

# TP11 – Administration du site – connexion (2/2)

## Contexte et Objectifs

Nous allons maintenant aborder les dernières fonctionnalités classiques du système de connexion : inscription et renouvellement du mot de passe oublié.

Les templates associés à ces fonctionnalités sont classiques :

- `register.html` pour l'inscription,
- `password_reset.html` pour le formulaire à présenter en cas de mot de passe oublié,
- `password_reset_done.html`, pour la redirection après envoi du formulaire précédent,
- `password_reset_email.html`, qui affiche le message, avec le lien de réinitialisation du mot de passe,
- `password_reset_confirm.html`, page qu'on ouvre avec le lien, pour y saisir le nouveau mot de passe,
- `password_reset_complete.html`, page où on redirige l'utilisateur après changement du mot de passe.

## L'inscription

L'inscription est la fonctionnalité la plus simple. Elle comprend cinq étapes :

- la création du formulaire d'inscription
- la création du path pour le template `register.html`
- la création du template `registr.html`
- la création de la view `register`
- la modification du lien d'inscription dans `login.html`

1. Créez le fichier forms.py de l'application connexion. L'objectif est d'y coder le formulaire d'inscription :

```
webpizza > connexion > forms.py > ...
1 from django import forms
2 from django.contrib.auth.models import User
3 from django.contrib.auth.forms import UserCreationForm
4
5 class UserRegistrationForm(UserCreationForm):
6     first_name = forms.CharField(label='Prénom')
7     last_name = forms.CharField(label='Nom')
8     email = forms.EmailField(label='Adresse e-mail')
9
10    class Meta(UserCreationForm.Meta):
11        model = User
12        fields = UserCreationForm.Meta.fields + ('first_name', 'last_name', 'email')
```

2. ajoutez le path nécessaire à l'appel du futur template register.html

```
urlpatterns = [
    path('login/', auth_views.LoginView.as_view(template_name='connexion/login.html'), name = 'login'),
    path('logout/', auth_views.LogoutView.as_view(template_name='connexion/logout.html'), name = 'logout'),
    path('register/', views.register, name='register'),
]
```

3. créez maintenant le template register.html avec le code du formulaire pré-conçu :

```
webpizza > connexion > templates > connexion > register.html > ...
1 {% extends 'connexion/base.html' %}
2 {% block titre %}inscription{% endblock %}
3 {% block contenu %}
4     <h2>S'inscrire</h2>
5     <form id="form_login" method="post">
6         {% csrf_token %}
7         <table>
8             {% for field in form %}
9                 <tr>
10                     <td>{{ field.label_tag }}</td>
11                     <td>{{ field }}</td>
12                 </tr>
13                 {% for error in field.errors %}
14                     <tr><td></td><td class="alert alert-danger">{{ error }}</td></tr>
15                 {% endfor %}
16             {% endfor %}
17         </table>
18         <button type="submit">s'inscrire</button>
19     </form>
20     <div>
21         <small>
22             Déjà inscrit ?
23             <a href="{% url 'login' %}">Se connecter</a>
24         </small>
25     </div>
26 {% endblock %}
```

4. créez enfin, dans le fichier views.py de connexion, la view register :

```
webpizza > connexion > views.py > ...
1  from django.shortcuts import render, redirect
2  from connexion.forms import UserRegistrationForm
3  from django.contrib.auth import login, authenticate
4  from django.contrib import messages
5
6  # Create your views here.
7  def register(request):
8      if request.method == 'POST':
9          form = UserRegistrationForm(request.POST)
10         if form.is_valid():
11             form.save()
12             username = form.cleaned_data.get('username')
13             password = form.cleaned_data.get('password1')
14             user = authenticate(username=username, password=password)
15             login(request, user)
16             messages.success(request, f'Coucou {username}, Votre compte a été créé avec succès !')
17             return redirect('pizzas')
18         else :
19             form = UserRegistrationForm()
20             return render(request, 'connexion/register.html', {'form' : form})
```

5. dans le template login.html, complétez le lien actuellement vide pour l'inscription :

```
<a href="{% url 'register' %}">
```

6. créez un nouvel utilisateur, connectez-vous par cet utilisateur et vérifiez que tout fonctionne.

## Le renouvellement du mot de passe oublié

Les étapes pour le renouvellement sont plus nombreuses et détaillées. Mais le gros du travail est de créer les path et les templates. Pas besoin de view supplémentaire.

1. Dans le template login.html, renseignez le lien pour renouveler le mot de passe :

```
Mot de passe oublié ?
<a href="{% url 'password_reset' %}">
    Renouveler mon mot de passe
</a>
```

2. créez le nouveau path pour password\_reset.html :

```
urlpatterns = [
    path('login/', auth_views.LoginView.as_view(template_name='connexion/login.html'), name = 'login'),
    path('logout/', auth_views.LogoutView.as_view(template_name='connexion/logout.html'), name = 'logout'),
    path('register/', views.register, name='register'),
    path('password_reset/',
        auth_views.PasswordResetView.as_view(
            template_name='connexion/password_reset.html',
            email_template_name='connexion/password_reset_email.html'),
        name = 'password_reset'),
]
```

