

# Walkthrough: Connecting Django to Supabase (Session Pooler)

This guide explains how to configure your Django project to use **Supabase PostgreSQL** via the **Session Pooler** connection string.

---

## 1. Prerequisites

- A working **Django project**.
  - **Supabase account** with an active project.
  - Installed Python packages:
  - `pip install psycopg2 dj-database-url python-dotenv`
- 

## 2. Retrieve Supabase Session Pooler Connection

1. Log in to the Supabase Dashboard.
  2. Select your project.
  3. Go to **Project** → **Connect** → **Connection Info**.
  4. Copy the **Session Pooler** connection string (not the direct DB string).  
It should look like:
  5. `postgresql://postgres:YOUR_PASSWORD@aws-0-<project-ref>.pooler.supabase.com:5432/postgres`
- 

## 3. Configure .env File

In your project root, create or update a `.env` file with:

```
# Supabase PostgreSQL (Session Pooler)
```

```
DATABASE_URL='postgresql://postgres:YOUR_PASSWORD@aws-0-<project-ref>.pooler.supabase.com:5432/postgres?sslmode=require'
```

 Notes:

- Replace YOUR\_PASSWORD with your **Supabase DB password**.
- Replace <project-ref> with your project reference (found in the dashboard).
- Keep sslmode=require since Supabase enforces SSL.

---

## 4. Update settings.py

In your Django settings.py, configure the database:

```
# settings.py
```

```
import os
```

```
from pathlib import Path
```

```
import dj_database_url
```

```
from dotenv import load_dotenv
```

```
# Load environment variables from .env
```

```
load_dotenv()
```

```
BASE_DIR = Path(__file__).resolve().parent.parent
```

```
# Database configuration using Supabase Session Pooler
```

```
DATABASES = {
```

```
    "default": dj_database_url.config(
```

```
        default="sqlite:///db.sqlite3",
```

```
        conn_max_age=600, # persistent connections
```

```
        ssl_require=True # enforce SSL
```

```
    )
```

```
}
```

---


## 5. Verify Environment Variables (OPTIONAL)

Ensure Django is reading your .env correctly:

```
python manage.py shell
```

```
import os
```

```
print(os.environ.get("DATABASE_URL"))
```

- If you see your connection string →  Good.
- If it prints None → your .env is not being loaded.

---

## 6. Test the Database Connection

Run migrations to confirm connectivity:

```
python manage.py migrate
```

If successful, Django is now connected to Supabase via the session pooler.

---

 You're Done!