Django Password Reset Tutorial

learndjango.com/tutorials/django-password-reset-tutorial

- By Will Vincent
- Mar 22, 2022
- 30 Comments

In this tutorial we'll add a password reset sequence to our Django application. This builds upon our previous work where we added Login & Logout pages and then a Signup page.

Complete source code can be found on Github if you get stuck along the way.

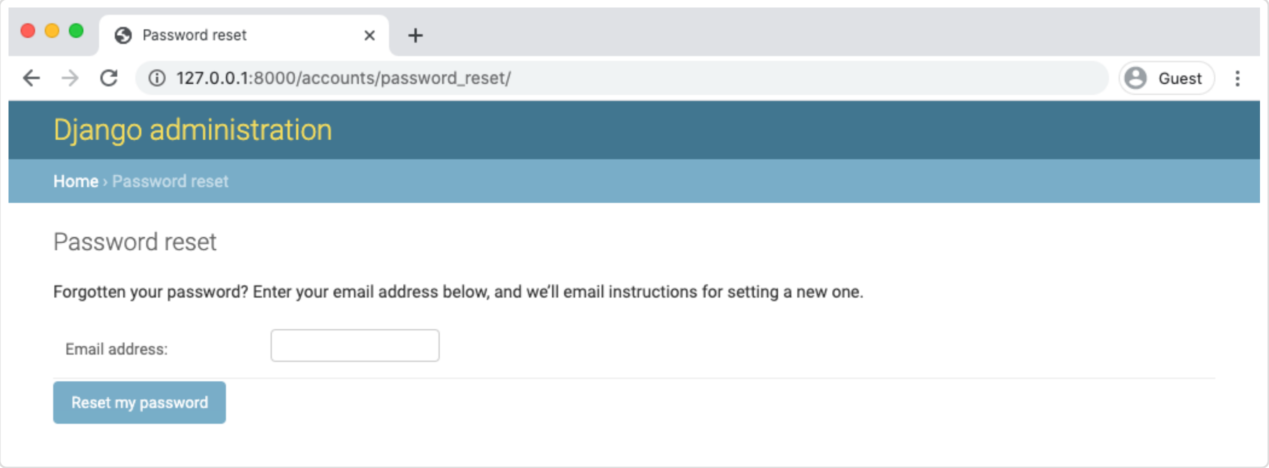
Django auth app

What we want is a `password_reset` page where the user can enter their email address, and be sent a cryptographically secure email with a one-time link to a `reset` page. Fortunately Django has us covered.

If you recall the complete set of views and URLs provided by the Django `auth` app, there are already several for resetting a password.

```
accounts/login/ [name='login']
accounts/logout/ [name='logout']
accounts/password_change/ [name='password_change']
accounts/password_change/done/ [name='password_change_done']
accounts/password_reset/ [name='password_reset']
accounts/password_reset/done/ [name='password_reset_done']
accounts/reset/<uidb64>/<token>/ [name='password_reset_confirm']
accounts/reset/done/ [name='password_reset_complete']
```

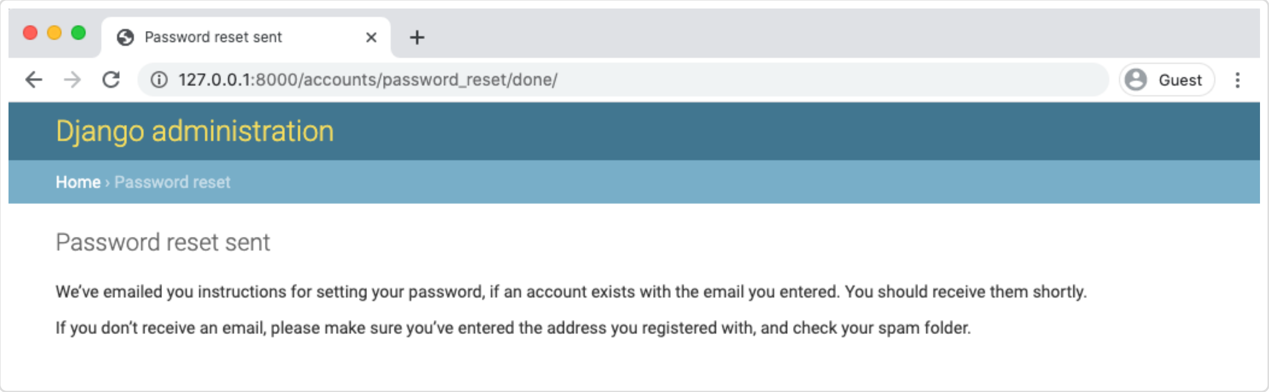
The default templates however are quite ugly and we need to customize them. For example, here are the default password reset and password reset done pages:



Password reset

Forgotten your password? Enter your email address below, and we'll email instructions for setting a new one.

