

---

# **TxMongo Documentation**

*Release 15.1*

**Alexandre Fiori, Bret Curtis**

April 30, 2016



<b>1</b>	<b>What is TxMongo</b>	<b>1</b>
<b>2</b>	<b>Quick Usage Example</b>	<b>3</b>
<b>3</b>	<b>User's Guide</b>	<b>5</b>
<b>4</b>	<b>Meta</b>	<b>15</b>
<b>5</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>



---

## What is TxMongo

---

TxMongo is a pure-Python Twisted MongoDB client library that implements:

- Asynchronous client driver to MongoDB

Get it from [PyPI](#), find out what's new in the [Changelog](#)!



---

## Quick Usage Example

---

```
from OpenSSL import SSL
from txmongo.connection import ConnectionPool
from twisted.internet import defer, reactor, ssl

class ServerTLSContext(ssl.DefaultOpenSSLContextFactory):
    def __init__(self, *args, **kw):
        kw['sslmethod'] = SSL.TLSv1_METHOD
        ssl.DefaultOpenSSLContextFactory.__init__(self, *args, **kw)

@defer.inlineCallbacks
def example():
    tls_ctx = ServerTLSContext(privateKeyFileName='./mongodb.key', certificateFileName='./mongodb.crt')
    mongodb_uri = "mongodb://localhost:27017"

    mongo = yield ConnectionPool(mongodb_uri, ssl_context_factory=tls_ctx)

    foo = mongo.foo # `foo` database
    test = foo.test # `test` collection

    # fetch some documents
    docs = yield test.find(limit=10)
    for doc in docs:
        print doc

if __name__ == '__main__':
    example().addCallback(lambda ign: reactor.stop())
    reactor.run()
```



## 3.1 txmongo package

### 3.1.1 Submodules

### 3.1.2 txmongo.collection module

```
class txmongo.collection.Collection (database, name, write_concern=None)
    Bases: object

    aggregate (pipeline, full_response=False)
    count (spec=None, fields=None)
    create_index (sort_fields, **kwargs)
    database
    delete_many (filter)
    delete_one (filter)
    distinct (key, spec=None)
    drop (**kwargs)
    drop_index (index_identifier)
    drop_indexes ()
    ensure_index (sort_fields, **kwargs)
    filemd5 (spec)
    find (spec=None, skip=0, limit=0, fields=None, filter=None, cursor=False, **kwargs)
    find_and_modify (query=None, update=None, upsert=False, **kwargs)
    find_one (spec=None, fields=None, **kwargs)
    find_one_and_delete (filter, projection=None, sort=None, **kwargs)
    find_one_and_replace (filter, replacement, projection=None, sort=None, upsert=False, re-
        turn_document=False, **kwargs)
    find_one_and_update (filter, update, projection=None, sort=None, upsert=False, re-
        turn_document=False, **kwargs)
```

**find\_with\_cursor** (*spec=None, skip=0, limit=0, fields=None, filter=None, \*\*kwargs*)  
find method that uses the cursor to only return a block of results at a time. Arguments are the same as with find() returns deferred that results in a tuple: (docs, deferred) where docs are the current page of results and deferred results in the next tuple. When the cursor is exhausted, it will return the tuple ([], None)

**full\_name**

**group** (*keys, initial, reduce, condition=None, finalize=None*)

**index\_information** ()

**insert** (*docs, safe=None, flags=0, \*\*kwargs*)

**insert\_many** (*documents, ordered=True*)

**insert\_one** (*document*)

**map\_reduce** (*map, reduce, full\_response=False, \*\*kwargs*)

**name**

**options** ()

**remove** (*spec, safe=None, single=False, flags=0, \*\*kwargs*)

**rename** (*new\_name*)

**replace\_one** (*filter, replacement, upsert=False*)

**save** (*doc, safe=None, \*\*kwargs*)

**update** (*spec, document, upsert=False, multi=False, safe=None, flags=0, \*\*kwargs*)

**update\_many** (*filter, update, upsert=False*)

**update\_one** (*filter, update, upsert=False*)

**with\_options** (*\*\*kwargs*)  
Get a clone of collection changing the specified settings.

**write\_concern**

### 3.1.3 txmongo.connection module

**class** txmongo.connection.**ConnectionPool** (*uri='mongodb://127.0.0.1:27017', pool\_size=1, ssl\_context\_factory=None, \*\*kwargs*)

