队

点击内容中的链接，将跳转到设置新密码的页面。

3) 设置新密码

在设置新密码的页面输入需要设置的新密码：

Django Auth Example

设置新密码

新密码:

- 你的密码不能与其他个人信息太相似。
- 你的密码必须包含至少 8 个字符。
- 你的密码不能是大家都爱用的常见密码。
- 你的密码不能全部为数字。

新密码确认:

提交

4) 新密码设置成功

点击提交后将跳转到新密码设置成功页面：

Django Auth Example

新密码设置成功

你的口令已经设置。现在你可以继续进行登录。

现在便可以用新设置的密码登录了。