

---

# **python-ezdb**

***Release 0.0.1.r0.459edd2***

**GeorgeRaven**

**Jul 15, 2021**



## TABLE OF CONTENTS

<b>1</b>	<b>The Open Software License 3.0 (OSL-3.0)</b>	<b>3</b>
<b>2</b>	<b>Serving with MongoDB</b>	<b>7</b>
2.1	Further Reading . . . . .	7
<b>3</b>	<b>Mongo</b>	<b>9</b>
3.1	Example usage . . . . .	9
3.2	API . . . . .	10
<b>4</b>	<b>Disambiguation</b>	<b>15</b>
4.1	General . . . . .	15
4.2	Authentication . . . . .	15
4.3	TLS Related . . . . .	15
4.4	Shard/ Replica Related . . . . .	15
<b>5</b>	<b>troubleshooting</b>	<b>17</b>
5.1	MongoDB/ Serving Issues . . . . .	17
	<b>Index</b>	<b>19</b>



Copyright (c) 2019 George Onoufriou (GeorgeRaven, archer, DreamingRaven)



## **THE OPEN SOFTWARE LICENSE 3.0 (OSL-3.0)**

This Open Software License (the “License”) applies to any original work of authorship (the “Original Work”) whose owner (the “Licensor”) has placed the following licensing notice adjacent to the copyright notice for the Original Work:

Licensed under the Open Software License version 3.0

- 1) Grant of Copyright License. Licensor grants You a worldwide, royalty-free, non-exclusive, sublicensable license, for the duration of the copyright, to do the following:
  - a) to reproduce the Original Work in copies, either alone or as part of a collective work;
  - b) to translate, adapt, alter, transform, modify, or arrange the Original Work, thereby creating derivative works (“Derivative Works”) based upon the Original Work;
  - c) to distribute or communicate copies of the Original Work and Derivative Works to the public, with the proviso that copies of Original Work or Derivative Works that You distribute or communicate shall be licensed under this Open Software License;
  - d) to perform the Original Work publicly; and
  - e) to display the Original Work publicly.
- 2) Grant of Patent License. Licensor grants You a worldwide, royalty-free, non-exclusive, sublicensable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, for the duration of the patents, to make, use, sell, offer for sale, have made, and import the Original Work and Derivative Works.
- 3) Grant of Source Code License. The term “Source Code” means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work.
- 4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior permission of the Licensor. Except as expressly stated herein, nothing in this License grants any license to Licensor’s trademarks, copyrights, patents, trade secrets or any other intellectual property. No patent license is granted to make, use, sell, offer for sale, have made, or import embodiments of any patent claims other than the licensed claims defined in Section 2. No license is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under terms different from this License any Original Work that Licensor otherwise would have a right to license.
- 5) External Deployment. The term “External Deployment” means the use, distribution, or communication of the Original Work or Derivative Works in any way such that the Original Work or Derivative Works may be used by anyone other than You, whether those works are distributed or communicated to those persons or made available

as an application intended for use over a network. As an express condition for the grants of license hereunder, You must treat any External Deployment by You of the Original Work or a Derivative Work as a distribution under section 1(c).

- 6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent, or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an “Attribution Notice.” You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.
- 7) Warranty of Provenance and Disclaimer of Warranty. Licensor warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensor are owned by the Licensor or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately preceding sentence, the Original Work is provided under this License on an “AS IS” BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of non-infringement, merchantability or fitness for a particular purpose. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to the Original Work is granted by this License except under this disclaimer.
- 8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to anyone for any indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to the extent applicable law prohibits such limitation.
- 9) Acceptance and Termination. If, at any time, You expressly assented to this License, that assent indicates your clear and irrevocable acceptance of this License and all of its terms and conditions. If You distribute or communicate copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. This License conditions your rights to undertake the activities listed in Section 1, including your right to create Derivative Works based upon the Original Work, and doing so without honoring these terms and conditions is prohibited by copyright law and international treaty. Nothing in this License is intended to affect copyright exceptions and limitations (including “fair use” or “fair dealing”). This License shall terminate immediately and You may no longer exercise any of the rights granted to You by this License upon your failure to honor the conditions in Section 1(c).
- 10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.
- 11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of copyright or patent law in the appropriate jurisdiction. This section shall survive the termination of this License.
- 12) Attorneys’ Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys’ fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.
- 13) Miscellaneous. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.



- 14) Definition of “You” in This License. “You” throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, “control” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
- 15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You.
- 16) Modification of This License. This License is Copyright © 2005 Lawrence Rosen. Permission is granted to copy, distribute, or communicate this License without modification. Nothing in this License permits You to modify this License as applied to the Original Work or to Derivative Works. However, You may modify the text of this License and copy, distribute or communicate your modified version (the “Modified License”) and apply it to other original works of authorship subject to the following conditions: (i) You may not indicate in any way that your Modified License is the “Open Software License” or “OSL” and you may not use those names in the name of your Modified License; (ii) You must replace the notice specified in the first paragraph above with the notice “Licensed under <insert your license name here>” or with a notice of your own that is not confusingly similar to the notice in this License; and (iii) You may not claim that your original works are open source software unless your Modified License has been approved by Open Source Initiative (OSI) and You comply with its license review and certification process.



