Connecting Django with sqlite3 and PostgreSQL database - [Windows]

//**//

To connect your Django application with your default **sqlite3 database** just you have to do two changes in the setting.py file in the main project directory.

1. **Open the setting.py file**

Go to Installed_Apps there just add the your created app

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
     '……',
     '……',
]
```

For example I'm adding "**Main app**" as shown in the image below just change your app name. [Check apps.py file in your created app]

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'main.apps.MainConfig',
]
```

No need edit the remaining parts in the setting.py file if you are using sqlite3 database because Database field setting is provided by default in django.

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

2. **Go to your project directory on command line and make migrations.**
   Migrations will create models for your specified class in the models.py file.

   >>**python manage.py makemigrations**

   After creating migrations run the below command to permanently reflects in the database.
   >>**python manage.py migrate**

   Now you check your migrations folder in the project directory.

//**********//

To connect your Django application with **PostgreSQL database** follow the below steps. First step is adding your app with setting.py file in the Installed_Apps and adding database specifications in the Databases.

1. **Open the setting.py file**

Go to Installed Apps there just add the your created app

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
     '……',
     '……',
]
```

For example I'm adding "**Course app**" as shown in the image below just change your app name. [Check apps.py file in your created app]

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'course.apps.CourseConfig',
]
```

In the same setting.py file change the database specifications according to the PostgreSQL database.

Before editing that Databases let's download PostgreSQL database and other requirements according your OS and install it.

Link to download PostgreSQL: https://www.postgresql.org/download/

And also you need a UI for PostgreSQL so you need to download the PgAdmin software and install it.

Link to download pgAdmin: https://www.pgadmin.org/download/

Next come back to the **setting.py** file go to Databases and change this defaults to your PostgreSQL.

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

For example change according to name, user and password given while installing the pgAdmin software.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'test',
        'USER': 'postgres',
        'PASSWORD': 'password',
        'HOST': 'localhost',
    }
}
```

We know that **Django [Python]** is different software and **Postgre** is different software to connect this we need one more connector/database adapter called **PSYCOPG2**. Let's download and install it on command line.

Command to install psycopg2: pip install psycopg2

2. **Go to your project directory on command line and make migrations.**
   Migrations will create models for your specified class in the models.py file.

   >>**python manage.py makemigrations**

   After creating migrations run the below command to permanently reflects in the database.

   >>**python manage.py migrate**

   Now you check your migrations folder in the project directory.