MongoEngine

MongoEngine是一个ODM（Object-Document Mapper）库，底层使用Pymongo。

http://mongoengine.org/

要求：Pymongo 3.4+

```
$ pip install mongoengine
```

连接

https://mongoengine-odm.readthedocs.io/guide/connecting.html#guide-connecting

```
from mongoengine import connect

conn_str = 'mongodb://192.168.142.130:27017/mytest'
client = connect(host=conn_str)
```

```
from mongoengine import connect

client = connect('mytest', host='127.0.0.1', port=27017)
```

模型

```
from mongoengine import Document, StringField, IntField

class User(Document):
    name = StringField(required=True, max_length=30)
    age = IntField(default=20, max_value=150, min_value=0)
```

集合名称

```
from mongoengine import Document, StringField, IntField

class User(Document):
    name = StringField(required=True, max_length=30)
    age = IntField(default=20, max_value=150, min_value=0)
    meta = {'collection':'users'} # 指定collection
```

如果类名是`BlogUser`，默认名称为`blog_user`，可以手动指定。

`meta = {'collection':'users'}`手动指定使用的集合。

管理器

类似于Django的管理器，默认名字也叫作objects。操作的方法也很类似。

增

```
from mongoengine import Document, StringField, IntField

class User(Document):
    name = StringField(required=True, max_length=30)
    age = IntField(default=20, max_value=150, min_value=0)
    meta = {'collection':'users'} # 指定collection

u1 = User.objects.create(name='jerry', age=31)

u2 = User()
u2.name = 'tom'
u2.save()
```

users

	_id ObjectId	name String	age Int32
1	6180de7d7e4abfe50579fd6b	"jerry"	31
2	6180de7d7e4abfe50579fd6c	"tom"	20

查所有

```
from mongoengine import Document, StringField, IntField

class User(Document):
    name = StringField(required=True, max_length=30)
    age = IntField(default=20, max_value=150, min_value=0)
    meta = {'collection':'users'} # 指定collection

for u in User.objects:
    print(type(u), u.pk, u.id, u.name, u.age)
```

查一个

```
print(User.objects.get(name='tom'))
print(User.objects(name='tom').get())
```

上面的代码要保证使用get的时候一定得到唯一的文档。

```
print(User.objects.get(age__gt=10))
# MultipleObjectsReturned: 2 or more items returned, instead of 1
# DoesNotExist: User matching query does not exist.
```

文档pk

http://docs.mongoengine.org/guide/document-instances.html#document-ids

每一个文档（集合中的一条记录）都有唯一的id。

不设置主键

如果没有定义主键，那么会自动生成一个主键`_id`，可以使用.id或.pk访问

```
from mongoengine import Document, StringField, IntField

class User(Document):
    name = StringField(required=True, max_length=30)
    age = IntField(default=20, max_value=150, min_value=0)
    meta = {'collection':'users'} # 指定collection

for u in User.objects:
    print(u.pk, u.id, u.name, u.age)
```

```
6180de7d7e4abfe50579fd6b 6180de7d7e4abfe50579fd6b jerry 31
6180de7d7e4abfe50579fd6c 6180de7d7e4abfe50579fd6c tom 20
```

不设置主键，会自动创建一个id属性，作为pk，和_id字段建立映射关系

定义主键

```
from mongoengine import Document, StringField, IntField

class User(Document):
    name = StringField(required=True, max_length=30, primary_key=True)
    age = IntField(default=20, max_value=150, min_value=0)
    meta = {'collection':'users'} # 指定collection

for u in User.objects:
    print(u.pk, u.id, u.name, u.age)
```

```
jerry jerry jerry 31
tom tom tom 20
```