## **SERVING WITH MONGODB**

MongoDB is an object based database system. Python-ezdb provides a nice higher level interface “*Mongo*” by using PyMongo and os commands to make managing MongoDB more streamlined and less reliant on direct connection management to MongoDB.

### **2.1 Further Reading**



## MONGO

Nemesyst MongoDB abstraction/ Handler. This handler helps abstract some pymongo functionality to make it easier for us to use a MongoDB database for our deep learning purposes.

### 3.1 Example usage

This unit test also briefly shows how to use grids by dumping tuple items in the form (dict(), object), where the dict will become the files metadata and the object is some form of the data that can be sequentialized into the database.

**Warning:** Mongo uses subprocess.Popen in init, start, and stop, since these threads would otherwise lock up nemesyst, with time.sleep() to wait for the database to startup, and shutdown. Depending on the size of your database it may be necessary to extend the length of time time.sleep() as larger databases will take longer to startup and shutdown.

Setting up a basic database, and initializing it with a user.

```
def setUp(self):
    """Predefined setUp function for preparing tests, in our case
    creating the database."""
    import os
    from ezdb.mongo import Mongo

    self.db_path = "./unit_test_db"
    self.db = Mongo({"pylog": null_printer, "db_path": self.db_path,
                    "db_log_path": self.db_path})

    self.db.init()
    self.db.start()
    self.assertTrue(os.path.isdir(self.db_path))
```

Connecting to and dumping data to a database using normal mongodb requests.

```
def test_dump(self):
    """Test/ example of dump and retrieve from a MongoDB database."""
    from ezdb.mongo import Mongo

    db = Mongo({"pylog": null_printer})
    self.assertIsInstance(db, Mongo)
    db.connect()
    db.dump(db_collection_name="test", data={"success": 1})
```

(continues on next page)

(continued from previous page)

```

cursor = db.getCursor(db_collection_name="test")
for batch in db.getBatches(db_data_cursor=cursor):
    self.assertEqual(len(batch), 1)
    for doc in batch:
        self.assertEqual(doc["success"], 1)

```

Completely removing the database, this completely removes all your data.

```

def tearDown(self):
    """Predefined tearDown function for cleaning up after tests,
    in our case deleting any generated db files."""
    import os
    import shutil
    from ezdb.mongo import Mongo

    self.db.stop()
    if(self.db_path is not None):
        shutil.rmtree(self.db_path)
    self.assertFalse(os.path.isdir(self.db_path))

```

## 3.2 API

**class** ezdb.mongo.**Mongo**(args: Optional[dict] = None, logger: Optional[print] = None)  
 Python2/3 compatible MongoDB utility wrapper.

This wrapper saves its state in an internal overridable dictionary such that you can adapt it to your requirements, if you should need to do something unique, the caveat being it becomes harder to read.

### Parameters

- **args** (dictionary) – Dictionary of overrides.
- **logger** (function address) – Function address to print/ log to (default: print).

**Example** Mongo({"db\_user\_name": "someUsername", "db\_password": "somePassword"})

**Example** Mongo()

**connect**(db\_ip: Optional[str] = None, db\_port: Optional[str] = None, db\_authentication: Optional[str] = None, db\_authentication\_database=None, db\_user\_name: Optional[str] = None, db\_password: Optional[str] = None, db\_name: Optional[str] = None, db\_replica\_set\_name: Optional[str] = None, db\_replica\_read\_preference: Optional[str] = None, db\_replica\_max\_staleness: Optional[str] = None, db\_tls: Optional[bool] = None, db\_tls\_ca\_file: Optional[str] = None, db\_tls\_certificate\_key\_file: Optional[str] = None, db\_tls\_certificate\_key\_file\_password: Optional[str] = None, db\_tls\_crl\_file: Optional[str] = None, db\_collection\_name: Optional[str] = None) → pymongo.database.Database

Connect to a specific mongod database.

This sets the internal db client which is necessary to connect to and use the associated database. Without it operations such as dump into the database will fail. This is replica set capable.

### Parameters

- **db\_ip** (string) – Database hostname or ip to connect to.
- **db\_port** (string) – Database port to connect to.

