Connecting Django to Different Databases:

1. SQLite3 (Default Database)

Step 1: Install Django

Make sure you have **Django** installed. If not, you can install it using:

```
pip install django
```

Step 2: Create a New Django Project

Create a new **Django** project (replace `myproject` with your desired project name):

```
django-admin startproject myproject
cd myproject
```

Step 3: Configure `settings.py`

Open `myproject/settings.py` and locate the `DATABASES` section. By default, it uses **SQLite3**. No further configuration is needed for **SQLite3**.

```
# myproject/settings.py

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

Step 4: Run Migrations

Create the initial database schema:

```
python manage.py migrate
```

2. MySQL

Step 1: Install MySQL

Ensure you have **MySQL** installed. If not, follow the instructions for your operating system.

Step 2: Create a MySQL Database

Log in to **MySQL** as the root user:

```
sudo mysql
```

Create a database for your project (replace `myproject_db` with your desired database name):

```
CREATE DATABASE myproject_db;
```

Step 3: Configure `settings.py`

Update the `DATABASES` section in `myproject/settings.py` to use **MySQL**:

```
# myproject/settings.py

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'myproject_db',
        'USER': 'your_mysql_user',
        'PASSWORD': 'your_mysql_password',
        'HOST': 'localhost',  # Change if needed
        'PORT': '3306',       # Default MySQL port
    }
}
```

Step 4: Run Migrations

Apply migrations to create the database tables:

```
python manage.py migrate
```

3. PostgreSQL

Step 1: Install PostgreSQL

Install **PostgreSQL** on your system.

Step 2: Create a PostgreSQL Database

Log in to **PostgreSQL** as the superuser:

```
sudo -u postgres psql
```

Create a database (replace `myproject_db` with your desired name):

```
CREATE DATABASE myproject_db;
```

Step 3: Configure `settings.py`

Update the `DATABASES` section in `myproject/settings.py` for **PostgreSQL**:

```
# myproject/settings.py

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'myproject_db',
        'USER': 'your_postgres_user',
        'PASSWORD': 'your_postgres_password',
        'HOST': 'localhost',  # Change if needed
        'PORT': '5432',       # Default PostgreSQL port
    }
}
```

Step 4: Run Migrations

Apply migrations:

```
python manage.py migrate
```

4. Oracle 11g

Step 1: Install Oracle Instant Client

Install the Oracle Instant Client on your system.

Step 2: Configure `settings.py`

Update the DATABASES section in myproject/settings.py for **Oracle**:

```
# myproject/settings.py

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.oracle',
        'NAME': 'your_oracle_service_name',
        'USER': 'your_oracle_user',
        'PASSWORD': 'your_oracle_password',
        'HOST': 'localhost',  # Change if needed
        'PORT': '1521',       # Default Oracle port
    }
}
```

Step 3: Run Migrations

Apply migrations:

```
python manage.py migrate
```

5. MongoDB

Step 1: Install PyMongo

Install the **PyMongo** package:

```
pip install pymongo
```

Step 2: Configure `settings.py`

Update the DATABASES section in myproject/settings.py for **MongoDB**:

```
# myproject/settings.py

DATABASES = {
    'default': {
        'ENGINE': 'djongo',
        'NAME': 'myproject_db',  # Your MongoDB database name
        'CLIENT': {
            'host':
'mongodb+srv://sampleUser:samplePassword@cluster0-
gbdot.mongodb.net/sampleDB?retryWrites=true&w=majority',
        },
    }
}
```

Replace 'myproject_db' with your actual **MongoDB** database name and adjust the connection string accordingly.

Step 3: Run Migrations

Apply migrations to create the necessary database tables:

```
python manage.py migrate
```

Conclusion

You've successfully configured your **Django** project to connect to various databases! Remember to adjust the settings according to your specific setup for each database. Happy coding! 🚀🐍🍃

NOTE

For sqlite 3: no installations

For Mysql: pip install mysql client

For Postgres: No installations

For Oracle: pip install cx_Oracle

For mongodb: pip install djongo/pymongo