EBook Gratis

APRENDIZAJE
django-models

Free unaffiliated eBook created from
Stack Overflow contributors.

#django-models

Tabla de contenido

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: django-models

It is an unofficial and free django-models ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official django-models.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con los modelos django

Observaciones

Esta sección proporciona una visión general de qué es django-models y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de los modelos django, y vincular a los temas relacionados. Dado que la Documentación para django-modelos es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

En general, cada modelo se asigna a una única tabla de base de datos. Para escribir el tipo de campo, límites, tamaño, etc. en el archivo model.py de la aplicación. Esto creará la tabla y los campos necesarios en la base de datos.

```
'''   models.py    '''
from django.db import models

    class table_name(models.Model):
        field_name= models.field_type(conditions)
```

Luego debemos informar a Django en `settinggs.py` sobre la aplicación que usará este modelo.

```
    '''  settinggs.py  ''''

    INSTALLED_APPS = [
    #...
    'app_name',
    #...    ]
```

Casi terminamos. Luego necesitamos migrar esta aplicación para que se creen las tablas de la base de datos. En la terminal escriba lo siguiente:

```
python manage.py migrate
```

`migrate` creará las bases de datos necesarias marcando la app_installed en el `setting.py`

Por `makemigrations` , Django sabrá los cambios que se realizan en los modelos.

```
python manage.py makemigrations
```

Eso es. Se crea su base de datos y puede ver el esquema en el terminal

```
python manage.py sqlmigrate app_name 0001
```

¿Qué son los modelos Django?

Un `Django model` generalmente se refiere a una tabla en la base de datos, los atributos de ese modelo se convierten en la columna de esa tabla. En más de un ejemplo del mundo real, crearía un modelo para cualquier entidad en su aplicación y almacenaría sus atributos con `django fields` que manejan automáticamente las conversiones de tipos de datos para la base de datos que estaría usando.

Una de las grandes características de Django es su `ORM` , no tiene que escribir ninguna consulta de base de datos, e incluso se recomienda NO escribir una al usar Django. ORM convierte sus `Django models` y todas las operaciones que realiza con las consultas de base de datos correspondientes. Esto significa que toda la manipulación que tiene que hacer, ahora con los objetos de Python creados a partir de ese modelo, y todo el material de la base de datos subyacente será cuidado por el `ORM` de Django. Hay un montón de ajustes y personalizaciones que podría hacer con él.

El `ORM` de Django es compatible con todas las bases de datos principales como `Postgres` , `MySQL` , `sqlite3` y otras bases de datos de empresas provistas con los controladores adecuados. Esto también significa que no tiene que preocuparse por la base de datos subyacente que está utilizando, o incluso si desea cambiar de una base de datos a otra, puede hacerlo sin cambiar una sola línea de la lógica de su aplicación, solo cambie la cadena de la base de datos desde `settings.py` , descargue los datos antiguos y debería estar listo.

Ejemplo de modelo Django

Un ejemplo simple sería para una aplicación de administración de bibliotecas; Tendrías 2 modelos, por ejemplo, `student` y `book`

en `models.py`:

```
from django.db import models

class student(models.Model):
    roll_no = models.IntegerField(primary_key=True)
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
```

Aquí le hemos dado a roll_no una clave principal para el modelo de estudiante, pero incluso si no le damos una clave primaria a ningún atributo, Django asignaría automáticamente un atributo llamado `id` , que se asignaría automáticamente y se incrementaría en la creación de nuevas filas.
.

Ahora puede importar este modelo en sus `views` o en un proyecto e interactuar con él simplemente creando un objeto de ese modelo.

Django tiene muchos Campos incorporados disponibles, o incluso puedes crear los tuyos también.

Django también admite relaciones entre modelos, `many-to-many` , `one-to-one` , `many-to-one` .

El detallado documento de Django para Modelos.

Lea Empezando con los modelos django en línea: https://riptutorial.com/es/django-models/topic/7575/empezando-con-los-modelos-django

Creditos

S. No	Capítulos	Contributors
1	Empezando con los modelos django	Community, Md.Sifatul Islam, Vatsal Parekh