Email address:



Django administration

Home › Password reset

Password reset sent

We’ve emailed you instructions for setting your password, if an account exists with the email you entered. You should receive them shortly.

If you don’t receive an email, please make sure you’ve entered the address you registered with, and check your spam folder.

But first we need to setup a way to deliver, or at least fake deliver, our email messages.

SMTP Server

In the real-world you would integrate with an email service like MailGun or SendGrid. For development purposes Django lets us store emails either in the console or as a file. We'll choose the latter and store all sent emails in a folder called `sent_emails` in our project directory.

To configure this, update our `django_project/settings.py` file by adding the following two lines at the bottom under our redirect URLs.

```
# django_project/settings.py
EMAIL_BACKEND = "django.core.mail.backends.filebased.EmailBackend"
EMAIL_FILE_PATH = BASE_DIR / "sent_emails"
```

Now let's change the appearance of the password reset pages.

Password Reset Form

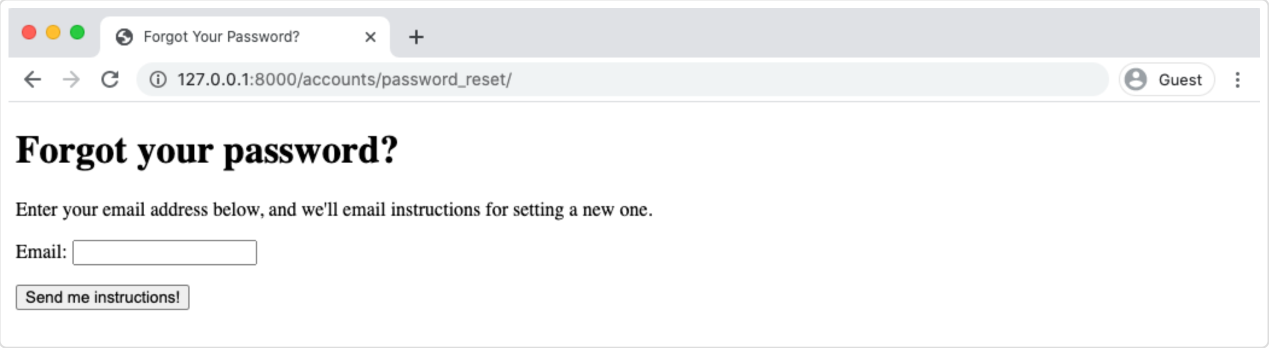
The default template for password reset is located at `templates/registration/password_reset_form.html`. We can customize it by creating our own `password_reset_form.html` file:

```
(.venv) $ touch templates/registration/password_reset_form.html
```

Then add the following code:

```
<!-- templates/registration/password_reset_form.html -->
{% extends 'base.html' %}

{% block title %}Forgot Your Password?{% endblock %}

{% block content %}
  <h1>Forgot your password?</h1>
  <p>Enter your email address below, and we'll email instructions for setting a
new one.</p>

  <form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Send me instructions!">
  </form>
{% endblock %}
```

If you refresh the page at `http://127.0.0.1:8000/accounts/password_reset/` you can see our new update:



Forgot your password?

Enter your email address below, and we'll email instructions for setting a new one.

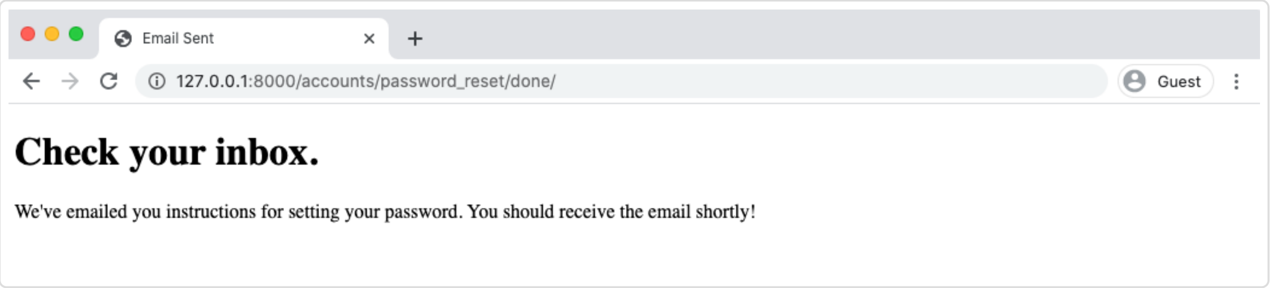
Email:

Now go ahead and enter the email address that matches an actual user you've created. Then click on the button to submit it.