Bases: object

**authenticate** (*database, username, password, mechanism='DEFAULT'*)

**disconnect** ()

**get\_default\_database** ()

**getprotocol** ()

**getprotocols** ()

**uri**

**write\_concern**

**class** txmongo.connection.**MongoConnection** (*host='127.0.0.1', port=27017, pool\_size=1, \*\*kwargs*)

Bases: txmongo.connection.ConnectionPool

`txmongo.connection.MongoConnectionPool`  
 alias of `MongoConnection`

`txmongo.connection.lazyMongoConnection`  
 alias of `MongoConnection`

`txmongo.connection.lazyMongoConnectionPool`  
 alias of `MongoConnection`

### 3.1.4 txmongo.database module

**class** `txmongo.database.Database` (*factory, database\_name, write\_concern=None*)  
 Bases: `object`

**authenticate** (*name, password, mechanism='DEFAULT'*)  
 Send an authentication command for this database. mostly stolen from pymongo

**collection\_names** ()

**command** (*command, value=1, check=True, allowable\_errors=None, \*\*kwargs*)

**connection**

**create\_collection** (*name, options=None*)

**drop\_collection** (*name\_or\_collection*)

**name**

**write\_concern**

### 3.1.5 txmongo.filter module

`txmongo.filter.ASCENDING` (*keys*)  
 Ascending sort order

`txmongo.filter.DESCENDING` (*keys*)  
 Descending sort order

`txmongo.filter.GEO2D` (*keys*)  
 Two-dimensional geospatial index <http://www.mongodb.org/display/DOCS/Geospatial+Indexing>

`txmongo.filter.GEO2DSPHERE` (*keys*)  
 Two-dimensional geospatial index <http://www.mongodb.org/display/DOCS/Geospatial+Indexing>

`txmongo.filter.GEOHAYSTACK` (*keys*)  
 Bucket-based geospatial index <http://www.mongodb.org/display/DOCS/Geospatial+Haystack+Indexing>

**class** `txmongo.filter.comment` (*comment*)  
 Bases: `txmongo.filter._QueryFilter`

**class** `txmongo.filter.explain`  
 Bases: `txmongo.filter._QueryFilter`  
 Returns an explain plan for the query.

**class** `txmongo.filter.hint` (*index\_list\_or\_name*)  
 Bases: `txmongo.filter._QueryFilter`  
 Adds a *hint*, telling Mongo the proper index to use for the query.

**class** `txmongo.filter.snapshot`  
Bases: `txmongo.filter._QueryFilter`

**class** `txmongo.filter.sort` (*key\_list*)  
Bases: `txmongo.filter._QueryFilter`  
Sorts the results of a query.

### 3.1.6 txmongo.gridfs module

### 3.1.7 txmongo.protocol module

Low level connection to Mongo.

This module contains the wire protocol implementation for txmongo. The various constants from the protocol are available as constants.

This implementation requires pymongo so that as much of the implementation can be shared. This includes BSON encoding and decoding as well as Exception types, when applicable.

**class** `txmongo.protocol.Delete`  
Bases: `txmongo.protocol.Delete`

**class** `txmongo.protocol.Getmore`  
Bases: `txmongo.protocol.Getmore`

**class** `txmongo.protocol.Insert`  
Bases: `txmongo.protocol.Insert`

**class** `txmongo.protocol.KillCursors`  
Bases: `txmongo.protocol.KillCursors`

**exception** `txmongo.protocol.MongoAuthenticationError`  
Bases: `Exception`

**class** `txmongo.protocol.MongoClientProtocol`  
Bases: `twisted.internet.protocol.Protocol`

**get\_request\_id**()

**send** (*request*)

**send\_DELETE** (*request*)

**send\_GETMORE** (*request*)

**send\_INSERT** (*request*)

**send\_KILL\_CURSORS** (*request*)

**send\_MSG** (*request*)

**send\_QUERY** (*request*)

**send\_REPLY** (*request*)

**send\_UPDATE** (*request*)

**class** `txmongo.protocol.MongoDecoder`  
Bases: `object`

**dataBuffer** = `None`

**static decode** (*message\_data*)

**feed** (*data*)

**next** ()

**class** `txmongo.protocol.MongoProtocol`

Bases: `txmongo.protocol.MongoServerProtocol`, `txmongo.protocol.MongoClientProtocol`

