Django ORM и административная панель

№ урока:	7	**Курс:**	Django Starter

Средства обучения:	Персональный компьютер с установленными: Python 3.8.2 Django 3.0.4

Обзор, цель и назначение урока

Цель урока. Целью данного урока является ознакомиться с основами работы Django приложения с базой данных. Научиться создавать данные в базе данных, а также рассмотреть процессы работы с моделями: вытягивание, сортировка, фильтрация, удаление, удаление связи, обновление и другое. В конце урока будет рассмотрено то, как можно добавлять модели в админ панель.

Изучив материал данного занятия, учащийся сможет:

- Создавать данные в базе данных с помощью моделей.
- Научиться работать с моделями и Django ORM, а именно вытягивать данные, сортировать, фильтровать, удалять и обновлять.
- Создавать администратора.
- Научиться подключать к админ панели модели из проекта.

Содержание урока

1) QuerySet и работа с ним
2) Создание записей с помощью моделей
3) Сортировка и фильтрация: детальный разбор
4) Сложные запросы
5) Администратор и админ панель
    a) Как подключать модели
    b) Как создавать Администратора
    c) Как можно улучшить админ панель

Резюме

- **QuerySet**, по сути, — список объектов заданной модели. QuerySet позволяет читать данные из базы данных, фильтровать и изменять их порядок.

- Внутренне QuerySet может быть создан, отфильтрован, нарезан и, как правило, передан без фактического запроса к базе данных. На самом деле никаких действий с базой данных не происходит, пока вы не сделаете что-то для оценки набора запросов.
    - Итерация. QuerySet является итеративным, и он выполняет свой запрос к базе данных при первой итерации по нему.
    - Срезы. Как объяснено в Ограничение QuerySet, QuerySet может быть нарезан, используя синтаксис Python для срезов массивов. Срез невыясненного QuerySet обычно возвращает другой не вычисленный QuerySet, но Django выполнит запрос к базе данных, если вы используете параметр «step» синтаксиса среза, и вернет список. Срез QuerySet, который был вычислен, также возвращает список.
    - repr(). QuerySet вычисляется, когда вы вызываете repr() для него. Это удобно для интерактивного интерпретатора Python, поэтому вы можете сразу увидеть свои результаты при интерактивном использовании API.

CyberBionic systematics
CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- len(). QuerySet вычисляется, когда вы вызываете len() для него. Это, как вы могли ожидать, возвращает длину списка результатов.
- list(). Принудительное вычисление QuerySet путем вызова list() для него.
- bool(). Тестирование QuerySet в логическом контексте, например, с использованием bool(), or, and или оператора if, вызовет выполнение запроса. Если есть хотя бы один результат - QuerySet равен True, иначе - False.

