Бесплатная электронная книга

УЧУСЬ

django-models

Free unaffiliated eBook created from
Stack Overflow contributors.

#django-models

Около

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

глава 1: Начало работы с django-models

замечания

В этом разделе представлен обзор того, что такое модели django, и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в моделях джанго, а также ссылки на связанные темы. Поскольку документация для моделей django является новой, возможно, вам придется создавать начальные версии этих связанных тем.

Examples

Установка или настройка

Как правило, каждая модель сопоставляется с одной таблицей базы данных. Мы должны писать тип поля, ограничения, размер и т. Д. В файле model.py приложения. Это создаст необходимую таблицу и поля в базе данных.

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

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

Затем нам нужно сообщить Django в `settinggs.py` о приложении, которое будет использовать эту модель.

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

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

Мы почти закончили. Затем нам нужно перенести это приложение, чтобы таблицы базы данных были созданы. В терминальном типе:

```
python manage.py migrate
```

`migrate` будет создавать необходимые базы данных, проверяя app_installed в `setting.py`

Посредством `makemigrations` Django будет знать изменения, внесенные в модели.

```
python manage.py makemigrations
```

Вот и все. Ваша база данных создана, и вы можете увидеть схему в терминале

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

Что такое модели Django?

Модель `Django model` обычно ссылается на таблицу в базе данных, атрибуты этой модели становятся столбцом этой таблицы. В более реальном примере вы создадите модель для любого объекта в своем приложении и сохраните его атрибуты с `django fields` которые автоматически обрабатывают преобразования типов данных для используемой базы данных.

Одной из замечательных особенностей Django является `ORM` , вам не нужно писать какой-либо запрос к базе данных, и даже рекомендуется не писать его при использовании Django. ORM преобразует ваши `Django models` и все операции, которые вы выполняете с ним, в соответствующие запросы к базе данных. Это означает, что все манипуляции, которые вы должны выполнить, теперь теперь с объектами python, созданными из этой модели, и все базовые элементы базы данных будут позаботиться о `ORM` Django. Есть множество настроек и настроек, с которыми вы могли бы справиться.

`ORM` Django поддерживает все основные базы данных, такие как `Postgres` , `MySQL` , `sqlite3` и другие базы данных предприятий, снабженные надлежащими драйверами. Это также означает, что вам не нужно заботиться о том, какую базовую базу данных вы используете, или даже если вы хотите перейти от одной базы данных к другой, вы можете сделать это, не меняя ни одной строки вашей логики приложения, просто измените строку базы данных из `settings.py` , выгрузите старые данные, и вам должно быть хорошо идти.

Пример модели Django

Простым примером может служить приложение для управления библиотекой; у вас было бы 2 модели, например, `student` и `book`

в `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)
```

Здесь мы предоставили roll_no первичный ключ для модели ученика, но даже если мы не предоставим первичный ключ для любого атрибута, Django автоматически присвоит атрибут с именем `id` , который будет автоматически назначаться и увеличиваться при создании новых строк ,

Теперь вы можете просто импортировать эту модель в свои `views` или в проект и взаимодействовать с ней, просто создав объект этой модели.

Django имеет множество встроенных полей , или даже вы можете создавать свои собственные.

Django также поддерживает отношения между моделями, `many-to-many` , `one-to-one` , `many-to-one` .

Подробный документ Django для моделей

Прочитайте Начало работы с django-models онлайн: https://riptutorial.com/ru/django-models/topic/7575/начало-работы-с-django-models

кредиты

S. No	Главы	Contributors
1	Начало работы с django-models	Community, Md.Sifatul Islam, Vatsal Parekh