Способы кастомизации модели пользователя

№ урока:	2	**Курс:**	Django Essential

Средства обучения:	Pycharm Community, GIT, аккаунт в GitHub, Python 3.6 и выше, Django проект, Docker образ.

Обзор, цель и назначение урока

Научиться настраивать модель пользователя, создавать разные группы пользователей. Рассмотреть методы изменения аутентификации и способы присоединения дополнительных данных пользователя.

Изучив материал данного занятия, учащийся сможет:

- Создать менеджера модели.
- Создать proxy модель.
- Создавать сигналы в Django.
- Создать профайл пользователя.
- Настроить AbstractBaseUser под свои требования.
- Создать свой AbstractUser.
- Добавить картинку к профилю пользователя.

Содержание урока

1. Запуск проекта
2. Создание суперпользователя
3. Создание менеджера
4. Создание базового пользователя
5. Создание proxy
6. Создание профиля пользователя
7. Сигналы. Что это и зачем
8. Создание своего AbstractBaseUser и AbstractUser
9. Рассмотрение тонкостей при расширении с помощью абстрактных моделей
10. Добавление к профилю картинок

Резюме

- python manage.py createsuperuser - создать суперпользователя. Несколько слов о безопасности и выборе паролей.
- **Менеджер** — это интерфейс, через который создаются запросы для работы с моделью в Django. Делаем код универсальным и более читабельным.
- **Proxy** - "представитель" модели. Самая простая стратегия расширения модели пользователя. Использовать для изменения поведения модели.
- Сигналы. Основные типы: pre_save, post_save, pre_delete, post_delete, m2m_changed, request_started и request_finished.
- **User profile** - личное пространство пользователя. Использовать, когда надо сохранять много информации. Самый безопасный и простой подход.
- **AbstractBaseUser** - косвенно меняем исходники Django. Использовать, когда нужно изменить аутентификацию пользователей, или когда не подходит процесс работы с пользователями, который идет "из коробки" Django.

CyberBionic systematics

CyberBionic Systematics ®
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

- Mixin классы (Классы-примеси). Mixin представляет собой набор свойств и методов, которые могут быть использованы в различных классах, которые не приходят из базового класса. В объектно - ориентированных языках программирования, вы, как правило, использовать наследование , чтобы дать объекты различных классов ту же функциональность; если множество объектов есть некоторые способности, вы помещаете эту способность в базовом классе , что оба объекта унаследованы из.
- 4 правила использования AbstractBaseUser:
    - USERNAME_FIELD - описывает имя поля в модели, должен быть уникальным.
    - is_active - boolean параметр, описывает является ли пользователь "активным".
    - get_full_name() - строка, которая описывает пользователя.
    - get_short_name() - имя или ник пользователя.
- AbstractUser - сабклас AbstractBaseUser. Полная версия AbstractBaseUser. Использовать также, как и AbstractBaseUser.
- ImageField - поле для ссылки на изображения.

Закрепление материала

- Что такое менеджер в Django?
- В каких случаях стоит использовать связь один-к-одному с моделью пользователя?
- Какие есть типы сигналов в Django?
- Чем отличается расширение AbstractBaseUser от AbstractUser?
- Назовите правила наследования от AbstractBaseUser.

Дополнительное задание

Задание
Создать пользователя с профайлом и дополнительными данными.

Самостоятельная деятельность учащегося

Задание 1
Выучить и понять все преимущества и недостатки от использования подходов, которые были рассмотрены на уроке.

Задание 2
Расширить менеджера: добавить методы для удаления пользователя, фильтрации и вытягивание всех пользователей.

Задание 3
Создать возможность для пользователя прикреплять к профилю другие файлы. Например, музыку или документы.

Рекомендуемые ресурсы

Django документация:

- https://docs.djangoproject.com/en/2.2/topics/auth/customizing/#extending-the-existing-user-model
- https://tutorial.djangogirls.org/ru/django_admin/
- https://docs.djangoproject.com/en/2.2/topics/signals/
- https://docs.djangoproject.com/en/2.2/ref/models/fields/
- https://docs.djangoproject.com/en/2.2/topics/db/managers/

CyberBionic Systematics ®
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

CyberBionic Systematics ®
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