

TRAINING DAY19 REPORT

15 JULY 2025

Implementing Django Signals

Django Signals allow **decoupled communication** between different parts of a Django application.

They help automate responses to specific events, such as:

- When a model instance is **saved** (`post_save`)
- When a model instance is **deleted** (`post_delete`)
- When a user is **created or updated**

Using signals ensures that business logic (like sending emails, updating logs, etc.) happens automatically — without explicitly writing it in every view or form.

Implementation Example

1. File Structure:

We'll create a new file called `signals.py` inside the blog app.

```
blog/  
├── models.py  
├── views.py  
├── signals.py  
├── apps.py  
└── ...
```

2. Writing the Signal:

```
blogapi > blog > signal.py > ...
1  # blog/signals.py
2  from django.db.models.signals import post_save
3  from django.dispatch import receiver
4  from .models import Post
5
6  @receiver(post_save, sender=Post)
7  def notify_post_creation(sender, instance, created, **kwargs):
8      if created:
9          print(f"🟢 New Post Created: '{instance.title}' by {instance.author}")
10     else:
11         print(f"🟡 Post Updated: '{instance.title}'")
12
```

3. Connecting the Signal:

To ensure Django loads the signal automatically when the app starts, modify the apps.py file:

```
blogapi > blog > apps.py > ...
1  # blog/apps.py
2  from django.apps import AppConfig
3
4  class BlogConfig(AppConfig):
5      default_auto_field = 'django.db.models.BigAutoField'
6      name = 'blog'
7
8      def ready(self):
9          import blog.signals
10
```

And update `__init__.py`:

```
blogapi > blogapi > __init__.py > ...
1  # blog/__init__.py
2  default_app_config = 'blog.apps.BlogConfig'
3
4
```

4. Testing the Signal:

Whenever a new post is created or updated via the admin panel or code, the message will automatically appear in the terminal (server console):

```
pgsql
● New Post Created: 'Django Basics' by shivam
● Post Updated: 'Django Basics'
```

Practical Use Cases:

Use Case	Description
Auto-create user profile	When a new user registers, automatically create their profile
Send notification	When a post is published, send an email/notification
Auto-update logs	Keep track of changes to important models
Sync data	Between related tables or external APIs