

2.12. Use mongomock for testing

Although we recommend running your tests against a regular MongoDB server, it is sometimes useful to plug MongoEngine to alternative implementations (mongomock, montydb, mongita, etc).

[mongomock](#) is historically the one suggested for MongoEngine and is a package to do just what the name implies, mocking a mongo database.

To use with mongoengine, simply specify mongomock when connecting with mongoengine:

```
import mongomock

connect('mongoenginetest', host='mongodb://localhost',
mongo_client_class=mongomock.MongoClient)
conn = get_connection()
```

or with an alias:

```
connect('mongoenginetest', host='mongodb://localhost',
mongo_client_class=mongomock.MongoClient, alias='testdb')
conn = get_connection('testdb')
```

2.12.1. Example of test file:

```
import unittest
from mongoengine import connect, disconnect

class Person(Document):
    name = StringField()

class TestPerson(unittest.TestCase):

    @classmethod
    def setUpClass(cls):
        connect('mongoenginetest', host='mongodb://localhost',
        mongo_client_class=mongomock.MongoClient)

    @classmethod
    def tearDownClass(cls):
        disconnect()

    def test_thing(self):
        pers = Person(name='John')
        pers.save()

        fresh_pers = Person.objects().first()
        assert fresh_pers.name == 'John'
```