1분 스피치

주간 회의 발표 자료

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

안녕하세요. 오늘은 Django 프레임워크의 QuerySet을 이용한 데이터 조작에 대해 이야기해 보겠습니다.

Django는 데이터베이스를 다루기 위해 QuerySet이라는 개념을 제공합니다. QuerySet은 데이터베이스의 데이터를 다루기 위한 객체입니다. QuerySet을 사용하면 데이터베이스의 데이터를 쉽게 조작할 수 있습니다.

QuerySet을 사용할 때는 다양한 메서드를 사용할 수 있습니다. 예를 들어, 데이터를 정렬할 때는 `order\_by()` 메서드를 사용합니다. 예를 들어, `Student.objects.order\_by('age')`를 사용하면 데이터를 나이 순으로 정렬할 수 있습니다.

또한, 데이터를 삭제할 때는 `delete()` 메서드를 사용합니다. 예를 들어, `Student.objects.delete()`를 사용하면 데이터를 모두 삭제할 수 있습니다.

QuerySet을 사용할 때는 `filter()` 메서드를 사용하여 데이터를 필터링할 수 있습니다. 예를 들어, `Student.objects.filter(name='kim')`을 사용하면 이름이 'kim'인 데이터를 모두 가져올 수 있습니다.

또한, `get()` 메서드를 사용하여 데이터를 하나만 가져올 수 있습니다. 예를 들어, `Student.objects.get(name='kim')`을 사용하면 이름이 'kim'인 데이터를 하나만 가져올 수 있습니다.

QuerySet을 사용할 때는 다양한 메서드를 사용할 수 있습니다. 예를 들어, `\_\_lt()`, `\_\_lte()`, `\_\_gt()`, `\_\_gte()` 메서드를 사용하여 데이터를 비교할 수 있습니다. 예를 들어, `Student.objects.filter(age\_\_lt=20)`을 사용하면 나이가 20보다 작은 데이터를 모두 가져올 수 있습니다.

또한, `\_\_isnull()` 메서드를 사용하여 null 값을 필터링할 수 있습니다. 예를 들어, `Student.objects.filter(name\_\_isnull=True)`을 사용하면 이름이 null인 데이터를 모두 가져올 수 있습니다.

마지막으로, `\_\_contains()` 메서드를 사용하여 특정 문자열을 포함하는 데이터를 필터링할 수 있습니다. 예를 들어, `Student.objects.filter(name\_\_contains='kim')`을 사용하면 이름에 'kim'이 포함된 데이터를 모두 가져올 수 있습니다.

QuerySet을 사용하면 데이터베이스의 데이터를 쉽게 조작할 수 있습니다. 다양한 메서드를 사용하여 데이터를 정렬, 삭제, 필터링, 비교할 수 있습니다. QuerySet을 사용하면 데이터베이스의 데이터를 더 효율적으로 다룰 수 있습니다.

작성일: 2025년 08월 12일