

TP14 – inscription, renouvellement du mot de passe

Introduction, Contexte et Objectifs

Nous allons maintenant aborder les dernières fonctionnalités classiques du système de connexion : inscription et renouvellement du mot de passe oublié. Comme pour la connexion et la déconnexion (login et logout), il existe un contexte tout fait dans django. Mais nous allons le personnaliser pour bien le comprendre, et pour y utiliser bootstrap.

Les templates associés à ces fonctionnalités sont classiques. Nous garderons les noms par défaut de django, mais nous adapterons le contenu :

- register.html pour le formulaire d'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 au site

Le lien pour l'inscription, qui existe déjà dans le template login.html (`href="/register/"`) n'est pas encore opérationnel.

- Créez d'abord l'url relative à la view register. Ce path mènera à une view formulaireInscription :

```
path('register/', views.formulaireInscription),
```

- Créez la view formulaireInscription de l'applicompte, qui ne fait qu'afficher le template (pas encore créé, c'est pour après) formulaireInscription.html.
- Créez les template formulaireInscription.html qui propose les champs suivants (avec ici des classes bootstrap). Vous noterez le champ image pour la photo de profil :

```
webpizza > applicompte > templates > applicompte > register.html > ...
1  {% extends 'applipizza/base.html' %}
2  {% block title %} inscription {% endblock %}
3  {% block main %}
4  <h2>Remplissez notre formulaire d'inscription</h2>
5  <form enctype="multipart/form-data" id="form_register" method="post" action="/inscription/">
6    {% csrf_token %}
7    <div class="mb-3">
8      <label for="id_username"> identifiant </label>
9      <input type="text" class="form-control" id="id_username" name="username" placeholder="votre identifiant">
10   </div>
11   <div class="mb-3">
12     <label for="first_name"> prénom </label>
13     <input type="text" class="form-control" id="first_name" name="first_name" placeholder="votre prénom">
14   </div>
15   <div class="mb-3">
16     <label for="last_name"> nom </label>
17     <input type="text" class="form-control" id="last_name" name="last_name" placeholder="votre nom">
18   </div>
19   <div class="mb-3">
20     <label for="email"> email </label>
21     <input type="email" class="form-control" id="email" name="email" placeholder="votre email">
22   </div>
23   <div class="mb-3">
24     <label for="id_password"> mot de passe </label>
25     <input type="password" class="form-control" id="id_password" name="password" placeholder="votre mot de passe">
26   </div>
27   <div class="mb-3">
28     <label for="id_image" class="form-label">votre photo</label>
29     <input class="form-control" name="image" type="file" accept="image/*" id="id_image">
30   </div>
31   <button type="submit" class="btn btn-primary">s'inscrire</button>
32 </form>
33 {% endblock %}
```

- Ce formulaire contient un lien (`action="/inscription/"`) ce qui signifie que :

- on doit créer un path du type

```
path('inscription/', views.traitemetFormulaireInscription),
```

b. on doit créer une view traitementFormulaireInscription qui va d'abord traiter les données envoyées pas le formulaire :

- on récupère les champs de request.POST et request.FILES
- on crée un PizzaUser
- on lui affecte en attributs les champs récupérés
- on sauvegarde ce PizzaUser

Puis cette view va :

- connecter ce PizzaUser, grâce à la méthode login de django.contrib.auth
- récupérer les pizzas
- retourner un contexte appelant le template pizzas.html, avec la liste des pizzas et le PizzaUser.

```

36  > def formulaireInscription(request) : ...
41
42  def traitementFormulaireInscription(request) :
43      # récupération des champs du formulaire
44      fst = request.POST['first_name']
45      lst = request.POST['last_name']
46      usr = request.POST['username']
47      eml = request.POST['email']
48      pwd = request.POST['password']
49      img = request.FILES['image']
50
51      # création d'un PizzaUser
52      user = PizzaUser()
53
54      # affectation des champs récupérés aux attributs du PizzaUser
55      user.first_name = fst
56      user.last_name = lst
57      user.username = usr
58      user.email = eml
59      user.set_password(pwd)
60      user.image = img
61
62      # sauvegarde du PizzaUser dans la base de données
63      user.save()
64
65      # connexion du PizzaUser
66      login(request, user)
67
68      # récupération des pizzas
69      lesPizzas = Pizza.objects.all()
70
71      # on renvoie l'appel du template, avec les pizzas et le PizzaUser
72      return render(
73          request,
74          'applipizza/pizzas.html',
75          {"pizzas" : lesPizzas, "user" : user}
76      )

```

5.

Mettez tout ça en œuvre et testez une inscription d'un nouvel utilisateur.

