

Querysets

Objekte filtern

Definition

Ein Queryset repräsentiert eine Sammlung von Objekten aus der Datenbank. Auf ein Queryset können Filter angewandt werden.

Mithilfe von Filtern lässt sich die Ergebnismenge eingrenzen.

Ein Queryset wird intern in ein SQL-Statement umgeformt und die Filter-Methoden werden als WHERE oder LIMIT interpretiert.

Queryset erstellen

Ein Queryset wird durch einen Model-Manager erstellt.

```
qs = Company.objects.all()
```

```
qs
```

```
<QuerySet [{  
    'id': 1,  
    'name': 'Superduper AG',  
    'description': 'eine IT  
    Firma aus München',  
    'number_of_employees': 5,  
    'company_type':  
    'tech',  
    'sub_title': None}]>
```

Queryset Elemente addressieren

Die Elemente eines Querysets können über den **Index** adressiert werden.

```
company = qs[0]
```

Damit erhalten wir das Company-Objekt an **Index 0**. Das Queryset verhält sich also wie eine Liste.

Ein Iterieren über in Queryset ist somit auch möglich. Allerdings ist ein QS selbst kein Iterator (implementiert keine `__next__()`-Methode).

SQL eines Querysets betrachten

Ein **Queryset** wird immer als SQL-Statement interpretiert, wenn es ausgeführt wird. Um sich dieses generierte SQL zu betrachten, können wir das **query**-Attribut des Queryset-Objekts aufrufen.

```
print(qs.query)
```

```
SELECT "company_company"."id", "company_company"."name", "company_company"."description",
"company_company"."number_of_employees", "company_company"."company_type",
"company_company"."sub_title" FROM "company_company"
```

Filtern

Wir können das Queryset mit diversen Filtern eingrenzen.

<https://docs.djangoproject.com/en/3.1/ref/models/querysets/>

Um zum Beispiel alle Objekte eines Querysets zu filtern, deren **Name-Attribut** mit einem "A" anfängt, könnte man folgenden Filter verwenden

```
qs = qs.filter(name__startswith="A")
```

Das **angesprochene Feld** wird mit zwei Underscores von dem **Filter-Attribut startswith** getrennt.

Field Lookups

Es stehen neben `startswith` eine ganze Reihe von sogenannten Field-Lookups zur Verfügung.

Zb.

```
Entry.objects.get(id__exact=None)
```

```
Entry.objects.get(headline__contains='Lennon')
```

```
Entry.objects.filter(headline__in='abc')
```

```
Entry.objects.filter(id__gt=4)
```

<https://docs.djangoproject.com/en/4.0/ref/models/querysets/#field-lookups>