**authenticate** (*database\_name*, *username*, *password*, *mechanism*)

**authenticate\_mongo\_cr** (*database\_name*, *username*, *password*)

**authenticate\_scram\_sha1** (*database\_name*, *username*, *password*)

**connectionLost** (*reason*=<`twisted.python.failure.Failure` `twisted.internet.error.ConnectionDone`:  
Connection was closed cleanly.>)

**connectionMade** ()

**connectionReady** ()

**fail** (*reason*)

**get\_last\_error** (*db*, *\*\*options*)

**handle\_REPLY** (*request*)

**inflight** ()

**max\_wire\_version** = None

**min\_wire\_version** = None

**send\_GETMORE** (*request*)

**send\_QUERY** (*request*)

**set\_wire\_versions** (*min\_wire\_version*, *max\_wire\_version*)

**class** `txmongo.protocol.MongoServerProtocol`

Bases: `twisted.internet.protocol.Protocol`

**dataReceived** (*data*)

**handle** (*request*)

**handle\_DELETE** (*request*)

**handle\_GETMORE** (*request*)

**handle\_INSERT** (*request*)

**handle\_KILL\_CURSORS** (*request*)

**handle\_MSG** (*request*)

**handle\_QUERY** (*request*)

**handle\_REPLY** (*request*)

**handle\_UPDATE** (*request*)

**class** `txmongo.protocol.Msg` (*len*, *request\_id*, *response\_to*, *opcode*, *message*)

Bases: `tuple`

**len**

Alias for field number 0

**message**

Alias for field number 4

**opcode**

Alias for field number 3

**request\_id**

Alias for field number 1

**response\_to**

Alias for field number 2

**class** `txmongo.protocol.Query`  
Bases: `txmongo.protocol.Query`

**class** `txmongo.protocol.Reply`  
Bases: `txmongo.protocol.Reply`

**class** `txmongo.protocol.Update`  
Bases: `txmongo.protocol.Update`

### 3.1.8 Module contents

## 3.2 txmongo.\_gridfs package

### 3.2.1 Submodules

#### 3.2.2 txmongo.\_gridfs.errors module

Exceptions raised by the `gridfs` package

**exception** `txmongo._gridfs.errors.CorruptGridFile`  
Bases: `txmongo._gridfs.errors.GridFSError`

Raised when a file in `GridFS` is malformed.

**exception** `txmongo._gridfs.errors.GridFSError`  
Bases: `Exception`

Base class for all `GridFS` exceptions.

New in version 1.5.

**exception** `txmongo._gridfs.errors.NoFile`  
Bases: `txmongo._gridfs.errors.GridFSError`

Raised when trying to read from a non-existent file.

New in version 1.6.

**exception** `txmongo._gridfs.errors.UnsupportedAPI`  
Bases: `txmongo._gridfs.errors.GridFSError`

Raised when trying to use the old `GridFS` API.

In version 1.6 of the PyMongo distribution there were backwards incompatible changes to the `GridFS` API. Upgrading shouldn't be difficult, but the old API is no longer supported (with no deprecation period). This exception will be raised when attempting to use unsupported constructs from the old API.

New in version 1.6.

### 3.2.3 txmongo.\_gridfs.grid\_file module

Tools for representing files stored in GridFS.

**class** txmongo.\_gridfs.grid\_file.**GridIn**(*root\_collection*, *\*\*kwargs*)

Bases: object

Class to write data to GridFS.

**chunk\_size**

Chunk size for this file.

This attribute is read-only.

**close**()

Flush the file and close it.

A closed file cannot be written any more. Calling *close()* more than once is allowed.

**closed**

Is this file closed?

**content\_type**

Mime-type for this file.

This attribute can only be set before *close()* has been called.

**filename**

Name of this file.

This attribute can only be set before *close()* has been called.

**length**

Length (in bytes) of this file.

This attribute is read-only and can only be read after *close()* has been called.

**md5**

MD5 of the contents of this file (generated on the server).

This attribute is read-only and can only be read after *close()* has been called.

**upload\_date**

Date that this file was uploaded.

This attribute is read-only and can only be read after *close()* has been called.

**write**(*data*)

Write data to the file. There is no return value.

*data* can be either a string of bytes or a file-like object (implementing *read()*).

Due to buffering, the data may not actually be written to the database until the *close()* method is called.

Raises `ValueError` if this file is already closed. Raises `TypeError` if *data* is not an instance of `str` or a file-like object.

#### Parameters

- *data*: string of bytes or file-like object to be written to the file

**writelines**(*sequence*)

Write a sequence of strings to the file.

