FRAMEWORKS

Что такое framework и с чем его едят

FRAMEWORK

- Набор классов и функций для решения типовых задач
- Заранее определенная структура проекта согласно паттерну проектирования
- Упрощает и ускоряет разработку конечного продукта
  - Сайт
  - Приложение
  - Чат-Бот
  - прочее

Создание проекта

- django-admin startproject project_name – создает базовую минимальную стрктуру проекта
- python manage.py startapp app_name – создает минимальную структуру приложения. * после создания приложения, его необходимо внести в список installed_apps в файле settings.py

DJANGO ORM
(ORM – Object-Relational Mapper)

Для описания структуры таблицы, необходимо описать модель (класс), дочерний от базовой модели Model
Атрибуты класса описывают атрибуты таблицы

Каждый атрибут класса должен иметь в качестве Значения определенный класс, соответсвующий Типу данных в РСУБД

Поля имеют общие аргументы, а так же аргументы Свойственные конкретному полю

```python
from django.db import models


class Category(models.Model):
    name = models.CharField(
        max_length=32,
        verbose_name='категория',
        unique=True
    )
    descr = models.CharField(
        max_length=1024,
        verbose_name='описание',
        null=True,
        blank=True
    )
```

Общие атрибуты

verbose_name – удобочитаемое имя, принято писать маленькими буквами
null – может ли атрибут хранить значение NULL
blank – может ли атрибут быть пустым
unique – должны ли значения в РСУБД в данном поле быть уникальными
choices – кортеж кортежей по 2 значения, добавляет выпадающий список значений для выбора
default – значение по умолчанию
help_text – справочный текст используемый с виджетом формы
primary_key – первичный ключ (если необходимо переопределить, по умолчанию Django самостоятельно описывает в каждой моделе первичный ключ с типом BigAutoField)

Meta

Для более тонкой настройки моделе, можно определить класс Meta внутри модели

Основные атрибуты:

abstract: bool – является ли модель базовым абстрактным классом

app_label: str – указание приложения к которому принаджелит модель, если она описана вне приложения

db_table: str – указание имени модели в РСУБД, по умолчанию Django формирует его как app_name_class_name

managed: bool – управляет ли Django „жизнью“ данной модели, используется в значении False если модель была создана стороним приложением

ordering: list[str] – список имен атрибутов по которым будет осуществляться сортировка в админ панели, если к имени в начале добавить „-“, это будет указывать на обратную сортировку

verbose_name: str – удобочитаемое имя модели в ед. Ч

verbose_name_plural: str – удобочитаемое имя модели во мн. ч

Связи таблиц

В Django используется немного свой подход к реализации типов связи
Для простой связи ManyToOne используется класс ForeignKey
Для реализации связи OneToOne используется класс OneToOneField (вы не можете указать атрибут unique=True в ForeignKey для реализации связи OneToOne)
Для реализации связи ManyToMany так же используется собственный класс ManyToManyField, в результате Django самостоятельно создает дополнительную таблицу для представления данного типа связи

Все данные классы обязаны принимать 2 обязательных атрибута:
to – ссылка на модель на которую ссылаемся, может быть представлен в виде ссылки или имени класса в виде строки (удобно использовать если модель находится в другом приложении или описана ниже текущей модели)

По умолчанию Django ссылается на атрибут id, если необходимо переопределить атрибут на который мы ссылаемся, необходимо передать атрибут to_field (ВАЖНО! Атрибут на который мы ссылаемся, обязан иметь unique=True)

***Имя атрибута который ссылается необходимо указывать без „_id“ Django допишет данный постфикс самостоятельно (в противном случае вы получите атрибут прим: category_id_id)**

Миграции

Для проведения миграций, необходимо выполнить 2 команды

- **python manage.py makemigrations – генерирует файлы миграции (приложение в котором описаны модели обязательно должно быть зарегестрировано, иначе Django не увидит их)**
- **python manage.py migrate – применение миграций**

При проведении первых миграций, Django создаст собственный таблицы необходимые для регистрации/авторизации и тд

Создание суперпользователя

- **python manage.py createsuperuser – суперпользователь это ваш администратор**

Запуск отладочного сервера

- **python manage.py runserver**

Сервер будет доступен по адресу http://127.0.0.1:8000

СКАЗКИ НА НОЧЬ

EN:

https://docs.djangoproject.com/en/4.1/topics/db/models/

https://docs.djangoproject.com/en/4.1/ref/models/options/

RUS:

https://django.fun/ru/docs/django/4.1/topics/db/models/

https://django.fun/ru/docs/django/4.1/ref/models/options/