

Django Inicial

Clase 30

django

CONQUER BLOCKS

Decoradores

Decoradores

Un decorador es esencialmente una función que toma otra función como argumento y devuelve una nueva función con funcionalidad añadida o modificada. En Django, se usan principalmente en vistas para tareas como:

- Autenticación de usuarios.
 - Control de permisos.
 - Cacheo de resultados.
 - Restricción de acceso.

Decoradores

1. @login_required

Este decorador asegura que el usuario esté autenticado antes de acceder a una vista. Si el usuario no ha iniciado sesión, será redirigido a la página de inicio de sesión.

python

 Copiar

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

@login_required
def mi_vista(request):
    # Código de la vista aquí
    return render(request, 'mi_template.html')
```

Por defecto, @login_required redirige a la URL de inicio de sesión definida en LOGIN_URL en el archivo settings.py. Puedes personalizar esto si es necesario:

python

 Copiar

```
@login_required(login_url='/mi-login/')
def mi_vista(request):
    # Código de la vista aquí
    return render(request, 'mi_template.html')
```


Decoradores

2. @permission_required

Este decorador se asegura de que el usuario tiene un permiso específico antes de permitirle acceder a la vista. Por ejemplo, para verificar si un usuario tiene el permiso para agregar un objeto:

python

 Copiar

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

@permission_required('app.add_modelo', raise_exception=True)
def agregar_modelo(request):
    # Código de la vista para agregar un modelo
    return render(request, 'agregar_modelo.html')
```

En este ejemplo, `app.add_modelo` es el nombre del permiso que se está verificando. Si el usuario no tiene este permiso, será redirigido a una página de error o se lanzará una excepción si `raise_exception=True` está habilitado.

Decoradores

3. @user_passes_test

Este decorador permite mayor flexibilidad al verificar si un usuario cumple con una condición personalizada. Por ejemplo, si deseas permitir el acceso solo a usuarios administradores:

python

 Copiar

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

def es_admin(user):
    return user.is_superuser

@user_passes_test(es_admin)
def vista_admin(request):
    # Vista accesible solo para superusuarios
    return render(request, 'admin_view.html')
```

Puedes definir cualquier función de prueba personalizada en lugar de solo `user.is_superuser`.

◀ Despedida ▶

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez