

مستندات: ساخت یک سیگنال شخصی سازی شده در جنگو

این مستندات مراحل ساخت، استفاده و معرفی یک سیگنال سفارشی را در جنگو توضیح می دهد. سیگنال ها در جنگو به ما امکان می دهند که در هنگام وقوع یک رویداد خاص (مانند ذخیره یک شیء یا حذف آن) به طور خودکار توابع یا کدهای خاصی را اجرا کنیم.

۱. ساختار اصلی سیگنال شخصی سازی شده

برای ساخت سیگنال های شخصی سازی شده در جنگو، باید مراحل زیر را دنبال کنید.

مراحل ایجاد سیگنال سفارشی

مرحله ۱: ساخت فایل `signals.py`

در پروژه جنگو، یک فایل به نام `signals.py` ایجاد کنید. این فایل مکانی است که سیگنال های سفارشی خود را در آن تعریف می کنید.

مرحله ۲: وارد کردن ماژول ها و کتابخانه های لازم

برای ایجاد یک سیگنال سفارشی، از کلاس `Signal` که در `django.dispatch` قرار دارد استفاده می کنیم. علاوه بر این، باید از مدل هایی که قصد داریم برای آن ها سیگنال تعریف کنیم نیز استفاده کنیم.

```
``python
```

```
# در فایل signals.py
```

```
from django.db.models.signals import post_save
```

```
from django.dispatch import receiver, Signal
```

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

```
'''
```

مرحله ۳: تعریف سیگنال سفارشی

برای ایجاد یک سیگنال سفارشی، یک نمونه از کلاس `Signal` ایجاد کنید. این نمونه یک سیگنال را نمایش می‌دهد که با پارامترهای دلخواه شما قابل فراخوانی است.

```
```python
```

```
تعریف سیگنال سفارشی
```

```
user_logged_in_custom = Signal(providing_args=["user", "timestamp"])
```

```
'''
```

مرحله ۴: تعریف دریافت‌کننده (Receiver) برای سیگنال

دریافت‌کننده سیگنال تابعی است که هنگامی که سیگنال ارسال می‌شود، اجرا خواهد شد. این تابع باید تزئین‌کننده `@receiver` (decorator) را با سیگنال دلخواه همراه داشته باشد.

```
```python
```

```
@receiver(user_logged_in_custom)
```

```
def log_user_login(sender, **kwargs):
```

```
    user = kwargs.get("user")
```

```
    timestamp = kwargs.get("timestamp")
```

```
    print(f"وارد سیستم شده است {timestamp} در زمان {user} کاربر")
```

```
'''
```

۲. فراخوانی سیگنال

سیگنال‌ها به طور مستقیم از طریق متد `send` ارسال می‌شوند. برای این منظور باید سیگنال خود را در جایی از کد خود فراخوانی کنید، مانند ویو (view) یا مدل.

مثال: فراخوانی سیگنال در ویو

فرض کنید می‌خواهیم سیگنال `user_logged_in_custom` را پس از ورود موفق کاربر فراخوانی کنیم.

```
```python
from django.utils import timezone
from django.contrib.auth import login
from django.shortcuts import render
from .signals import user_logged_in_custom

def user_login_view(request):
 # اعتبارسنجی و ورود کاربر
 user = request.user
 login(request, user)

 # ارسال سیگنال با اطلاعات کاربر و زمان ورود
 user_logged_in_custom.send(sender=user.__class__, user=user,
 timestamp=timezone.now())

 return render(request, "home.html")
```
```

در این مثال، پس از اینکه کاربر با موفقیت وارد سیستم شد، سیگنال `user_logged_in_custom` فراخوانی می‌شود و اطلاعات کاربر و زمان ورود وی به سیگنال ارسال می‌شود.

۳. معرفی سیگنال در جنگو

برای اطمینان از اینکه سیگنال‌های شما به درستی کار می‌کنند، باید فایل `signals.py` را به جنگو معرفی کنید. یکی از راه‌های متداول این است که سیگنال را در فایل `apps.py` اپلیکیشن خود ثبت کنید.

مثال: معرفی سیگنال در `apps.py`

ابتدا فایل `apps.py` را باز کرده و تغییرات زیر را اعمال کنید:

```
``python
# در فایل apps.py

from django.apps import AppConfig

class MyAppConfig(AppConfig):
    name = 'myapp'

    def ready(self):
        اینجا سیگنال‌ها را معرفی می‌کنیم #
        import myapp.signals
...

```

سپس در فایل `__init__.py` اپلیکیشن، کانفیگ جدید خود را اضافه کنید تا جنگو آن را بشناسد.

```
```python
__init__.py در فایل
default_app_config = 'myapp.apps.MyAppConfig'
```
```

نتیجه‌گیری

در این مستندات، مراحل کامل ایجاد یک سیگنال شخصی‌سازی‌شده در جنگو توضیح داده شد. این سیگنال در هنگام ورود کاربر با موفقیت فراخوانی می‌شود و اطلاعات مربوط به ورود کاربر را نمایش می‌دهد. به طور خلاصه:

-سیگنال در `signals.py` تعریف شد.

-دریافت‌کننده سیگنال با استفاده از `@receiver` تعریف شد.

-سیگنال در ویو فراخوانی شد.

-سیگنال در `apps.py` به جنگو معرفی شد.

این روش باعث می‌شود تا کدهایتان تمیزتر، قابل توسعه و انعطاف‌پذیرتر باشند.