



Universidad Autónoma de Baja California
Campus Tijuana
Facultad de contaduría y administración



Alumno:

García Bretado David Antonio

Grupo:

372

“Base de datos”

Programación en Python

Tijuana B.C. a 23 de octubre de 2018

Configuración de la base de datos

Ahora, abra el archivo **mysite/settings.py**. Es un módulo normal de Python con variables de nivel de módulo que representan la configuración de Django.

Por defecto la configuración utiliza SQLite. Si tiene poca experiencia con bases de datos o su interés es solo probar Django esta es la opción más fácil. SQLite está incluido en Python por lo que no tendrá que instalar nada más para soportar su base de datos. Sin embargo, al iniciar su primer proyecto real, es posible que desee utilizar una base de datos más potente como PostgreSQL para evitar dolores de cabeza en el futuro al tener que cambiar entre base de datos.

Si desea utilizar otra base de datos, instale los conectores de base de datos apropiados, y cambie las siguientes claves en el ítem **DATABASES 'default'** para que se ajusten a la configuración de conexión de la base de datos:

- **ENGINE** – bien sea **'django.db.backends.sqlite3'**, **'django.db.backends.postgresql_psycopg2'**, **'django.db.backends.mysql'**, o **'django.db.backends.oracle'**. Otros backends también están disponibles.
- **NAME** – el nombre de su base de datos. Si está utilizando SQLite, la base de datos será un archivo en su computadora; en ese caso **NAME** debe ser la ruta absoluta completa, incluyendo el respectivo nombre del archivo. El valor predeterminado, **os.path.join(BASE_DIR, 'db.sqlite3')**, guardará el archivo en el directorio de su proyecto.

Si no está utilizando SQLite como su base de datos, se deben añadir ajustes adicionales tales como **USER**, **PASSWORD**, y **HOST** se deben añadir. Para más información, vea la documentación de referencia para **DATABASES**.

Mientras que usted está editando el archivo **mysite/settings.py**, configure **TIME_ZONE** a su zona horaria.

Además, observe que la configuración de **INSTALLED_APPS** se encuentra en la parte superior del archivo. Esta contiene los nombres de todas las aplicaciones Django que están activadas en esta instancia de Django. Las aplicaciones se pueden usar en diversos proyectos y usted puede empaquetar y distribuir las para que otras personas las puedan utilizar en sus proyectos.

Por defecto, **INSTALLED_APPS** contiene las siguientes aplicaciones y Django viene equipado con todas ellas:

- **django.contrib.admin** - El sitio administrativo. Usted lo utilizará dentro de poco.
- **django.contrib.auth** - Un sistema de autenticación.
- **django.contrib.contenttypes** - Un framework para los tipos de contenido.
- **django.contrib.sessions** - Un framework de sesión.
- **django.contrib.messages** - Un framework de mensajería.
- **django.contrib.staticfiles** - Un framework para la gestión de archivos estáticos.

Estas aplicaciones se incluyen de forma predeterminada como una conveniencia para el caso común.

Algunas de estas aplicaciones utilizan al menos una tabla de base de datos, por lo que necesitamos crear las tablas en la base de datos antes de poder utilizarlas. Para ello, ejecute el siguiente comando:

```
$ python manage.py migrate
```

El comando **migrate** analiza la configuración **INSTALLED_APPS** y crea las tablas de base de datos necesarias según la configuración de base de datos en su archivo **mysite/settings.py** y las migraciones de base de datos distribuidas con la aplicación (trataremos este tema más tarde). Verá un mensaje para cada migración que aplique. Si está interesado, ejecute el cliente de línea de comandos para su base de datos y teclee **\dt** (PostgreSQL), **SHOW TABLES;** (MySQL), **.schema** (SQLite), o **SELECT TABLE_NAME FROM USER_TABLES;** (Oracle) para desplegar las tablas que creó Django.

Fuentes

<https://docs.djangoproject.com/es/2.1/intro/tutorial02/#database-setup>