

2.2. 连接到 MongoDB

MongoEngine 中的连接是全局注册的，并用别名标识。如果 `alias` 在连接期间没有提供，它将使用“default”作为别名。

要连接到正在运行的 `mongod` 实例，请使用该 `connect()` 函数。第一个参数是要连接的数据库的名称：

```
from mongoengine import connect
connect('project1')
```

默认情况下，MongoEngine 假定 `mongod` 实例在端口 **27017** 上的本地主机上运行。

如果 MongoDB 在别处运行，您需要提供有关如何连接的详细信息。有两种方法可以做到这一点。使用 URI 格式的连接字符串（**这是首选方法**）或作为关键字参数提供的单个属性。

2.2.1. 使用 URI 字符串连接

使用 URI 格式的连接字符串时，您应该将连接详细信息指定为 `host` 到 `connect()`。例如，在 Web 应用程序上下文中，URI 通常是从配置文件中读取的：

```
connect(host="mongodb://127.0.0.1:27017/my_db")
```

如果数据库需要身份验证，您可以在 URI 中指定它。由于每个数据库都可以配置自己的用户，您需要告诉 MongoDB 在哪里寻找您正在使用的用户，这就是 `?authSource=admin` MongoDB 连接字符串的作用：

```
# Connects to 'my_db' database by authenticating
# with given credentials against the 'admin' database (by default as authSource isn't
# provided)
connect(host="mongodb://my_user:my_password@127.0.0.1:27017/my_db")

# Equivalent to previous connection but explicitly states that
# it should use admin as the authentication source database
connect(host="mongodb://my_user:my_password@hostname:port/my_db?authSource=admin")

# Connects to 'my_db' database by authenticating
# with given credentials against that same database
connect(host="mongodb://my_user:my_password@127.0.0.1:27017/my_db?authSource=my_db")
```

URI 字符串还可以用于配置高级参数，如 `ssl`、`replicaSet` 等。有关 URI 字符串的更多信息或示例，您可以参考官方[文档](#)：

```
connect(host="mongodb://my_user:my_password@127.0.0.1:27017/my_db?
authSource=admin&ssl=true&replicaSet=globaldb")
```

❗ 笔记

也可以使用包含 SRV 记录的 URI（例如“`mongodb+srv://server.example.com/`”）

2.2.2. 连接关键字属性

指定连接详细信息的第二个选项是将信息作为关键字属性提供给 `connect()`：

```
connect('my_db', host='127.0.0.1', port=27017)
```

如果数据库需要身份验证，`username` 则应 `password` 提供 `authentication_source` 参数：

```
connect('my_db', username='my_user', password='my_password',
authentication_source='admin')
```

识别的属性集 `connect()` 包括但不限于：`host`，`port`，`read_preference`，`username`，`password`，`authentication_source`，`authentication_mechanism`，`replicaset` 等。

`pymongo.MongoClient` `tls` 接受的大多数参数 都可以使用，并且在实例化 `pymongo.MongoClient` 时将简单地转发。 `connect()`

❗ 笔记

URI 字符串中的数据库、用户名和密码会覆盖 中的相应参数 `connect()`，这显然应该避免：

```
connect(
    db='test',
    username='user',
    password='12345',
    host='mongodb://admin:qwerty@localhost/production'
)
```

`production` 将使用 `admin` 用户名和密码建立与数据库的连接 `qwerty`。

❗ 笔记

默认情况下，不带参数的调用 `connect()` 将建立与“测试”数据库的连接

2.2.3. 阅读首选项

如上所述，通过连接支持读取首选项，但也通过传递 `read_preference` 的单个查询支持读取首选项

```
from pymongo import ReadPreference

Bar.objects().read_preference(ReadPreference.PRIMARY)
Bar.objects(read_preference=ReadPreference.PRIMARY)
```

2.2.4. 多个数据库

要使用多个数据库，您可以使用 `connect()` 并提供连接的别名 - 如果没有提供别名，则使用“默认”。

在后台，这用于 `register_connection()` 存储数据，如果需要，您可以预先注册所有别名。

2.2.4.1. 在不同数据库中定义的文档

通过在元数据中提供 `db_alias`，可以将各个文档附加到不同的数据库。这允许 `DBRef` 对象指向跨数据库和集合。下面是一个示例模式，使用 3 个不同的数据库来存储数据：

```

connect(alias='user-db-alias', db='user-db')
connect(alias='book-db-alias', db='book-db')
connect(alias='users-books-db-alias', db='users-books-db')

class User(Document):
    name = StringField()

    meta = {'db_alias': 'user-db-alias'}

class Book(Document):
    name = StringField()

    meta = {'db_alias': 'book-db-alias'}

class AuthorBooks(Document):
    author = ReferenceField(User)
    book = ReferenceField(Book)

    meta = {'db_alias': 'users-books-db-alias'}

```

2.2.4.2. 断开现有连接

该函数 `disconnect()` 可用于断开特定连接。这可用于全局更改连接：

```

from mongoengine import connect, disconnect
connect('a_db', alias='db1')

class User(Document):
    name = StringField()
    meta = {'db_alias': 'db1'}

disconnect(alias='db1')

connect('another_db', alias='db1')

```

❗ 笔记

不带参数调用 `disconnect()` 将断开“默认”连接

❗ 笔记

由于连接是全局注册的，因此使用MongoEngine 的 *断开连接*功能而不是现有连接 (pymongo.MongoClient) 的`disconnect()`方法很重要

❗ 笔记

`Document` 正在缓存 pymongo 集合。使用 *断开连接*可确保它也得到清洁

2.2.5. 上下文管理器

有时您可能希望切换要查询的数据库或集合。例如，出于性能原因将旧数据归档到单独的数据库中，或者编写动态选择集合以将文档写入其中的函数。

2.2.5.1。切换数据库

上下文 `switch_db` 管理器允许您更改给定类的数据库别名，从而允许跨数据库快速轻松地访问同一用户文档：

```
from mongengine.context_managers import switch_db

class User(Document):
    name = StringField()

    meta = {'db_alias': 'user-db'}

with switch_db(User, 'archive-user-db') as User:
    User(name='Ross').save() # Saves the 'archive-user-db'
```

❗ 笔记

`switch_db()` 当在允许继承的类上使用时，将仅更改给定类实例的数据库别名 - 子类实例仍将使用默认数据库。

2.2.5.2。开关合集

上下文 `switch_collection()` 管理器允许您更改给定类的集合，从而允许快速轻松地跨集合访问同一组文档：

```
from mongengine.context_managers import switch_collection

class Group(Document):
    name = StringField()

Group(name='test').save() # Saves in the default db

with switch_collection(Group, 'group2000') as Group:
    Group(name='hello Group 2000 collection!').save() # Saves in group2000 collection
```

❗ 笔记

确保 在使用上下文管理器之前 `register_connection()` 或之前已经注册了任何别名。

`connect()`