

AUTENTICAÇÃO COM DJANGO

1. Configuração Inicial

No Django, o pacote de autenticação já está disponível por padrão. Para garantir que a autenticação esteja ativa, verifique se `django.contrib.auth` está incluído em `INSTALLED_APPS` no arquivo `settings.py`:

incluir dentro do `installed_apps` os seguintes pacotes.

```
INSTALLED_APPS = [  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
  
]
```

2. Model de Usuário

O Django já possui um modelo de usuário embutido (`User`), que é similar ao que o Laravel fornece. Esse modelo está disponível em `django.contrib.auth.models.User`. Se você quiser personalizar o modelo de usuário, pode estender o modelo `AbstractUser` ou usar o `AbstractBaseUser` para maior controle.

Modelo Padrão:

python

Copiar código

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

Personalizando o Usuário:

python

Copiar código

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

```
class CustomUser(AbstractUser):  
    # Adicione campos extras se necessário  
    pass
```

Se você optar por um modelo de usuário personalizado, defina-o no `settings.py`:

python

Copiar código
AUTH_USER_MODEL = 'myapp.CustomUser'

3. Views de Autenticação

Django fornece views prontas para login, logout, e gerenciamento de senha. Você pode usá-las diretamente ou personalizá-las.

Login: Você pode criar uma rota para o login e usar a view pronta:

python
Copiar código

```
from django.contrib.auth import views as auth_views
```

```
urlpatterns = [  
    path('login/', auth_views.LoginView.as_view(template_name='login.html'), name='login'),  
]
```

Logout: Para logout, também há uma view pronta:

python
Copiar código
urlpatterns = [
 path('logout/', auth_views.LogoutView.as_view(), name='logout'),
]

4. Formulários de Autenticação

O Django fornece formulários prontos, como o AuthenticationForm para login e o UserCreationForm para registro de novos usuários.

Formulário de Login:

python
Copiar código
from django.contrib.auth.forms import AuthenticationForm

Formulário de Registro:

python
Copiar código

from django.contrib.auth.forms import UserCreationForm

5. Sistema de Sessão

Após o login bem-sucedido, o Django gera automaticamente uma sessão para o usuário, semelhante ao Laravel.

6. Redirecionamento Após o Login/Logout

Você pode especificar para onde o usuário será redirecionado após o login ou logout, configurando os seguintes parâmetros em settings.py:

```
python
```

```
Copiar código
```

```
LOGIN_REDIRECT_URL = 'home' # Rota ou URL após o login
```

```
LOGOUT_REDIRECT_URL = 'login' # Rota ou URL após o logout
```

7. Protegendo Páginas com Autenticação

Para proteger uma página e exigir que o usuário esteja logado, você pode usar o decorador `login_required`:

```
python
```

```
Copiar código
```

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

```
@login_required
```

```
def minha_view(request):
```

```
    # código da view
```

Ou, em suas urls.py:

```
python
```

```
Copiar código
```

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

```
urlpatterns = [
```

```
    path('minha-pagina/', login_required(minha_view), name='minha_pagina'),
```

```
]
```

8. Registro de Usuários

Para criar um sistema de registro, você pode usar o `UserCreationForm` e construir uma view que permita novos cadastros de usuários.

```
python
```

```
Copiar código
```

```
from django.shortcuts import render, redirect
```

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

```
def register(request):
```

```
    if request.method == 'POST':
```

```
        form = UserCreationForm(request.POST)
```

```
        if form.is_valid():
```

```
            form.save()
```

```
            return redirect('login')
```

```
    else:
```

```
        form = UserCreationForm()
```

```
    return render(request, 'register.html', {'form': form})
```

Adicione a rota para o registro:

```
python
Copiar código
urlpatterns = [
    path('register/', register, name='register'),
]
```

9. Gerenciamento de Senhas

Django também oferece funcionalidades para redefinição e alteração de senha, incluindo views e formulários.

Redefinição de Senha:

```
python
Copiar código
path('password_reset/', auth_views.PasswordResetView.as_view(), name='password_reset'
),
path('password_reset/
done/', auth_views.PasswordResetDoneView.as_view(), name='password_reset_done'),
```

Comparação com o Laravel

- **Autenticação:** O Django já fornece uma estrutura similar ao Laravel, com autenticação pronta para uso, incluindo login, logout, registro e gerenciamento de senhas.
- **Sessões:** As sessões são gerenciadas automaticamente no Django, assim como no Laravel.
- **Proteção de Rotas:** Assim como no Laravel, o Django oferece proteção de rotas com decoradores e middleware.
-

Com esses passos, você pode implementar um sistema de autenticação completo em Django, semelhante ao do Laravel.