

DjangoのModelからデータを取り出す方法をまとめとく

個人的まとめ

django

作業メモ

Djangoのmodelからデータを取り出す方法をまとめとく

参考)

[クエリを作成する](#) | [Django documentation](#) | [Django](#)

※こっちのほうがいい感じにまとまっているかも

[Django データベース操作 についてのまとめ - Qiita](#)

データ構造

```
from django.db import models

class Blog(models.Model):
    name = models.CharField(max_length=100)
    tagline = models.TextField()

    def __str__(self):
        return self.name

class Author(models.Model):
    name = models.CharField(max_length=200)
    email = models.EmailField()

    def __str__(self):
        return self.name

class Entry(models.Model):
    blog = models.ForeignKey(Blog, on_delete=models.CASCADE)
    headline = models.CharField(max_length=255)
    body_text = models.TextField()
    pub_date = models.DateField()
    mod_date = models.DateField()
    authors = models.ManyToManyField(Author)
    n_comments = models.IntegerField()
    n_pingbacks = models.IntegerField()
    rating = models.IntegerField()

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

全てのオブジェクトを取得する

```
all_entries = Entry.objects.all()
```

オーダー条件をつける

```
# asc
all_entries = Entry.objects.all().order_by("id")
# desc
all_entries = Entry.objects.all().order_by("-id")
```

```
# 複数条件
all_entries = Entry.objects.all().order_by("-id", "view_order")
```

取得数制限 (limit)

```
entries = Entry.objects.all().order_by("id")[0:10] # 先頭から10件取得
entries = Entry.objects.all().order_by("id")[5:15] # 6番目から10件取得
entries = Entry.objects.all().order_by("id")[0:10:2] # 先頭から10件取得を1件置きに取得する
```

フィルタを用いて特定のオブジェクトを取得する

プライマリーキー検索

```
entry = Entry.objects.get(pk=1)
```

フィールドルックアップ

filter()メソッドに[field__lookuptype=value]の形式で引数を渡す

lookuptypeはこちらを参考

[QuerySet API reference](#) | [Django documentation](#) | [Django](#)

```
entries = Entry.objects.filter(id__gte=2) # idが2以上
entries = Entry.objects.filter(id__exact=2) # idが2 (id=2でも同じ結果)
entries = Entry.objects.filter(body_text__contain="some_text") # body_textに"some_text"という文字列が含まれている
```

リレーションデータをフィールドルックアップの条件にする

filter()メソッドに[relationnamefield__lookuptype=value]の形式で引数を渡す。

[relationname**relationname**field__lookuptype=value]のように、いくらでも深くできる

```
entries = Entry.objects.filter(blog__name__exact='Beatles Blog')
```

逆方向にたどる場合には、model名を小文字にしたものを使う

```
blogs = Blog.objects.filter(entry__headline__contains='Lennon')
```

モデルの値を使ったフィルタ (F()オブジェクト)

比較条件にそのmodelの値を使いたい場合、F()オブジェクトにフィールド名を与えると参照できる。

```
from django.db.models import F
entries = Entry.objects.filter(n_comments__gt=F('n_pingbacks'))
```

```
entries = Entry.objects.filter(n_comments__gt=F('n_pingbacks') * 2) # 数値計算もできる
```

```
entries = Entry.objects.filter(authors__name=F('blog__name')) # 関連モデルのフィールドも参照できる
```

```
from datetime import timedelta
entries = Entry.objects.filter(mod_date__gt=F('pub_date') + timedelta(days=3)) # 日付計算もできる
```

```
F('somefield').bitand(16) # bit演算もできるとのこと
```

文字列の結合はできなかった

```
entries = Entry.objects.filter(authors__name=F('blog__name') + "suffix") # エラー
```

下記のようにする。

参考) [python - Can I use Django F\(\) objects with string concatenation? - Stack Overflow](#)

```
from django.db.models.functions import Concat
from django.db.models import Value
Entry.objects.filter(authors__name=Concat(F("blog__name"), Value("suffix"))) # これでいけた。
```

Qオブジェクトを使ったAND,OR条件指定

and条件

```
entries = Entry.objects.filter(Q(id=1) & Q(n_comments=1)) # idが1かつn_commentsが1のもの
```

or条件

```
entries = Entry.objects.filter(Q(id=1) | Q(n_comments=1)) # idが1もしくはn_commentsが1のもの
```

not条件

```
entries = Entry.objects.filter(~Q(id=1)) # idが1ではないもの
```