Решаем сложные задачи в MongoDB просто с Mnj

Serge Matveenko
DataArt

github.com/lig

MongoDB плохо пахнет?

Зато дорого и вкусно :)

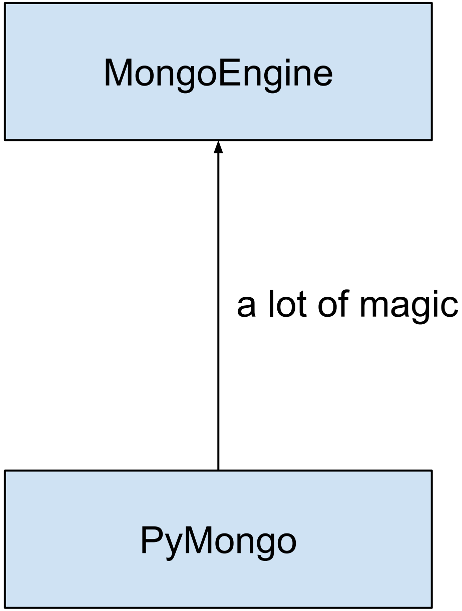
MongoDB state in Python

- PyMongo — main driver
  - Direct MongoDB syntax mapping to dicts.
  - Officially supported
  - Has async derivative “Motor”
- MongoEngine — most popular ODM
  - Class to Collection mapping
  - Django-like syntax
  - Has async derivative “MotorEngine”
- Other
  - TxMongo (Twisted, PyMongo-like)
  - MongoKit (alternative ODM, lack of maintenance, slow, was buggy)
  - Some Django hacks, etc

About Mnj

- “Mnj” stands for “Mongo Energy”
- Query DSL for PyMongo-like API
- Helper Library that implements useful features
- Object Document Mapping (ODM)
- Licensed under BSD 2-clause license
- Well coded and tested (PEP-8, 99% coverage)
- `` `pip install mnj` ``

MongoEngine

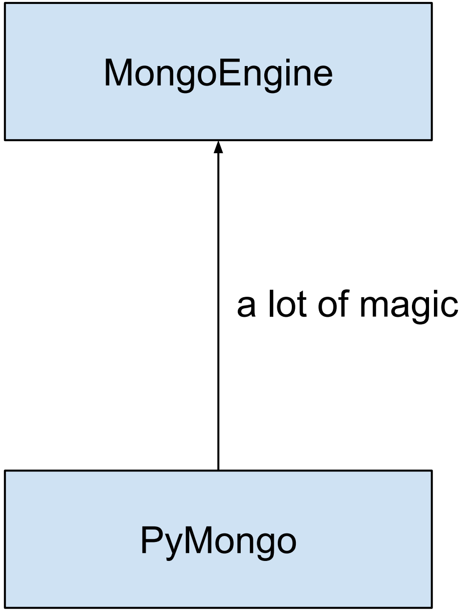


MongoEngine

a lot of magic

PyMongo

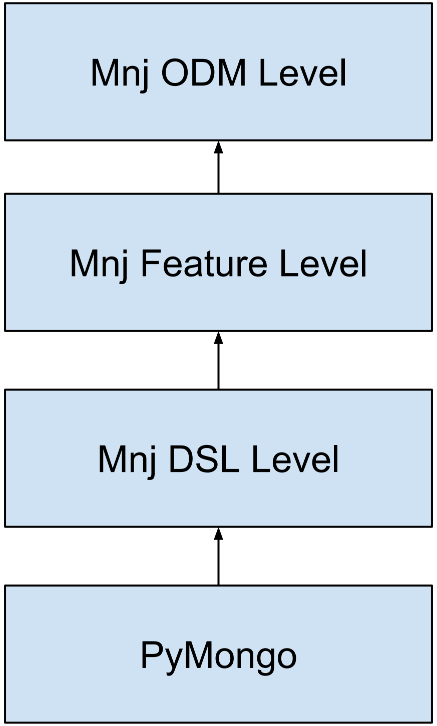
MongoEngine vs Mnj



MongoEngine

a lot of magic

PyMongo



Mnj ODM Level

Mnj Feature Level

Mnj DSL Level

PyMongo

Comparison

	PyMongo	MongoEngine	Mnj
Objects Mapping	-		
Validation	-		
Usability			
MongoDB like			
PyMongo Compatibility			
Defending errors			

PyMongo Example

```python
books = col.find({
    'author': {'$regex': 'Gibson', '$options': 'i'},
    'year': {'$gte': 2003},
})

for book in books:
    print(book['title'])
```

Comparison

	PyMongo	MongoEngine	Mnj
Objects Mapping	-		
Validation	-		
Usability	-		
MongoDB like	+		
PyMongo Compatibility	+ :)		
Defending errors	-		

MongoEngine Example

```
class Book(Document):
    author = StringField()
    year = IntField()
    title = StringField()

books = Book.objects(author__icontains='Gibson', year__gte=2003)

for book in books:
    print(book.title)
```

Comparison

	PyMongo	MongoEngine	Mnj
Objects Mapping	-	+	
Validation	-	+	
Usability	-	±	
MongoDB like	+	-	
PyMongo Compatibility	+ :)	-	
Defending errors	-	±	

Mnj Example

```
books = col.find(q(
    author=regex_('Gibson', re.IGNORECASE),
    year=gte_(2003)
))

for book in books:
    print(book['title'])
```

Mnj mixed with PyMongo Example

```
books = col.find(q(
    q(author=regex_('Gibson', re.IGNORECASE)),
    {'year': {'$gte': 2003}}
))

for book in books:
    print(book['title'])
```

Mnj mixed with PyMongo Example

```
books = col.find(
    q(year=mod_(divisor=1000, remainder=42)) |
    q(year=mod_(1000, 13))
)

for book in books:
    print(book['title'], book['author'])
```

Comparison

	PyMongo	MongoEngine	Mnj
Objects Mapping	-	+	+ unstable
Validation	-	+	+ planned
Usability	-	±	+
MongoDB like	+	-	+
PyMongo Compatibility	+ :)	-	+
Defending errors	-	±	+ improving

Planned Mnj Features

- Map/Reduce and Aggregation framework support
- Object Document Mapping (different types in the same query)
- Document linking/referencing
- Redundant data in document references
- Large documents (transparently split and merge docs>16MB)
- Bucketing (transparently store series in a single document)

How to help

- Provide use cases for improving DSL
  https://pypi.python.org/pypi/mnj
- Star and fork on Github
  https://github.com/lig/mnj
- Implement and contribute features
  https://github.com/lig/mnj/issues
- Join chat and ask questions on Gitter
  https://gitter.im/lig/mnj

Thank you!

github.com/lig/mnj