- Когда вы будете взаимодействовать с QuerySet, вы будете использовать его фильтры цепочки. Для этого большинство методов QuerySet возвращают новые наборы запросов. Методы, которые возвращают новый QuerySet:
  - filter() - возвращает новый QuerySet, содержащий объекты, которые соответствуют заданным параметрам поиска.
  - exclude() - возвращает новый QuerySet, содержащий объекты, которые не соответствуют указанным параметрам поиска.
  - order_by() - По умолчанию результаты, возвращаемые QuerySet, упорядочиваются с помощью кортежа, заданного параметром ordering в классе Meta модели. Вы можете переопределить это для каждого QuerySet, используя метод order_by.
  - reverse() - Используйте метод reverse() для изменения порядка, в котором возвращаются элементы набора запросов. Вызов reverse() во второй раз восстанавливает порядок в нормальном направлении.
  - distinct() - Возвращает новый QuerySet, который использует SELECT DISTINCT в своем SQL-запросе. Это исключает повторяющиеся строки из результатов запроса.
  - values() - Возвращает QuerySet, который возвращает словари, а не экземпляры модели, когда используется как итеративный.
  - values_list() - Это похоже на values(), за исключением того, что вместо возврата словарей он возвращает кортежи при повторении. Каждый кортеж содержит значение из соответствующего поля или выражения, переданное в вызов values_list() - поэтому первый элемент является первым полем и т.д.
  - dates() - Возвращает QuerySet, который вычисляет список объектов datetime.date, представляющих все доступные даты определенного вида в QuerySet.
  - datetimes() - Возвращает QuerySet, который оценивает список объектов datetime.datetime, представляющих все доступные даты определенного вида в содержимом QuerySet.
  - none() - Вызов none() создаст набор запросов, который никогда не вернет никаких объектов, и при доступе к результатам запрос не будет выполнен. Набор запросов qs.none() является экземпляром EmptyQuerySet.
  - all() - Возвращает копию текущего QuerySet (или подкласса QuerySet). Это может быть полезно в ситуациях, когда вы захотите передать либо менеджер модели, либо QuerySet и выполнить дальнейшую фильтрацию по результату. После вызова all() для любого объекта у вас обязательно будет QuerySet для работы.
  - union() - Использует оператор SQL UNION для объединения результатов двух или более QuerySet'ов.
  - select_related() - Возвращает QuerySet, который будет «следовать» отношениям внешнего ключа, выбирая дополнительные данные связанного объекта при выполнении своего запроса. Это повышение производительности, которое приводит к одному более сложному запросу, но означает, что дальнейшее использование отношений внешнего ключа не потребует запросов к базе данных.
  - extra() - Иногда синтаксис запроса Django сам по себе не может легко выразить сложное предложение WHERE. Для этих крайних случаев Django предоставляет модификатор extra() QuerySet - ловушку для вставки определенных предложений в SQL, генерируемый QuerySet.
  - get() - Возвращает объект, соответствующий заданным параметрам поиска, который должен быть в указанном формате.

CyberBionic systematics
CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

    - create() - Удобный метод для создания объекта и сохранения всего за один шаг.
    - get_or_create() - Удобный метод для поиска объекта с указанным kwargs (может быть пустым, если в вашей модели есть значения по умолчанию для всех полей), создавая его при необходимости.
    - delete() - Удавить QuerySet.
    - update() - Обновить QuerySet.

- Для представления данных таблицы в виде объектов Python, Django использует интуитивно понятную систему: класс модели представляет таблицу, а экземпляр модели - запись в этой таблице. Чтобы создать объект, создайте экземпляр класса модели, указав необходимые поля в аргументах и вызовите метод save() чтобы сохранить его в базе данных.
- Для сохранения изменений в объект, который уже существует в базе данных, также нужно использовать save().
- Обновление ForeignKey работает так же, как и сохранение обычных полей.
- Есть также методы, через которые можно сохранять данные:
    - bulk_create - этот метод позволяет сохранить в базе данных множество объектов одним запросом.
    - create
- Фильтры полей – это "операторы" для составления условий SQL WHERE. Они задаются как именованные аргументы для метода filter(), exclude() и get() в QuerySet.
- Фильтры полей выглядят как field__lookuptype=value. Используется двойное подчеркивание.
- Поля, указанные при фильтрации, должны быть полями модели. Есть одно исключение - для поля ForeignKey можно указать поле с суффиксом _id. В этом случае необходимо передать значение первичного ключа связанной модели.
- При передаче неверного именованного аргумента, будет вызвано исключение TypeError.
- API базы данных поддерживает около двух дюжин фильтров. Вот пример самых используемых фильтров:
    - exact - точное совпадение. Если передано значение None, оно будет интерпретировано как SQL NULL
    - contains - регистрозависимая проверка на вхождение.
    - in - проверяет на вхождение в список значений.
    - gt - больше чем.
    - gte - больше чем или равно.
    - lt - меньше чем.
    - lte - меньше чем или равно.
    - startswith - регистрозависимая проверка того, начинается ли поле с указанного значения.
    - endswith - регистрозависимая проверка того, оканчивается ли поле с указанного значения.
    - range - проверка на вхождение в диапазон (включающий).
    - search - полнотекстовый поиск, который использует преимущества полнотекстового индекса. Работает как и contains, но значительно быстрее, благодаря полнотекстовому индексу.
    - year, month, day, week_day, hour, minute, second - проверка времени для поля времени.
    - isnull - принимает True или False, что соответствует SQL запросу IS NULL и IS NOT NULL, соответственно.
    - regex - регистрозависимая проверка регулярным выражением.
- Django предлагает удобный и понятный интерфейс для фильтрации по связанным объектам, самостоятельно заботясь о JOIN в SQL. Для фильтра по полю из связанных

CyberBionic systematics
CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

моделей, используйте имена связывающих полей, разделенных двойным нижним подчеркиванием, пока вы не достигните нужного поля.

