BackEnd Workshop

Clarusway

Serializers

Task

- Create serializers for:
  - Artist,
  - Album (artists with the names),
  - Song (artist and album with the name).
- Create nested serializers for:
  - Songs with lyrics,
  - Albums with the songs and `number of the songs` in that album.
- Create views for each serializer (CRUD for first three serializer; only list for last two)
- Create urls for each view

Models

- Use models below:

```
from django.db import models


class Artist(models.Model):
    first_name = models.CharField(max_length=50, null = False, blank = True)
    last_name = models.CharField(max_length=50, null = True, blank = True)
    artist_pic = models.ImageField(upload_to='artists')
    num_stars = models.IntegerField(default=0, blank = True)

    def __str__(self):
        return self.first_name


class Album(models.Model):
    artist = models.ManyToManyField(Artist)
    name = models.CharField(max_length=100)
    released = models.DateField(null = True, blank=True)
    cover = models.ImageField(upload_to='covers')

    def __str__(self):
        return self.name


class Lyric(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField(null = False, blank=True)

    def __str__(self):
        return self.title


class Song(models.Model):
    album = models.ForeignKey(Album, on_delete=models.CASCADE,
related_name='songs')
    artist = models.ForeignKey(Artist, on_delete=models.CASCADE)
    lyric = models.OneToOneField(Lyric, on_delete=models.CASCADE, null=True)
    name = models.CharField(max_length=100)
    released = models.DateField(null = True, blank=True)

    def __str__(self):
        return self.name
```

☺ Thanks for Attending ✍

Clarusway