Upon successful submission, we're redirected to the *Password reset done* page which is also ugly. Let's change it. The default template is located at `templates/registration/password_reset_done.html`. So as before, in your text editor create a new template file `templates/registration/password_reset_done.html` and add the following code:

```
<!-- templates/registration/password_reset_done.html -->
{% extends "base.html" %}

{% block title %}Email Sent{% endblock %}

{% block content %}
  <h1>Check your inbox.</h1>
  <p>We've emailed you instructions for setting your password. You should receive
the email shortly!</p>
{% endblock %}
```

If you refresh the password reset done page at `http://127.0.0.1:8000/accounts/password_reset/done/` we can see our new page.



Check your inbox.

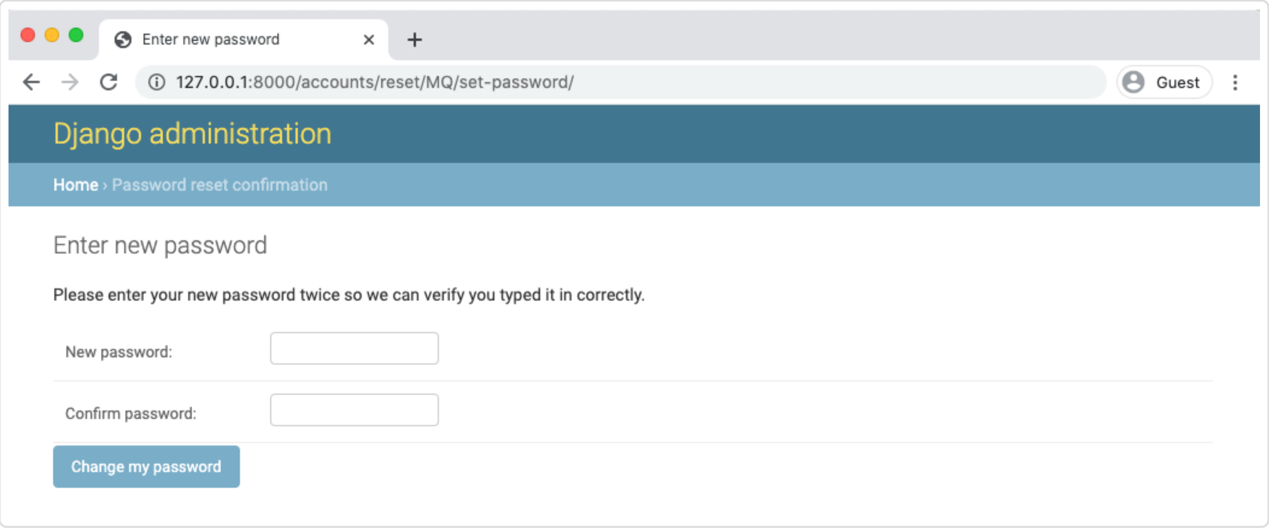
We've emailed you instructions for setting your password. You should receive the email shortly!

Password Reset Confirm

Remember how we configured our Django project to store emails in a local folder called `sent_emails`? If you look at your project now that folder exists! The format for the `txt` file will look something like this:

```
Content-Type: text/plain; charset="utf-8"
MIME-Version: 1.0
Content-Transfer-Encoding: 8bit
Subject: Password reset on 127.0.0.1:8000
From: webmaster@localhost
To: will@learndjango.com
Date: Fri, 18 Mar 2022 19:20:44 -0000
Message-ID:

<159985204410.36328.18347848425987829694@1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.



You're receiving this email because you requested a password reset for your user
account at 127.0.0.1:8000.

Please go to the following page and choose a new password:

http://127.0.0.1:8000/accounts/reset/MQ/aa1v2k-8ab2c9597a4f6cc754e3dc5baaf3c77f/

Your username, in case you’ve forgotten: wsv

Thanks for using our site!

The 127.0.0.1:8000 team
```

This contains Django's default language which we can customize. But the important section for now is the URL included. In the email above, mine is `http://127.0.0.1:8000/accounts/reset/MQ/aa1v2k-8ab2c9597a4f6cc754e3dc5baaf3c77f/`. Copy and paste yours into your browser and you'll be automatically routed to the *Password reset confirmation* page.