Does not add separators.

**class** `txmongo._gridfs.grid_file.GridOut` (*root\_collection, doc*)

Bases: `object`

Class to read data out of GridFS.

**aliases**

List of aliases for this file.

This attribute is read-only.

**chunk\_size**

Chunk size for this file.

This attribute is read-only.

**close** ()

**content\_type**

Mime-type for this file.

This attribute is read-only.

**length**

Length (in bytes) of this file.

This attribute is read-only.

**md5**

MD5 of the contents of this file (generated on the server).

This attribute is read-only.

**metadata**

Metadata attached to this file.

This attribute is read-only.

**name**

Name of this file.

This attribute is read-only.

**read** (*size=-1*)

Read at most *size* bytes from the file (less if there isn't enough data).

The bytes are returned as an instance of `str`. If *size* is negative or omitted all data is read.

**Parameters**

- *size* (optional): the number of bytes to read

**seek** (*pos, whence=0*)

Set the current position of this file.

**Parameters**

- *pos*: the position (or offset if using relative positioning) to seek to
- *whence* (optional): where to seek from. `os.SEEK_SET` (0) for absolute file positioning, `os.SEEK_CUR` (1) to seek relative to the current position, `os.SEEK_END` (2) to seek relative to the file's end.

**tell** ()

Return the current position of this file.

**upload\_date**

Date that this file was first uploaded.

This attribute is read-only.

**class** txmongo.\_gridfs.grid\_file.**GridOutIterator** (*grid\_out, chunks*)

Bases: object

**next** ()

### 3.2.4 Module contents

GridFS is a specification for storing large objects in Mongo.

The `gridfs` package is an implementation of GridFS on top of `pymongo`, exposing a file-like interface.

**class** txmongo.\_gridfs.**GridFS** (*database, collection='fs'*)

Bases: object

An instance of GridFS on top of a single Database.

**delete** (*file\_id*)

Delete a file from GridFS by "`_id`".

Removes all data belonging to the file with "`_id`": *file\_id*.

**Warning:** Any processes/threads reading from the file while this method is executing will likely see an invalid/corrupt file. Care should be taken to avoid concurrent reads to a file while it is being deleted.

#### Parameters

- *file\_id*: "`_id`" of the file to delete

New in version 1.6.

**get** (*file\_id*)

Get a file from GridFS by "`_id`".

Returns an instance of `GridOut`, which provides a file-like interface for reading.

#### Parameters

- *file\_id*: "`_id`" of the file to get

New in version 1.6.

**get\_last\_version** (*filename*)

Get a file from GridFS by "`filename`".

Returns the most recently uploaded file in GridFS with the name *filename* as an instance of `GridOut`. Raises `NoFile` if no such file exists.

An index on `{filename: 1, uploadDate: -1}` will automatically be created when this method is called the first time.

#### Parameters

- *filename*: "`filename`" of the file to get

New in version 1.6.

**list** ()

List the names of all files stored in this instance of `GridFS`.

Changed in version 1.6: Removed the *collection* argument.

**new\_file** (\*\*kwargs)

Create a new file in GridFS.

Returns a new `GridIn` instance to which data can be written. Any keyword arguments will be passed through to `GridIn()`.

**Parameters**

- **kwargs** (optional): keyword arguments for file creation

New in version 1.6.

**put** (data, \*\*kwargs)

Put data in GridFS as a new file.

Equivalent to doing:

```
>>> f = new_file(**kwargs)
>>> try:
>>>     f.write(data)
>>> finally:
>>>     f.close()
```

*data* can be either an instance of `str` or a file-like object providing a `read()` method. Any keyword arguments will be passed through to the created file - see `GridIn()` for possible arguments. Returns the `"_id"` of the created file.

**Parameters**

- *data*: data to be written as a file.
- **kwargs** (optional): keyword arguments for file creation

New in version 1.6.

## 4.1 Changelog

### 4.1.1 Release 15.2 (2015-09-05)

This release makes TxMongo fully Python3 compatible and has an API change that breaks older TxMongo compatibility by bringing it inline with PyMongo.

#### API Changes

- `txmongo.dbref` removed. Use `bson.dbref` instead. **Incompatibility note:** `bson.dbref.DBRef` takes collection name as string while `txmongo.dbref.DBRef` was able to accept `Collection` instance. Please use `collection.name` instead.
- Added `timeout` parameter for `connection.ConnectionPool` that can be passed on to Twisted's `connectTCP` and `connectSSL` methods.