Le renouvellement du mot de passe

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, on a déjà le lien pour le renouvellement du mot de passe :

```
Mot de passe oublié ?
Renouveler mon mot de passe
```

créez le nouveau path pour password_reset.html :

```
urlpatterns = [
    path('login/', auth_views.LoginView.as_view(
        template_name = 'applicompte/login.html'), name = 'login'),
    path('logout/', views.deconnexion, name = 'logout'),
    path('connexion/', views.connexion),
    path('user/update/', views.formulaireProfil),
    path('user/updated/', views.traitemenFormulaireProfil),
    path('register/', views.formulaireInscription),
    path('inscription/', views.traitemenFormulaireInscription),
    path('password_reset/', auth_views.PasswordResetView.as_view(
        template_name='applicompte/password_reset.html',
        email_template_name='applicompte/password_reset_email.html'), name='password_reset'),
]
```

2. créez ensuite le template password_reset.html :

```
webpizza > applicompte > templates > applicompte > password_reset.html > ...
1  {% extends 'applipizza/base.html' %} 
2  {% block title %} renouvellement du mot de passe {% endblock %} 
3  {% block main %} 
4  <div id="content">
5  <h2>mot de passe oublié ?</h2>
6  <p>
7  |   Saisissez votre adresse email dans le formulaire.<br>
8  |   Nous vous enverrons les instructions pour renouveler votre mot de passe.
9  </p>
10 <form id="form_password_reset" method="post">
11  |   {% csrf_token %}
12  |   <div class="mb-3">
13  |       |   <label for="id_email"> email </label>
14  |       |   <input type="email" class="form-control" id="id_email" name="email"
15  |       |   placeholder="votre email" autocomplete="" email" required>
16  |   </div>
17  |   <button type="submit" class="btn btn-primary">envoyer les instructions</button>
18 </form>
19 </div>
20 {% endblock %}
```

La particularité du précédent formulaire est qu'il va permettre l'appel du template couplé avec le précédent, qui fournit les instructions par une simulation d'email (les instructions seront envoyées sur la console pour ne pas gérer les problèmes liés à l'envoi d'email).

3. Créez maintenant le template password_reset_email.html dont le contenu est le suivant :

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

4. 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'
```

5. 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 :

```
password_reset_confirm.html ...
1 {% extends 'applicompte/base.html' %}
2 {% block title %} renouvellement du mot de passe {% endblock %}
3 {% block main %}
4 {% if validlink %}
5 <h2>renouvellement du mot de passe</h2>
6 <p>
7 | Veuillez saisir deux fois votre nouveau mot de passe.
8 </p>
9 <form id="form_password_reset" method="post">
10 | {% csrf_token %}
11 | <div class="mb-3">
12 | | <label for="id_new_password1"> nouveau mot de passe </label>
13 | | <input type="password" class="form-control" id="id_new_password1" name="new_password1" required>
14 | </div>
15 | <div class="mb-3">
16 | | <label for="id_new_password2"> confirmer mot de passe </label>
17 | | <input type="password" class="form-control" id="id_new_password2" name="new_password2" required>
18 | </div>
19 | <button type="submit" class="btn btn-primary">changer le mot de passe</button>
20 </form>
21 {% else %}
22 <h2>échec de réinitialisation du mot de passe</h2>
23 <p>Le lien de réinitialisation est invalide. Veuillez réessayer.</p>
24 {% endif %}
25 {% endblock %}
```

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

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

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

```
webpizza > applicompte > templates > applicompte > password_reset_done.html > ...
1  {% extends 'applipizza/base.html' %}
2  {% block title %} confirmation d'envoi d'instructions {% endblock %}
3  {% block main %}
4  <h2>message de réinitialisation du mot de passe envoyé</h2>
5  <p>
6      Nous vous avons envoyé par email les instructions pour changer de mot de passe,<br>
7      pour attaquer qu'un compte existe avec l'adresse que vous avez indiquée.<br>
8      Vous deviez rapidement recevoir ce message.<br>
9      Dans le cas contraire, contrôlez vos spams et vérifiez que l'adresse email mentionnée est<br>
10     bien votre adresse de contact.
11  </p>
12
13  {% endblock %}
```

puis le path :

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

8. Enfin (!!!), le template password_reset_complete.html

```
webpizza > applicompte > templates > applicompte > password_reset_complete.html > ...
1  {% extends 'applipizza/base.html' %}
2  {% load static %}
3  {% block title %} renouvellement effectué {% endblock %}
4  {% block main %}
5  <div class="div-logout">
6      <h2>Votre mot de passe a été réinitialisé.</h2>
7      
8      <a href="/login/"> connexion </a>
9  </div>
10 {% endblock %}
```

et son path :

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

9. Testez que tout fonctionne !