Enter new password
127.0.0.1:8000/accounts/reset/MQ/set-password/
Guest
Django administration
Home › Password reset confirmation

Enter new password

Please enter your new password twice so we can verify you typed it in correctly.

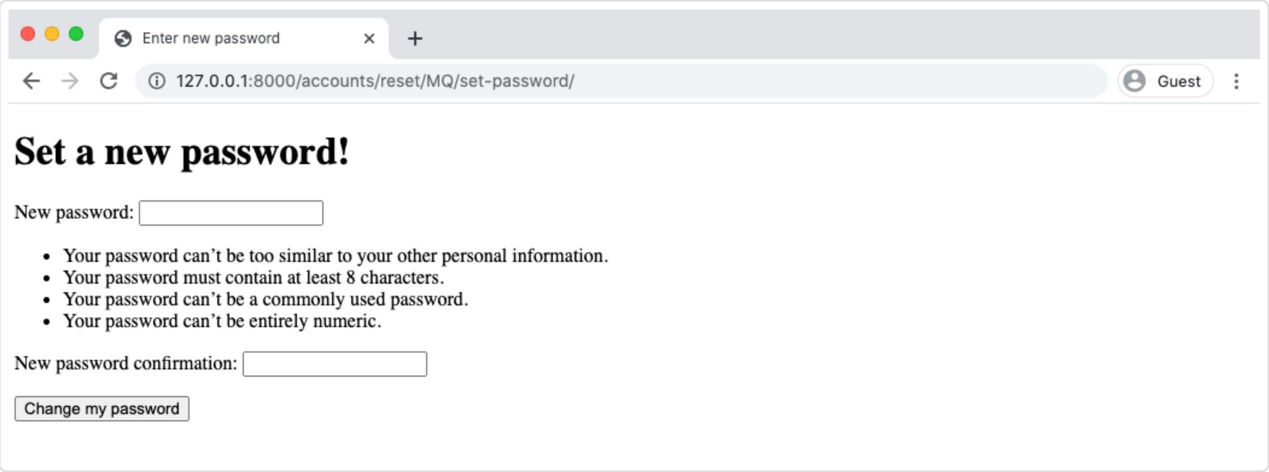
New password:

Confirm password:

Ugly, no? Let's create a new template with our familiar steps. In your text editor create the new template called `templates/registration/password_reset_confirm.html` and enter this new code:

```
<!-- templates/registration/password_reset_confirm.html -->
{% extends "base.html" %}

{% block title %}Enter new password{% endblock %}

{% block content %}

{% if validlink %}

<h1>Set a new password!</h1>
<form method="POST">
  {% csrf_token %}
  {{ form.as_p }}
  <input type="submit" value="Change my password">
</form>

{% else %}

<p>The password reset link was invalid, possibly because it has already been used.
Please request a new password reset.</p>

{% endif %}
{% endblock %}
```

Refresh the page at `http://127.0.0.1:8000/accounts/reset/Mg/set-password/` and you'll see our new template.



Set a new password!

New password:

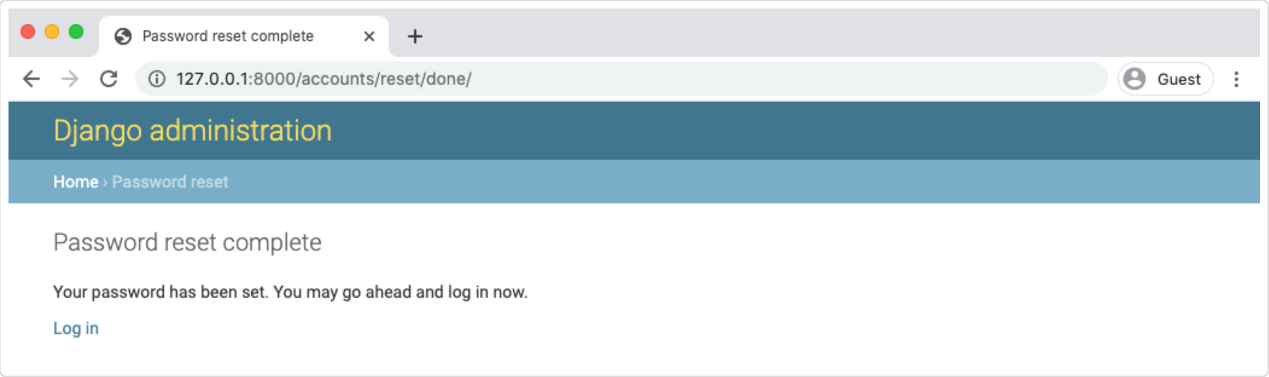
- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

