

Sécurité du framework Django

Pour garantir la sécurité de l'application du zoo Arcadia, nous avons pris des dispositions spécifiques concernant l'authentification, la gestion des formulaires, la protection contre les attaques CSRF et le hachage des mots de passe. Voici une explication détaillée de nos choix techniques.

Sécurité dans Django :

Authentification (Django Auth)

Django offre un système d'authentification intégré pour gérer les utilisateurs et les groupes.

Système d'Authentification:

Modèle User: Fournit un ensemble de champs tels que username, email, password, etc.

Groupes :

Permet d'organiser les utilisateurs avec des permissions spécifiques.

Permissions: Contrôle l'accès aux différentes fonctionnalités.

Formulaires d'Authentification:

Formulaire de Connexion :

python

```
from django.contrib.auth.forms import AuthenticationForm
```

```
form = AuthenticationForm(request, data=request.POST)
```

Formulaire d'Inscription :

python

```
from django.contrib.auth.forms import UserCreationForm
```

```
form = UserCreationForm()
```

Gestion des Mots de Passe

Hachage des Mots de Passe:

Django utilise un système de hachage basé sur PBKDF2 avec SHA256.

Le mot de passe est stocké sous la forme suivante:

shell

hasher\$salt\$hash

Exemple :

python

```
from django.contrib.auth.hashers import make_password, check_password
```

```
password = make_password('motdepasse123')
```

```
check = check_password('motdepasse123', password)
```

Protection CSRF

Django protège automatiquement les formulaires contre les attaques Cross-Site Request Forgery (CSRF).

Protection par Décorateur:

@csrf_exempt: Exempte une vue spécifique de la protection CSRF.

@csrf_protect: Applique explicitement la protection CSRF.

python

```
from django.views.decorators.csrf import csrf_exempt
```

```
@csrf_exempt
```

```
def vue_sans_csrf(request):
```

```
    return HttpResponse("CSRF exemptée")
```

Gestion des Permissions

Décorateurs :

@user_passes_test: Restreint l'accès à une vue si le test conditionnel échoue.

@login_required: Redirige vers la page de connexion si l'utilisateur n'est pas authentifié.

Exemples :

@login_required:

python

```
from django.contrib.auth.decorators import login_required
```

```
@login_required
```

```
def vue_protegee(request):  
    return HttpResponse("Vue accessible uniquement aux utilisateurs connectés")  
@user_passes_test:  
python  
from django.contrib.auth.decorators import user_passes_test  
  
def est_admin(user):  
    return user.is_staff  
  
@user_passes_test(est_admin)  
def vue_admin(request):  
    return HttpResponse("Vue accessible uniquement aux administrateurs")
```

En utilisant les fonctionnalités intégrées de Django pour l'authentification, la protection CSRF et le hachage des mots de passe, nous avons renforcé la sécurité de l'application du zoo Arcadia. L'utilisation judicieuse des décorateurs tels que `@csrf_exempt`, `@user_passes_test` et `@login_required` nous a permis d'assurer un contrôle d'accès sécurisé.