3. créez le template password\_reset.html :

```
webpizza > connexion > templates > connexion > <> password_reset.html > ...
1  {% extends 'connexion/base.html' %}
2  {% block contenu %}
3  <h2>Réinitialisation du mot de passe </h2>
4  <p>Mot de passe perdu ?</p>
5  <p>Saisissez votre adresse électronique dans le formulaire</p>
6  <p>Nous vous enverrons les instructions pour en créer un nouveau.</p>
7  <form id="form_login" method="post">
8      {% csrf_token %}
9      {% if form.email.errors %}
10         {% for error in form.email.errors %}
11             <p class="alert alert-danger">{{ error }}</p>
12         {% endfor %}
13     {% endif %}
14     <p>{{ form.email.label_tag }} {{ form.email }}</p>
15     <button type="submit">envoyer instructions</button>
16 </form>
17 {% endblock %}
```

4. créez maintenant le template couplé avec le précédent, qui fournit les instructions (elle seront envoyées sur la console pour ne pas gérer les problèmes liées à l'envoi d'email) :

```
webpizza > connexion > templates > connexion > <> password_reset_email.html
1  Bonjour, Vous avez demandé la réinitialisation de votre
2  mot de passe en indiquant ce courriel {{ email }}.
3  Cliquez sur le lien ci-dessous :
4  {{ protocol }}://{{ domain }}{% url 'password_reset_confirm' uidb64=uid token=token %}
```

5. Pour que ce message s'affiche bien dans la console (ce qui permet de gérer de manière fictive les envois d'email), précisez dans le fichier settings.py :

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

6. ce template évoque un template password\_reset\_confirm.html, il faut donc le coder avant de tester. Ce template est assez long, car il va demander de saisir deux fois le nouveau mot de passe, et envisager les cas d'erreur :

```
webpizza > connexion > templates > connexion > <> password_reset_confirm.html > ...
1  {% extends 'connexion/base.html' %}
2  {% block contenu %}
3  {% if validlink %}
4  <h2>Réinitialisation de mot de passe </h2>
5  <p>Veuillez saisir votre nouveau mot de passe.</p>
6  <form method="post">
7      {% csrf_token %}
8      {% if form.new_password1.errors %}
9          <ul class="alert alert-danger">
10             {% for error in form.new_password1.errors %}
11                 <li>{{ error }}</li>
12             {% endfor %}
13         </ul>
14     {% endif %}
15     {% if form.new_password2.errors %}
16         <ul class="alert alert-danger">
17             {% for error in form.new_password2.errors %}
18                 <li>{{ error }}</li>
19             {% endfor %}
20         </ul>
21     {% endif %}
22     <table>
23         <tr>
24             <td>
25                 <label for="id_new_password1">nouveau mot de passe :</label>
26             </td>
27             <td>{{ form.new_password1 }}</td>
28         </tr>
29         <tr>
30             <td>
31                 <label for="id_new_password2">Confirmer nouveau mot de passe :</label>
32             </td>
33             <td>{{ form.new_password2 }}</td>
34         </tr>
35     </table>
36     <button type="submit">Changer mon mot de passe</button>
37 </form>
38 {% else %}
39 <h2>Echec de réinitialisation du mot de passe</h2>
40 <p>Le lien de réinitialisation de mot de passe est invalide. Veuillez réessayer.</p>
41 {% endif %}
42 {% endblock %}
```

7. il faut maintenant définir le path pour ce template :

```
path(
    'reset/<uidb64>/<token>/',
    auth_views.PasswordResetConfirmView.as_view(
        template_name='connexion/password_reset_confirm.html'),
    name='password_reset_confirm'),
```



8. encore deux étapes : il faut maintenant le template `password_reset_done.html`, comme ci-dessous :

```
webpizza > connexion > templates > connexion > <> password_reset_done.html > ...
1  {% extends 'connexion/base.html' %}
2  {% block contenu %}
3  <h2>Message de réinitialisation du mot de passe envoyé</h2>
4  <p>Nous vous avons envoyé par courriel les instructions pour changer de mot de passe,</p>
5  <p>pour autant qu'un compte existe avec l'adresse que vous avez indiquée.</p>
6  <p>Vous devriez recevoir rapidement ce message.</p>
7  <p>Si vous ne recevez pas de message, vérifiez que vous avez saisi l'adresse avec laquelle</p>
8  <p>vous vous êtes enregistré et contrôlez votre dossier de spams.</p>
9  {% endblock %}
```

```
path(
    'password_reset/done/',
    auth_views.PasswordResetDoneView.as_view(
        template_name='connexion/password_reset_done.html'),
    name='password_reset_done'),
```

9. Enfin (!!!), le template `password_reset_complete.html`

```
webpizza > connexion > templates > connexion > <> password_reset_complete.html > ...
1  {% extends 'connexion/base.html' %}
2  {% block contenu %}
3  <h2>Mot de passe réinitialisé.</h2>
4  <p>Votre mot de passe a été réinitialisé avec succès.</p>
5  <p>Revenir à <a href="{% url 'login' %}">la page de connexion</a>.</p>
6  {% endblock %}
```

et son path :

```
path(
    'reset/done/',
    auth_views.PasswordResetCompleteView.as_view(
        template_name='connexion/password_reset_complete.html'),
    name='password_reset_complete'),|
```

Testez que tout fonctionne !

Il est malheureusement possible que des erreurs se soient glissées dans les pages du TP.

Au besoin, s'inspirer de ma source :

<https://lasteminista.com/comment-implementer-un-systeme-de-connexion-authentification-django-1-partie/>

<https://lasteminista.com/reinitialisation-du-mot-de-passe-authentification-django-2-partie/>