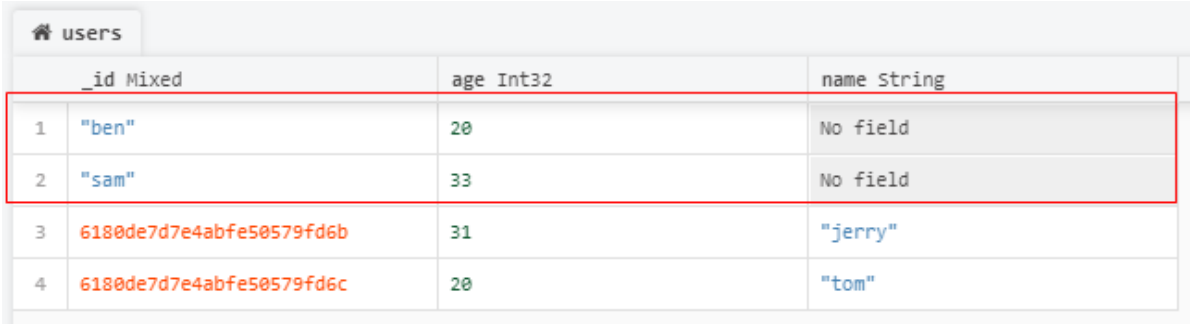
原来的数据保留，新插入数据

```
from mongoengine import Document, StringField, IntField

class User(Document):
    name = StringField(required=True, max_length=30, primary_key=True)
    age = IntField(default=20, max_value=150, min_value=0)
    meta = {'collection':'users'} # 指定collection

u3 = User.objects.create(name='ben')
u4 = User.objects.create(name='sam', age=33)

for u in User.objects:
    print(u.pk, u.id, u.name, u.age)
```



users

	_id Mixed	age Int32	name String
1	"ben"	20	No field
2	"sam"	33	No field
3	6180de7d7e4abfe50579fd6b	31	"jerry"
4	6180de7d7e4abfe50579fd6c	20	"tom"

```
jerry jerry jerry 31
tom tom tom 20
ben ben ben 20
sam sam sam 33
```

指定name为主键pk，自动创建一个id属性直接引用name字段，那么内部就会把_id和name之间的映射关系。也就是在这种情况下，_id、id、name就是同一个字段。

条件查询

查询操作符 https://mongoengine-odm.readthedocs.io/guide/querying.html#query-operators

```
print(*User.objects(age__lt=30), sep='\n')
print(*User.objects(age__not__lt=30), sep='\n') # not
print(*User.objects(name__not__istartswith='T'), sep='\n') # 字符串 ignore
```

排序

```
print(*User.objects().order_by('-pk'), sep='\n')
```

分页

```
print(User.objects.order_by('-pk').first()) # 第一个
print(User.objects.order_by('-pk')[0])
print(*User.objects.order_by('-pk').limit(2), sep='\n')
print(*User.objects.order_by('-pk').limit(2).skip(2), sep='\n')
print(*User.objects[2:4])
```

聚合

```
print(User.objects(age__gt=30).count())
print(User.objects(age__gt=30).sum('age'))
print(User.objects(age__gt=30).average('age'))
```

聚合pipeline

```
pipeline = [
    {'$match':{'age':{'$gt':30}}},
]
print(*User.objects.aggregate(pipeline))

{'_id': ObjectId('6180de7d7e4abfe50579fd6b'), 'name': 'jerry', 'age': 31}
{'_id': 'sam', 'age': 33}
```

```
pipeline = [
    {'$match':{'age':{'$gt':30}}},
    {'$project': {'name':1}}
]
print(*User.objects.aggregate(pipeline))

{'_id': ObjectId('6180de7d7e4abfe50579fd6b'), 'name': 'jerry'} {'_id': 'sam'}
```

Q

```
print(*User.objects(name='tom', age__gt=10))
print(*User.objects(Q(name='tom') & Q(age__gt=10)))

print(*User.objects(Q(age__lt=10) | Q(age__gt=30)))
```

改

参考 https://mongoengine-odm.readthedocs.io/guide/querying.html#atomic-updates

```
user = User.objects.get(name='tom')
print(user) # age:20
user.age += 20
user.save()
print(user) # age:40
```

```
user = User.objects.get(name='tom')
print(user) # age:40
User.objects(name='tom').update_one(inc__age=10)
print(user) # age:40
user.reload() # 一定要reload
print(user) # age:50

<User 6180de7d7e4abfe50579fd6c, tom, 40>
<User 6180de7d7e4abfe50579fd6c, tom, 40>
<User 6180de7d7e4abfe50579fd6c, tom, 50>
```

删

```
users = User.objects(age__in=[20, 33])
print(*users)
users.delete()
print(*User.objects)
```

马哥教育
IT人的高薪职业学院