- Для сортировки используется метод order_by:
    - По умолчанию, результат возвращаемый QuerySet, отсортирован по полям, указанным в аргументе ordering класса Meta модели. Вы можете переопределить сортировку, используя метод order_by.
    - Знак “минус” в "-pub_date" указывает на “нисходящую” сортировку. Сортировка по возрастанию подразумевается по умолчанию.
    - Чтобы отсортировать случайно - используйте "?".
      Заметка: запрос с order_by('?') может быть медленным и сильно нагружать базу данных, зависит от типа базы данных, которую вы используете.
    - Для сортировки по полю из другой модели, используйте синтаксис, аналогичный тому, который используется при фильтрации по полям связанной модели. То есть, название поля, далее два нижних подчеркивания (__), и имя поля в новой модели, и так далее.
    - Нет способа указать должна ли сортировка учитывать регистр. Поэтому Django возвращает результат в таком порядке, в каком его вернула используемая база данных. Можно отсортировать по полю преобразовав значение в нижний регистр, используя Lower.
    - Если вы не хотите использовать сортировку, даже указанную по умолчанию, выполните метод order_by() без аргументов.
- Именованные аргументы функции filter() и др. – объединяются оператором “AND”. Если вам нужны более сложные запросы (например, запросы с оператором OR), вы можете использовать объекты Q.
    - **Объект Q** (django.db.models.Q) – объект, используемый для инкапсуляции множества именованных аргументов для фильтрации. Аргументы определяются так же.
    - Объекты Q могут быть объединены операторами & и |, при этом будет создан новый объект Q.
    - Вы можете комбинировать различные объекты Q с операторами & и |, и использовать скобки. Можно использовать оператор ~ для отрицания (NOT) в запросе.
    - Каждый метод для фильтрации, который принимает именованные аргументы (например, filter(), exclude(), get()get()) также может принимать объекты Q. Если вы передадите несколько объектов Q как аргументы, они будут объединены оператором “AND”.
    - Вы можете использовать одновременно объекты Q и именованные аргументы. Все аргументы (будь то именованные аргументы или объекты Q) объединяются оператором “AND”. Однако, если присутствует объект Q, он должен следовать перед именованными аргументами.
- Одна из сильных сторон Django – это автоматический интерфейс администратора. Он использует мета-данные модели, чтобы предоставить многофункциональный, готовый к использованию интерфейс для работы с содержимым сайта.
- Чтобы добавить модель в админ-панель используем admin.site.register(Author).

Закрепление материала

- Что такое QuerySet?
- Какие есть действия для вычисления QuerySet?
- Какие есть методы для работы с QuerySet?
- Какие есть фильтры?
- Когда нужно использовать объект Q?

CyberBionic systematics
CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Для чего служит админ-панель?

Дополнительное задание

Задание
Вытягивать данные из созданных моделей.

Вытянуть данные, которые начинаются на большую букву “Л”

Вытянуть данные, которые включают цифру.

Отсортировать данные по дате создания.

Самостоятельная деятельность учащегося

Задание 1
Изучить и понять все преимущества и недостатки инструментов, которые были рассмотрены на уроке.

Задание 2
Создать новые модели, которые будут иметь отношение много-к-многим, запустить миграции и попробовать использовать все те фильтры, которые были рассмотрены на уроке.

Задание 3
Попрактиковаться в вытягивание данных. Вытянуть данные и отправить их на страницу с помощью запроса GET.

Рекомендуемые ресурсы

Официальная документация Django:
https://www.djangoproject.com/

MongoDB документация:
https://django-mongodb-engine.readthedocs.io/en/latest/topics/setup.html

PostgreSQL документация:
https://www.postgresql.org/download/windows/

CyberBionic systematics
CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com