



Conectar Django con Postgres

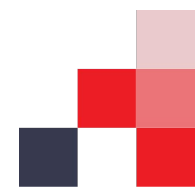




PostgreSQL

+

django



Instalar psycopg2

[Help](#)[Sponsors](#)[Log in](#)[Register](#)

psycopg2 2.8.6

```
pip install psycopg2
```

[Latest version](#)

Released: Sep 7, 2020

psycopg2 - Python-PostgreSQL Database Adapter

Navigation

[Project description](#)[Release history](#)

Project description

Psycopg is the most popular PostgreSQL database adapter for the Python programming language. Its main features are the complete implementation of the Python DB API 2.0 specification and the thread safety (several threads can share the same connection). It was designed for heavily multi-threaded applications that create and destroy lots of cursors and make a large number of concurrent “INSERT”s or “UPDATE”s.

Create Database

Para crear una base de datos inicial postgresql sobre la que trabajar podemos utilizar pgadmin4 o utilizar el terminal.

Ejecutamos los siguientes comandos en la terminal:

```
psql postgres
```

```
CREATE DATABASE bookstore;
```

```
\connect bookstore;
```



pgAdmin



Settings.py

Una vez que ya tenemos nuestra base de datos. El siguiente paso será configurar nuestra aplicación.

Para ello modificamos el archivo settings.py.



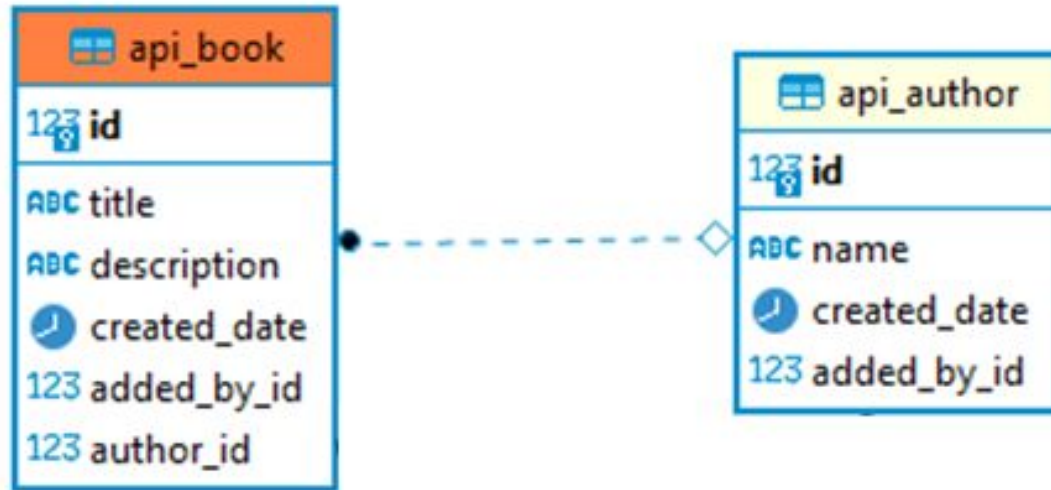
```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql_psycopg2',  
        'NAME': 'bookstore',  
        'USER': 'username',  
        'PASSWORD': 'password',  
        'HOST': 'localhost',  
        'PORT': '5432',  
    }  
}
```



Modelos ER



Models.py



PostgreSQL

+

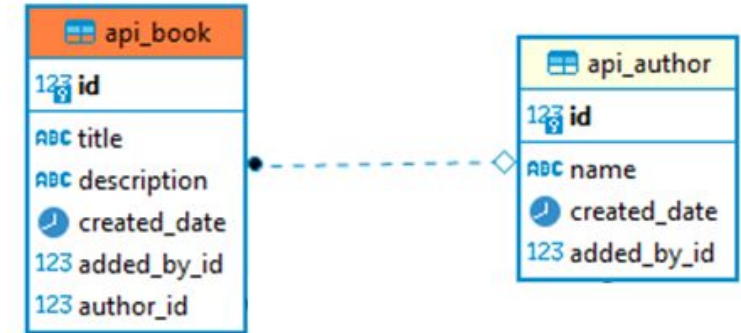
django



Models.py

En esta ocasión vamos a relacionar a nivel de modelo las diferentes entidades que aparecen en nuestro sistema de base de datos Postgres. En este ejemplo veremos la relación de 1/N.

Para ello modificamos el archivo api/models.py.

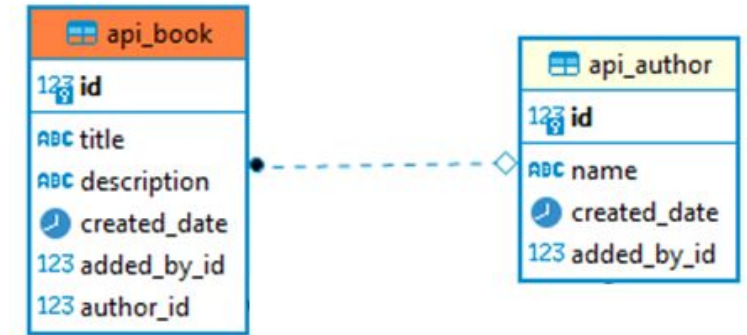


```
class Author(models.Model):
    name = models.CharField(max_length=200)
    added_by = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    created_date = models.DateTimeField(default=timezone.now)

    def __str__(self):
        return self.name
```


Hacemos lo mismo con el modelo book. Debemos establecer la relación de N/1 al igual que hicimos en la entidad autor para poder mantener la integridad referencial entre ambas. En este ejemplo veremos la relación es de N a 1 al contrario de lo que ocurría en autor que era de 1 a N.

Para ello modificamos el archivo api/models.py.



```
class Book(models.Model):
    title = models.CharField(max_length=200)
    description = models.CharField(max_length=300)
    author = models.ForeignKey(Author, on_delete=models.CASCADE)
    added_by = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    created_date = models.DateTimeField(default=timezone.now)

    def __str__(self):
        return self.title
```

Migrar BBDD

Recuerda que Django te permite definir modelos de bases de datos usando Python.

Siempre que creamos o hacemos cambios a un modelo, necesitamos decirle a Django que migre esos cambios a la base de datos. El ORM de Django luego escribe todos los comandos SQL CREATE TABLE para nosotros.

Code First

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

```
Operations to perform:
```

```
  Apply all migrations: admin, auth, contenttypes, sessions
```

```
Running migrations:
```

```
Applying contenttypes.0001_initial... OK
```

```
Applying auth.0001_initial... OK
```

```
Applying admin.0001_initial... OK
```

```
Applying admin.0002_logentry_remove_auto_add... OK
```

```
Applying admin.0003_logentry_add_action_flag_choices... OK
```

```
Applying contenttypes.0002_remove_content_type_name... OK
```

```
Applying auth.0002_alter_permission_name_max_length... OK
```

```
Applying auth.0003_alter_user_email_max_length... OK
```

```
Applying auth.0004_alter_user_username_opts... OK
```

```
Applying auth.0005_alter_user_last_login_null... OK
```

```
Applying auth.0006_require_contenttypes_0002... OK
```

```
Applying auth.0007_alter_validators_add_error_messages... OK
```

```
Applying auth.0008_alter_user_username_max_length... OK
```

```
Applying auth.0009_alter_user_last_name_max_length... OK
```

```
Applying auth.0010_alter_group_name_max_length... OK
```

```
Applying auth.0011_update_proxy_permissions... OK
```

```
Applying sessions.0001_initial... OK
```