



この概念を理解した上で実装すると戸惑わないと思います。それでは標準ビューに4つについてそれぞれ説明していきます。

PasswordResetViewについて

パスワードリセットを開始するため、メールを発信するビューです。django.contrib.auth.views.PasswordResetViewには以下のクラス変数があり、この値を継承クラスでオーバーライドしたり、as_viewメソッドで設定することで制御することが出来ます。

- email_template_name: Emailテンプレート。デフォルトは'registration/password_reset_email.html'
- extra_email_context: Eメールテンプレートに渡す追加コンテキスト。デフォルトはNone
- form_class: デフォルトはPasswordResetForm
- from_email: Emailのfromアドレス。デフォルトはNone
- html_email_template_name = HTML用のEmailテンプレート。デフォルトはNone
- subject_template_name: メールタイトル用テンプレート。デフォルトは'registration/password_reset_subject.txt'
- success_url: デフォルトはreverse_lazy('password_reset_done')
- template_name: デフォルトは'registration/password_reset_form.html'
- title = デフォルトは'Password reset'の翻訳
- token_generator = デフォルトはdefault_token_generator

今回はすべてデフォルトの状態を使ってみます。認証機能のカスタマイズは本章の後半で扱うので、それまでお待ち下さい。

PasswordResetDoneViewについて

メール送信後はPasswordResetDoneViewが表示されます。django.contrib.auth.views.PasswordResetDoneViewには以下のクラス変数があり、この値を継承クラスでオーバーライドしたり、as_viewメソッドで設定することで制御することが出来ます。

- template_name: デフォルトは'registration/password_reset_done.html'
- title: デフォルトは'Password reset sent'の翻訳（言語がjaの場合は日本語訳）

このビューはメール送信したことをユーザーに伝えるだけの画面であり、ほぼ機能はTemplateViewと同様と考えれば良いと思います。

PasswordResetConfirmViewについて

送信されたメールに記載されているURLからアクセスした場合にPasswordResetConfirmViewが表示されます。
django.contrib.auth.views.PasswordResetConfirmViewには以下のクラス変数があり、この値を継承クラスでオーバーライドしたり、as_viewメソッドで設定することで制御することが出来ます。

- form_class: デフォルトはSetPasswordForm
- post_reset_login: デフォルトはFalse
- post_reset_login_backend: デフォルトはNone
- success_url: デフォルトはreverse_lazy('password_reset_complete')
- template_name: デフォルトは'registration/password_reset_confirm.html'
- title: デフォルトは'Enter new password'の翻訳
- token_generator: デフォルトはdefault_token_generator

このビューに関しても今回はカスタマイズせずそのまま使います。post_reset_loginについて簡単に解説しておきます。この値は新パスワードを登録した時にログインをするかどうかです。この値はデフォルトではFalseですが、Trueにしておくで新パスワードでログインしリダイレクトします。

PasswordResetCompleteViewについて

パスワードのリセット処理が無事に終了した場合にPasswordResetCompleteViewが表示されます。
django.contrib.auth.views.PasswordResetCompleteViewには以下のクラス変数があり、この値を継承クラスでオーバーライドしたり、as_viewメソッドで設定することで制御することが出来ます。

- template_name: デフォルトは'registration/password_reset_complete.html'
- title: デフォルトは'Password reset complete'の翻訳

このビューはあくまでリセットが完了したことを伝える画面ですので、PasswordResetConfirmViewでsuccess_urlを別画面に指定した場合はなくとも問題ありません。

パスワードリセット画面用のテンプレート作成

ではパスワードリセット画面用のテンプレートを作成していきます。django.contrib.auth.views.PasswordChangeViewのデフォルトテンプレートは'registration/password_change_form.html'ですので、その名前で作成していきます。

templates/registration/password_reset_form.html

```
{% extends 'base/base.html' %}
{% block title %}ITについて切磋琢磨する掲示板 - {{ block.super }}{% endblock %}
{% block content %}
<div class="ui grid stackable">
  <div class="eleven wide column">
    <div class="ui breadcrumb">
      <a class="section" href="{% url 'base:top' %}">TOP</a>
      <i class="right angle icon divider"></i>
      <div class="active section">パスワードリセット</div>
    </div>
    <div class="ui segment">
      <div class="content">
        <div class="header"><h3>パスワードリセット</h3></div>
        <form class="ui form" method="POST" action="">
          {% csrf_token %}
          {{form.as_p}}
          <button type="submit" class="ui orange button">メールを送信する</button>
        </form>
      </div>
    </div>
  </div>
  {% include 'base/sidebar.html' %}
</div>
{% endblock %}
```

次にPasswordResetDoneViewのテンプレートを準備していきます。

templates/registration/password_reset_done.html

```

{% extends 'base/base.html' %}
{% block title %}ITについて切磋琢磨する掲示板 - {{ block.super }}{% endblock %}
{% block content %}
<div class="ui grid stackable">
  <div class="eleven wide column">
    <div class="ui breadcrumb">
      <a class="section" href="{% url 'base:top' %}">TOP</a>
      <i class="right angle icon divider"></i>
      <div class="active section">パスワード変更</div>
    </div>
    <div class="ui segment">
      <div class="content">
        <div class="content">
          <div class="header"><h3>パスワードリセット</h3></div>
          <p>パスワードリセット手続きのためメールを送信しました。メールを確認の上、リセット手続きを行って下さい。</p>
        </div>
      </div>
    </div>
  </div>
  {% include 'base/sidebar.html' %}
</div>
{% endblock %}

