Programação de Sistemas para Internet

Prof. Diego Cirilo

Aula 11: Autorização

Autorização

- A autorização define os recursos que podem ser acessados pelo usuário logado;
- É possível desenvolver um sistema próprio de autorização por meio de atributos;
- O Django já tem o seu próprio sistema.

Autorização no Django

- Existem 3 conceitos principais no sistema de autorização do Django:
  - Usuários;
  - Grupos;
  - Permissões.

Usuários e Grupos

- Usuários como vistos na aula passada;
- Grupos permitem atribuir permissões para conjuntos de usuários;
- Um usuário pode ter permissões individuais e fazer parte de um ou mais grupos.

Permissões

- Regras de acesso a recursos;
- No Django as permissões são relativas a um *Model*;
- Para cada model, o Django cria automaticamente 4 permissões:
  - `add_nomedomodel`;
  - `change_nomedomodel`;
  - `delete_nomedomodel`;
  - `view_nomedomodel`;
- Também é possível criar outras permissões além das padrão.

Permissões customizadas

- Usamos uma classe Meta na definição do model:

```
class Livro(models.Model):
    titulo = models.CharField(max_length=100)
    autor = models.CharField(max_length=100)

    class Meta:
        permissions = [
            ("pode_publicar", "Pode publicar um livro"),
            ("pode_arquivar", "Pode arquivar um livro"),
        ]
```

Permissões

- Cada uma dessas permissões pode ser habilitada para cada usuário ou grupo;
- As permissões podem ser configuradas na interface de Admin ou em código.

Configurando Grupos/Permissões

- Criar grupo:

```
from django.contrib.auth.models import Group

grupo, sucesso = Group.objects.get_or_create(name="Nome do Grupo")
if sucesso:
    print(f"Sucesso!")
else:
    print(f"Falha!")
```

- Adicionar permissão a grupo:

```
from django.contrib.auth.models import Permission

permissao = Permission.objects.get(codename='change_nomedomodel')

grupo.permissions.add(permission)
```

Configurando Grupos/Permissões

- Adicionando usuário a grupo:

```
from django.contrib.auth.models import User # Ou o seu custom user

usuario = User.objects.get(id=1)

usuario.groups.add(grupo)
```

- Remover usuário de grupo:

```
usuario.groups.remove(grupo)
```

Configurando Grupos/Permissões

- Remover permissão de um grupo:

```
grupo.permissions.remove(permission)
```

- Remover um grupo:

```
grupo.delete()
```

Configurando Grupos/Permissões

- Adicionando permissões para um usuário:

```
usuario = User.objects.get(id=4)
permissao = Permission.objects.get(codename='nomedapermissao')
usuario.user_permissions.add(permissao)
```

- Remover permissão de usuário:

```
usuario.user_permissions.remove(permissao)
```

- Remover todas as permissões de usuário:

```
usuario.user_permissions.clear()
```

Verificando Permissões

- As permissões podem ser verificadas:
  - Usando o *decorator* `@permission_required("nomedoapp.nomedapermissao")`;
  - Testando o usuário com `user.has_perm("nomedoapp.nomedapermissao")`;
  - Diretamente nos templates com `{% if perms.nomedoapp.nomedapermissao %}`;
- Para o *decorator* é necessário importar:

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

Referências

- https://docs.djangoproject.com/en/5.1/topics/auth/

Dúvidas?