#### Features

- `name`, `full_name` and `database` properties of `Collection`
- Python3 compatible.

### 4.1.2 Release 15.1 (2015-06-08)

This is a major release in that while increasing code coverage to 95% ( see <https://coveralls.io/builds/2749499> ), we've also caught several bugs, added features and changed functionality to be more inline with PyMongo.

This is no small thanks to travis-ci and coveralls while using tox to cover all iterations that we support.

We can officially say that we are Python 2.6, 2.7 and PyPy compatible.

#### API Changes

- **TxMongo now requires PyMongo 3.x**, if you need PyMongo 2.x support, please use 15.0, otherwise it is highly recommended to use PyMongo 3.x which still supports MongoDB 2.6.
- Better handling of replica-sets, we now raise an `autoreconnect` when master is unreachable.

- Changed the behaviour of `find_one` to return `None` instead of an empty dict `{}` when no result is found.
- New-style query methods: `insert_one/many`, `update_one/many`, `delete_one/many`, `replace_one` and `find_one_and_update/replace`

### Features

- Added `db.command` function, just like PyMongo.
- Added support for named indexes in `filter`.
- `insert()`, `update()`, `save()` and `remove()` now support write-concern options via named args: `w`, `wtimeout`, `j`, `fsync`. `safe` argument is still supported for backward compatibility.
- Default write-concern can be specified for `Connection` using named arguments in constructor or by URI options.
- Write-concern options can also be set for `Database` and `Collection` with `write_concern` named argument of their constructors. In this case write-concern is specified by instance of `pymongo.write_concern.WriteConcern`
- `txmongo.protocol.INSERT_CONTINUE_ON_ERROR` flag defined for using with `insert()`
- Replaced all traditional deferred callbacks (and errbacks) to use `@defer.inlineCallbacks`

### Bugfixes

- Fixed typo in `map_reduce()` when returning results.
- Fixed hang in `create_collection()` in case of error.
- Fixed typo in `rename()` that wasn't using the right factory.
- Fixed exception in `drop_index` that was being thrown when dropping a non-existent collection. This makes the function idempotent.
- Fixed URI prefixing when “`mongodb://`” is not present in URI string in `connection`.
- Fixed fail-over when using replica-sets in `connection`. It now raises `autoreconnect` when there is a problem with the existing master. It is then up to the client code to reconnect to the new master.
- Fixed number of cursors in protocol so that it works with py2.6, py2.6 and pypy.

### 4.1.3 Release 15.0 (2015-05-04)

This is the first release using the Twisted versioning method.

### API Changes

- `collections.index_information` now mirrors PyMongo's method.
- `getrequestid` is now `get_request_id`

## Features

- Add support for 2dsphere indexes, see <http://docs.mongodb.org/manual/tutorial/build-a-2dsphere-index/>
- PEP8 across files as we work through them.
- Authentication reimplemented for ConnectionPool support with multiple DBs.
- Add support for MongoDB 3.0

## Bugfixes

- Fixed failing tests due to changes in Python in 2.6
- Fixed limit not being respected, which should help performance.
- Find now closes MongoDB cursors.
- Fixed 'hint' filter to correctly serialize with double dollar signs.

## Improved Documentation

- Added, updated and reworked documentation using Sphinx.
- The documentation is now hosted on <https://txmongo.readthedocs.org/>.

### 4.1.4 Release 0.6 (2015-01-23)

This is the last release in this version scheme, we'll be switching to the Twisted version scheme in the next release.

## API Changes

- TxMongo: None

## Features

- Added SSL support using Twisted SSLContext factory
- Added "find with cursor" like pymongo
- Test coverage is now measured. We're currently at around 78%.

## Bugfixes

- Fixed import in database.py

### 4.1.5 Release 0.5 (2014-10-02)

Code review and cleanup

## Bugfixes

- Bug fixes

### 4.1.6 Release 0.4 (2013-01-07)

Significant performance improvements.

#### API Changes

- TxMongo: None

#### Features

- Support AutoReconnect to connect to fail-over master.
- Use pymongo instead of in-tree copy.

#### Bugfixes

- Bug fixes

### 4.1.7 Release 0.3 (2010-09-13)

Initial release.

