

Monkey patch phù hợp từng trường hợp trong Django

I. Monkey patch một **model method**

Ví dụ: bạn muốn override method `save` của `User` model trong Django mà không sửa trực tiếp `User` gốc

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

original_save = User.save

def custom_save(self, *args, **kwargs):
    print(f"[Custom Save] Saving user: {self.username}")
    return original_save(self, *args, **kwargs)

User.save = custom_save
```

II. Monkey patch một view của app khác (hoặc của Django)

```
from django.contrib.auth.views import LoginView

original_dispatch = LoginView.dispatch

def custom_dispatch(self, request, *args, **kwargs):
    print("[Custom LoginView] User is logging in")
    return original_dispatch(self, request, *args, **kwargs)

LoginView.dispatch = custom_dispatch
```

III. Monkey patch Django Signals

Ví dụ: override xử lý mặc định của `post_save` signal

```
from django.db.models.signals import post_save
from django.contrib.auth.models import User

def custom_user_post_save(sender, instance, created, **kwargs):
    if created:
        print(f"[Custom Signal] New user created: {instance.username}")

# Ngắt kết nối các signal hiện tại nếu cần
post_save.disconnect(dispatch_uid="default_user_post_save", sender=User)

# Gắn signal monkey patch
post_save.connect(custom_user_post_save, sender=User)
```

IV. Monkey patch một form method

Ví dụ: bạn muốn thay đổi method `clean_email` trong `UserCreationForm`

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

def custom_clean_email(self):
    email = self.cleaned_data.get("email")
    if not email.endswith("@mycompany.com"):
        raise forms.ValidationError("Email must be a company email")
    return email

UserCreationForm.clean_email = custom_clean_email
```

V. Monkey patch Django admin

Ví dụ: thêm log mỗi lần object được lưu trong admin

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

original_save_model = admin.ModelAdmin.save_model

def custom_save_model(self, request, obj, form, change):
    print(f"[Admin Log] {obj} saved by {request.user}")
    return original_save_model(self, request, obj, form, change)

admin.ModelAdmin.save_model = custom_save_model
```

VI. Gợi ý tổ chức Monkey Patch sạch sẽ

```
myapp/
├─ monkey/
│   ├── __init__.py
│   ├── user_patch.py
│   └── view_patch.py
```

Trong `myapp/monkey/__init__.py`:

```
from .user_patch import patch_user
from .view_patch import patch_login_view

def apply_monkey_patches():
    patch_user()
    patch_login_view()
```

Trong `apps.py` của app bạn:

```
from django.apps import AppConfig

class MyAppConfig(AppConfig):
    name = 'myapp'

    def ready(self):
        from myapp.monkey import apply_monkey_patches
```

apply_monkey_patches()

Mẹo vặt

Tình huống	Lời khuyên
App bạn override chưa load	Hãy đảm bảo thứ tự cài đặt trong <code>INSTALLED_APPS</code>
Gọi lại hàm gốc	Luôn backup lại hàm gốc <code>original_func = Class.func</code>
Muốn patch tạm	Có thể dùng <code>unittest.mock.patch()</code>
Patch không hoạt động	Kiểm tra xem <code>ready()</code> trong <code>apps.py</code> có được gọi không

Tác giả: **Đỗ Ngọc Tú**
Công Ty Phần Mềm **VHTSoft**

Phiên bản #1
Được tạo 15 tháng 4 2025 16:51:53 bởi Đỗ Ngọc Tú
Được cập nhật 15 tháng 4 2025 17:01:55 bởi Đỗ Ngọc Tú