```

次にPasswordResetConfirmView用のテンプレートを作成していきます。

templates/registration/password_reset_confirm.html

```

{% extends 'base/base.html' %}
{% block title %}ITについて切磋琢磨する掲示板 - {{ block.super }}{% endblock %}
{% block content %}
<div class="ui grid stackable">
  <div class="eleven wide column">
    <div class="ui breadcrumb">
      <a class="section" href="{% url 'base:top' %}">TOP</a>
      <i class="right angle icon divider"></i>
      <div class="active section">パスワードリセット</div>
    </div>
    <div class="ui segment">
      <div class="content">
        <div class="header"><h3>パスワードリセット</h3></div>
        <form class="ui form" method="POST" action="">
          {% csrf_token %}
          {{form.as_p}}
          <button type="submit" class="ui orange button">変更する</button>
        </form>
      </div>
    </div>
  </div>
  {% include 'base/sidebar.html' %}
</div>
{% endblock %}

```

次にPasswordResetConfirmView用のテンプレートを作成していきます。

templates/registration/password_reset_complete.html

```
{% extends 'base/base.html' %}
{% block title %}ITについて切磋琢磨する掲示板 - {{ block.super }}{% endblock %}
{% block content %}
<div class="ui grid stackable">
  <div class="eleven wide column">
    <div class="ui breadcrumb">
      <a class="section" href="{% url 'base:top' %}">TOP</a>
      <i class="right angle icon divider"></i>
      <div class="active section">パスワードリセット完了</div>
    </div>
    <div class="ui segment">
      <div class="content">
        <div class="content">
          <div class="header"><h3>パスワードリセットが完了しました</h3></div>
          <a href="{% url 'base:top' %}">TOPに戻る</a>
        </div>
      </div>
    </div>
  </div>
  {% include 'base/sidebar.html' %}
</div>
{% endblock %}
```

URLリンクの修正

ログイン画面のリンクも修正しておきましょう。
templates/registration/login.html(一部抜粋)

```
<a class="ui item" href="{% url 'accounts:create' %}">ユーザー登録</a>／
- <a class="ui item" href="">パスワードを忘れた場合</a>
+ <a class="ui item" href="{% url 'password_reset' %}">パスワードを忘れた場合</a>
```

urlのショートカットが'password_reset'であることに注意して下さい。

動作確認

では動作を確認してみましょう。プロフィール画面から「パスワード変更」を押してパスワード変更画面を表示パスワードを変更してみます。パスワード変更成功の画面が出ればOKです。念の為パスワードが変更されているか再度ログインして確かめてみましょう。

※動作確認前に必ず[LoginViewを使用してログイン画面を作成する](#)で紹介した要領でmysite/urls.pyの変更をしておいて下さい。

パスワードリセット開始

[TOP](#) > パスワードリセット

パスワードリセット

メールアドレス:

user1@example.com

メールを送信する

メール送信済みの画面表示

[TOP](#) > パスワード変更

パスワードリセット

パスワードリセット手続きのためメールを送信しました。メールを確認の上、リセット手続きを行って下さい。

送信されたメールを確認

このメールは mysite で、あなたのアカウントのパスワードリセットが要求されたため、送信されました。

次のページで新しいパスワードを選んでください:

<http://mydomain.com/accounts/reset/NA/55b-e6a3b38d787906755e98/>

あなたのユーザー名 (念のため): user1

ご利用ありがとうございました！

mysite チーム

送信されたメールからパスワードリセット画面にアクセスしパスワード変更

[TOP](#) > [パスワードリセット](#)

パスワードリセット

新しいパスワード:

.....

- あなたの他の個人情報と似ているパスワードにはできません。
- パスワードは最低 8 文字以上必要です。
- よく使われるパスワードにはできません。
- 数字だけのパスワードにはできません。

新しいパスワード(確認用):

.....

変更する

パスワードリセット完了画面

[TOP](#) > [パスワードリセット完了](#)

パスワードリセットが完了しました

[TOPに戻る](#)

メールテンプレートの作成

さて、送信されたメールの文言どうでしょうか？ ちょっとこのまま使うのは難しい気がしますよね。ではメールテンプレートを変更していきましょう。

templates/registration/password_reset_complete.html

```
{{ user.username }} 様
```

下記URLよりサイトにアクセスの上、パスワードの再設定を行ってください。

再設定用URL

```
{{ protocol }}://{{ domain }}{% url 'password_reset_confirm' uidb64=uid token=token %}
```

本メールは{{protocol}}://{{domain}}より自動送信されています。
心当たりのない場合は破棄をお願いします。

あと、メールタイトルも変更しておきましょう。これも組込みテンプレートと同名のファイルを作成することで解決します。拡張子がtxtなことに注意してくださいね。

templates/registration/password_reset_subject.txt

Copy

IT学習ちゃんねるのパスワードリセットのお知らせ

これでパスワードリセット処理時に送信されるメールは以下のようになります。

user1 様

下記URLよりサイトにアクセスの上、パスワードの再設定を行ってください。

再設定用URL

<http://mydomain.com/accounts/reset/NA/55d-f430cc6402285a9bb42a/>

本メールは<http://mydomain.com>より自動送信されています。

心当たりのない場合は破棄をお願いします。