#### License

- Apache 2.0

## 4.2 Status and History

TxMongo was created by [Alexandre Fiori](#) who developed it during the years 2009-2010. From 2010 and onwards mainly bug fixes were added, as Alexandre entered maintenance mode with many contributions being made by others. Development picked back up in 2014 by [Bret Curtis](#) with Alexandre's consent and was migrated to Twisted where it is a first-party Twisted library. TxMongo can be found here:

<https://github.com/twisted/txmongo>

The MongoDB client library functionality is in active use. It is stable and works very well.

## 4.3 Contributions

### 4.3.1 How to Contribute

Head over to: <https://github.com/twisted/TxMongo> and submit your bugs or feature requests. If you wish to contribute code, just fork it, make a branch and send us a pull request. We'll review it, and push back if necessary.

TxMongo generally follows the coding and documentation standards of the Twisted project.

### 4.3.2 Contributors

- 10gen, Inc
- Alexandre Fiori
- Alexey Palazhchenko (AlekSi)
- Amplidata
- Andre Ferraz
- Bret Curtis
- Carl D'Halluin (Amplidata)
- Christian Hergert
- Dave Peticolas
- Gleicon Moraes
- Ilya Skriblovsky
- Jonathan Stoppani
- Mark L
- Mike Dirolf (mdirolf)
- Renzo Sanchez-Silva (rnz0)
- Runar Petursson
- Silas Sewell
- Stiletto
- Toby Padilla
- Tryggvi Björgvinsson
- Vanderson Mota (chunda)
- flanked
- renzo
- shyilent



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



**t**

txmongo, 10  
txmongo.\_gridfs, 13  
txmongo.\_gridfs.errors, 10  
txmongo.\_gridfs.grid\_file, 11  
txmongo.collection, 5  
txmongo.connection, 6  
txmongo.database, 7  
txmongo.filter, 7  
txmongo.gridfs, 8  
txmongo.protocol, 8



**A**

aggregate() (txmongo.collection.Collection method), 5  
 aliases (txmongo.\_gridfs.grid\_file.GridOut attribute), 12  
 ASCENDING() (in module txmongo.filter), 7  
 authenticate() (txmongo.connection.ConnectionPool method), 6  
 authenticate() (txmongo.database.Database method), 7  
 authenticate() (txmongo.protocol.MongoProtocol method), 9  
 authenticate\_mongo\_cr() (txmongo.protocol.MongoProtocol method), 9  
 authenticate\_scram\_sha1() (txmongo.protocol.MongoProtocol method), 9

**C**

chunk\_size (txmongo.\_gridfs.grid\_file.GridIn attribute), 11  
 chunk\_size (txmongo.\_gridfs.grid\_file.GridOut attribute), 12  
 close() (txmongo.\_gridfs.grid\_file.GridIn method), 11  
 close() (txmongo.\_gridfs.grid\_file.GridOut method), 12  
 closed (txmongo.\_gridfs.grid\_file.GridIn attribute), 11  
 Collection (class in txmongo.collection), 5  
 collection\_names() (txmongo.database.Database method), 7  
 command() (txmongo.database.Database method), 7  
 comment (class in txmongo.filter), 7  
 connection (txmongo.database.Database attribute), 7  
 connectionLost() (txmongo.protocol.MongoProtocol method), 9  
 connectionMade() (txmongo.protocol.MongoProtocol method), 9  
 ConnectionPool (class in txmongo.connection), 6  
 connectionReady() (txmongo.protocol.MongoProtocol method), 9  
 content\_type (txmongo.\_gridfs.grid\_file.GridIn attribute), 11  
 content\_type (txmongo.\_gridfs.grid\_file.GridOut at-

tribute), 12  
 CorruptGridFile, 10  
 count() (txmongo.collection.Collection method), 5  
 create\_collection() (txmongo.database.Database method), 7  
 create\_index() (txmongo.collection.Collection method), 5

**D**

Database (class in txmongo.database), 7  
 database (txmongo.collection.Collection attribute), 5  
 dataBuffer (txmongo.protocol.MongoDecoder attribute), 8  
 dataReceived() (txmongo.protocol.MongoServerProtocol method), 9  
 decode() (txmongo.protocol.MongoDecoder static method), 8  
 Delete (class in txmongo.protocol), 8  
 delete() (txmongo.\_gridfs.GridFS method), 13  
 delete\_many() (txmongo.collection.Collection method), 5  
 delete\_one() (txmongo.collection.Collection method), 5  
 DESCENDING() (in module txmongo.filter), 7  
 disconnect() (txmongo.connection.ConnectionPool method), 6  
 distinct() (txmongo.collection.Collection method), 5  
 drop() (txmongo.collection.Collection method), 5  
 drop\_collection() (txmongo.database.Database method), 7  
 drop\_index() (txmongo.collection.Collection method), 5  
 drop\_indexes() (txmongo.collection.Collection method), 5