New password confirmation:

Change my password

Password Reset Done

Go ahead and create a new password in our form. Upon submission you'll be redirected to our final default page which is for *Password reset complete*:



Password reset complete

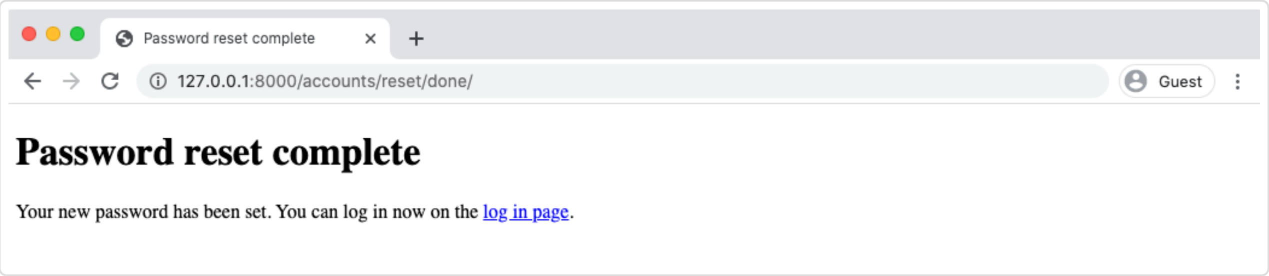
Your password has been set. You may go ahead and log in now.

Log in

To customize this page we'll create a new file called `password_reset_complete.html` and enter the following code:

```
<!-- templates/registration/password_reset_complete.html -->
{% extends 'base.html' %}

{% block title %}Password reset complete{% endblock %}

{% block content %}
<h1>Password reset complete</h1>
<p>Your new password has been set. You can log in now on the <a href="{% url
'login' %}">log in page</a>.</p>
{% endblock %}
```

Now reset the page at `http://127.0.0.1:8000/accounts/reset/done/` and view our work.



Password reset complete

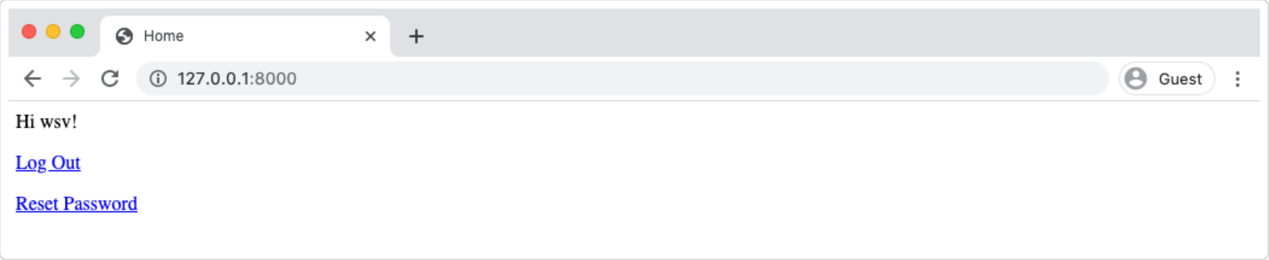
Your new password has been set. You can log in now on the log in page.

Add to home page

Let's add the password reset link to the homepage now so that logged-in users will see it. We can use the built-in tag `{% url 'password_reset' %}.` Here's the code.

```
<!-- templates/home.html -->
{% extends 'base.html' %}

{% block title %}Home{% endblock %}

{% block content %}
{% if user.is_authenticated %}
  Hi {{ user.username }}!
  <p><a href="{% url 'logout' %}">Log Out</a></p>
  <p><a href="{% url 'password_reset' %}">Reset Password</a></p>
{% else %}
  <p>You are not logged in</p>
  <a href="{% url 'login' %}">Log In</a>
{% endif %}
{% endblock %}
```

Navigate to `http://127.0.0.1:8000/` and log in using the "Log In" link. The updated homepage has both the "Log Out" and "Reset Password" links.



Hi wsv!

Log Out

Reset Password

Conclusion

We've now implemented a robust user authentication flow for our web app with login, logout, signup, and password reset. Congrats!

What else might we want? Using a custom user model instead of the built-in `User` model is a good idea. The Django docs note one is **highly recommended**. You could also add a "Password Change" feature, wire up emails to send, or even install the 3rd party django-allauth package to enable login via email only and other goodies.

I cover all of this--and more--in my book, Django for Beginners.