**[Django Signals](https://docs.djangoproject.com/en/3.2/topics/signals/) Answer**

1. **Are Django signals executed synchronously or asynchronously by default?**

* **Synchronous** – Django signals are **executed immediately and synchronously** by default.

1. **Do Django signals run in the same thread as the caller?**

* **Yes** – Since signals are synchronous by default, they run in **the same thread** as the caller.

1. **Do Django signals run in the same database transaction as the caller?**

* **Yes** – If you connect to signals like post\_save, they are triggered **within the same transaction** (unless you manually use transaction.on\_commit() or similar).

1. **Creating Model:**

from django.db import models

class Book(models.Model):

title = models.CharField(max\_length=100)

1. **Signals:**

import threading

from django.db.models.signals import post\_save

from django.dispatch import receiver

from .models import Book

@receiver(post\_save, sender=Book)

def book\_saved\_signal(sender, instance, \*\*kwargs):

print(" Signal Triggered")

print("Signal Thread:", threading.current\_thread().name)

if not Book.objects.filter(id=instance.id).exists():

print("Book not in DB (shouldn’t happen if in same transaction)")

else:

print("Book exists in DB (same transaction)")

1. **Views:**

import threading

from myapp.models import Book

print("Caller Thread:", threading.current\_thread().name)

Book.objects.create(title="Django")

**Output:**

Caller Thread: MainThread

Signal Triggered

Signal Thread: MainThread

Book exists in DB (same transaction)