**E**

ensure\_index() (txmongo.collection.Collection method), 5  
 explain (class in txmongo.filter), 7

**F**

fail() (txmongo.protocol.MongoProtocol method), 9  
 feed() (txmongo.protocol.MongoDecoder method), 8  
 filemd5() (txmongo.collection.Collection method), 5

- filename (txmongo.\_gridfs.grid\_file.GridIn attribute), 11
- find() (txmongo.collection.Collection method), 5
- find\_and\_modify() (txmongo.collection.Collection method), 5
- find\_one() (txmongo.collection.Collection method), 5
- find\_one\_and\_delete() (txmongo.collection.Collection method), 5
- find\_one\_and\_replace() (txmongo.collection.Collection method), 5
- find\_one\_and\_update() (txmongo.collection.Collection method), 5
- find\_with\_cursor() (txmongo.collection.Collection method), 5
- full\_name (txmongo.collection.Collection attribute), 6
- ## G
- GEO2D() (in module txmongo.filter), 7
- GEO2DSPHERE() (in module txmongo.filter), 7
- GEOHAYSTACK() (in module txmongo.filter), 7
- get() (txmongo.\_gridfs.GridFS method), 13
- get\_default\_database() (txmongo.connection.ConnectionPool method), 6
- get\_last\_error() (txmongo.protocol.MongoProtocol method), 9
- get\_last\_version() (txmongo.\_gridfs.GridFS method), 13
- get\_request\_id() (txmongo.protocol.MongoClientProtocol method), 8
- Getmore (class in txmongo.protocol), 8
- getprotocol() (txmongo.connection.ConnectionPool method), 6
- getprotocols() (txmongo.connection.ConnectionPool method), 6
- GridFS (class in txmongo.\_gridfs), 13
- GridFSError, 10
- GridIn (class in txmongo.\_gridfs.grid\_file), 11
- GridOut (class in txmongo.\_gridfs.grid\_file), 11
- GridOutIterator (class in txmongo.\_gridfs.grid\_file), 13
- group() (txmongo.collection.Collection method), 6
- ## H
- handle() (txmongo.protocol.MongoServerProtocol method), 9
- handle\_DELETE() (txmongo.protocol.MongoServerProtocol method), 9
- handle\_GETMORE() (txmongo.protocol.MongoServerProtocol method), 9
- handle\_INSERT() (txmongo.protocol.MongoServerProtocol method), 9
- handle\_KILL\_CURSORS() (txmongo.protocol.MongoServerProtocol method), 9
- handle\_MSG() (txmongo.protocol.MongoServerProtocol method), 9
- handle\_QUERY() (txmongo.protocol.MongoServerProtocol method), 9
- handle\_REPLY() (txmongo.protocol.MongoProtocol method), 9
- handle\_REPLY() (txmongo.protocol.MongoServerProtocol method), 9
- handle\_UPDATE() (txmongo.protocol.MongoServerProtocol method), 9
- hint (class in txmongo.filter), 7
- ## I
- index\_information() (txmongo.collection.Collection method), 6
- inflight() (txmongo.protocol.MongoProtocol method), 9
- Insert (class in txmongo.protocol), 8
- insert() (txmongo.collection.Collection method), 6
- insert\_many() (txmongo.collection.Collection method), 6
- insert\_one() (txmongo.collection.Collection method), 6
- ## K
- KillCursors (class in txmongo.protocol), 8
- ## L
- lazyMongoConnection (in module txmongo.connection), 7
- lazyMongoConnectionPool (in module txmongo.connection), 7
- len (txmongo.protocol.Msg attribute), 9
- length (txmongo.\_gridfs.grid\_file.GridIn attribute), 11
- length (txmongo.\_gridfs.grid\_file.GridOut attribute), 12
- list() (txmongo.\_gridfs.GridFS method), 13
- ## M
- map\_reduce() (txmongo.collection.Collection method), 6
- max\_wire\_version (txmongo.protocol.MongoProtocol attribute), 9
- md5 (txmongo.\_gridfs.grid\_file.GridIn attribute), 11
- md5 (txmongo.\_gridfs.grid\_file.GridOut attribute), 12
- message (txmongo.protocol.Msg attribute), 9
- metadata (txmongo.\_gridfs.grid\_file.GridOut attribute), 12
- min\_wire\_version (txmongo.protocol.MongoProtocol attribute), 9
- MongoAuthenticationError, 8
- MongoClientProtocol (class in txmongo.protocol), 8
- MongoConnection (class in txmongo.connection), 6
- MongoConnectionPool (in module txmongo.connection), 6
- MongoDecoder (class in txmongo.protocol), 8
- MongoProtocol (class in txmongo.protocol), 9

MongoServerProtocol (class in txmongo.protocol), 9  
 Msg (class in txmongo.protocol), 9

## N

name (txmongo.\_gridfs.grid\_file.GridOut attribute), 12  
 name (txmongo.collection.Collection attribute), 6  
 name (txmongo.database.Database attribute), 7  
 new\_file() (txmongo.\_gridfs.GridFS method), 14  
 next() (txmongo.\_gridfs.grid\_file.GridOutIterator method), 13  
 next() (txmongo.protocol.MongoDecoder method), 9  
 NoFile, 10

## O

opcode (txmongo.protocol.Msg attribute), 9  
 options() (txmongo.collection.Collection method), 6

## P

put() (txmongo.\_gridfs.GridFS method), 14

## Q

Query (class in txmongo.protocol), 10

## R

read() (txmongo.\_gridfs.grid\_file.GridOut method), 12  
 remove() (txmongo.collection.Collection method), 6  
 rename() (txmongo.collection.Collection method), 6  
 replace\_one() (txmongo.collection.Collection method), 6  
 Reply (class in txmongo.protocol), 10  
 request\_id (txmongo.protocol.Msg attribute), 10  
 response\_to (txmongo.protocol.Msg attribute), 10

## S

save() (txmongo.collection.Collection method), 6  
 seek() (txmongo.\_gridfs.grid\_file.GridOut method), 12  
 send() (txmongo.protocol.MongoClientProtocol method), 8  
 send\_DELETE() (txmongo.protocol.MongoClientProtocol method), 8  
 send\_GETMORE() (txmongo.protocol.MongoClientProtocol method), 8  
 send\_GETMORE() (txmongo.protocol.MongoProtocol method), 9  
 send\_INSERT() (txmongo.protocol.MongoClientProtocol method), 8  
 send\_KILL\_CURSORS() (txmongo.protocol.MongoClientProtocol method), 8  
 send\_MSG() (txmongo.protocol.MongoClientProtocol method), 8  
 send\_QUERY() (txmongo.protocol.MongoClientProtocol method), 8

send\_QUERY() (txmongo.protocol.MongoProtocol method), 9  
 send\_REPLY() (txmongo.protocol.MongoClientProtocol method), 8  
 send\_UPDATE() (txmongo.protocol.MongoClientProtocol method), 8  
 set\_wire\_versions() (txmongo.protocol.MongoProtocol method), 9  
 snapshot (class in txmongo.filter), 7  
 sort (class in txmongo.filter), 8

## T

tell() (txmongo.\_gridfs.grid\_file.GridOut method), 12  
 txmongo (module), 10  
 txmongo.\_gridfs (module), 13  
 txmongo.\_gridfs.errors (module), 10  
 txmongo.\_gridfs.grid\_file (module), 11  
 txmongo.collection (module), 5  
 txmongo.connection (module), 6  
 txmongo.database (module), 7  
 txmongo.filter (module), 7  
 txmongo.gridfs (module), 8  
 txmongo.protocol (module), 8

## U

UnsupportedAPI, 10  
 Update (class in txmongo.protocol), 10  
 update() (txmongo.collection.Collection method), 6  
 update\_many() (txmongo.collection.Collection method), 6  
 update\_one() (txmongo.collection.Collection method), 6  
 upload\_date (txmongo.\_gridfs.grid\_file.GridIn attribute), 11  
 upload\_date (txmongo.\_gridfs.grid\_file.GridOut attribute), 12  
 uri (txmongo.connection.ConnectionPool attribute), 6

## W

with\_options() (txmongo.collection.Collection method), 6  
 write() (txmongo.\_gridfs.grid\_file.GridIn method), 11  
 write\_concern (txmongo.collection.Collection attribute), 6  
 write\_concern (txmongo.connection.ConnectionPool attribute), 6  
 write\_concern (txmongo.database.Database attribute), 7  
 writelines() (txmongo.\_gridfs.grid\_file.GridIn method), 11