- **db\_authentication** (*string*) – The authentication method to use on db.
- **db\_user\_name** (*string*) – Username to use for authentication to db\_name.
- **db\_password** (*string*) – Password for db\_user\_name in database db\_name.
- **db\_name** (*string*) – The name of the database to connect to.
- **db\_replica\_set\_name** (*string*) – Name of the replica set to connect to.
- **db\_replica\_read\_preference** (*string*) – What rep type to prefer reads from.
- **db\_replica\_max\_staleness** (*string*) – Max seconds behind is replica allowed.
- **db\_tls** (*bool*) – use TLS for db connection.
- **db\_tls\_ca\_file** (*string*) – Certificate authority file path.
- **db\_tls\_certificate\_key\_file** (*string*) – Certificate and key file for tls.
- **db\_tls\_certificate\_key\_file\_password** (*string*) – Cert and key file pass.
- **db\_tls\_crl\_file** (*string*) – Certificate revocation list file path.
- **db\_collection\_name** (*string*) – GridFS collection to use.

**Returns** database client object

**Return type** pymongo.database.Database

**debug**() → None

Log function to help track the internal state of the class.

Simply logs working state of args dict.

**dump**(*db\_collection\_name: str, data: dict, db: Optional[pymongo.database.Database] = None*) → None

Import data dictionary into database.

#### Parameters

- **db\_collection\_name** (*string*) – Collection name to import into.
- **data** (*dictionary*) – Data to import into database.
- **db** (*pymongo.database.Database*) – Database to import data into.

**Example** dump(db\_collection\_name="test", data={"subdict":{"hello": "world"}})

**getBatches**(*db\_batch\_size: Optional[int] = None, db\_data\_cursor:*

*Optional[pymongo.command\_cursor.CommandCursor] = None*) → list

Get database cursor data in batches.

#### Parameters

- **db\_batch\_size** (*integer*) – The number of items to return in a single round.
- **db\_data\_cursor** (*command\_cursor.CommandCursor*) – The cursor to use to retrieve data from db.

**Returns** yields a list of items requested.

**Return type** list of dicts

**Todo** desperately needs a rewrite and correction of bug. Last value always fails. I want this in a magic function too to make it easy.

**getCursor**(*db: Optional[pymongo.database.Database] = None, db\_pipeline: Optional[list] = None, db\_collection\_name: Optional[str] = None*) → pymongo.command\_cursor.CommandCursor

Use aggregate pipeline to get a data-cursor from the database.

This cursor is what mongodb provides to allow you to request the data from the database in a manner you control, instead of just getting a big dump from the database.

#### Parameters

- **db\_pipeline** (*list of dicts*) – Mongodb aggregate pipeline data to transform and retrieve the data as you request.
- **db\_collection\_name** (*str*) – The collection name which we will pull data from using the aggregate pipeline.
- **db** (*pymongo.database.Database*) – Database object to operate pipeline on.

**Returns** Command cursor to fetch the data with.

**Return type** pymongo.command\_cursor.CommandCursor

**getFiles**(*db\_batch\_size: Optional[int] = None, db\_data\_cursor: Optional[pymongo.command\_cursor.CommandCursor] = None, db\_collection\_name: Optional[str] = None, db: Optional[pymongo.database.Database] = None*) → list

Get gridfs files from mongodb by id using cursor to .files.

#### Parameters

- **db\_batch\_size** (*integer*) – The number of items to return in a single round.
- **db\_data\_cursor** (*command\_cursor.CommandCursor*) – The cursor to use to retrieve data from db.
- **db\_collection\_name** (*str*) – The top level collecton name not including .chunks or .files where gridfs is to operate.
- **db** (*pymongo.database.Database*) – Database object to operate pipeline on.

**Returns** yields a list of tuples containing (item requested, metadata).

**init**(*db\_path: Optional[str] = None, db\_log\_path: Optional[str] = None, db\_log\_name: Optional[str] = None, db\_config\_path: Optional[str] = None*) → None

Initialise the database.

Includes ensuring db path and db log path exist and generating, creating the DB files, and adding an authentication user. All of this should be done on a localhost port so that the unprotected database is never exposed.

#### Parameters

- **db\_path** (*string*) – Desired directory of MongoDB database files.
- **db\_log\_path** (*string*) – Desired directory of MongoDB log files.
- **db\_log\_name** (*string*) – Desired name of log file.
- **db\_config\_path** (*string*) – Config file to pass to MongoDB.

**login**(*db\_port: Optional[str] = None, db\_user\_name: Optional[str] = None, db\_password: Optional[str] = None, db\_name: Optional[str] = None, db\_ip: Optional[str] = None*) → None

Log in to database, interrupt, and available via cli.

#### Parameters

- **db\_port** (*string*) – Database port to connect to.



- **db\_user\_name** (*string*) – Database user to authenticate as.
- **db\_password** (*string*) – User password to authenticate with.
- **db\_name** (*string*) – Database to authenticate to, the authentication db.
- **db\_ip** (*string*) – Database ip to connect to.

**start**(*db\_ip*: *Optional[str] = None*, *db\_port*: *Optional[str] = None*, *db\_path*: *Optional[str] = None*, *db\_log\_path*: *Optional[str] = None*, *db\_log\_name*: *Optional[str] = None*, *db\_cursor\_timeout*: *Optional[int] = None*, *db\_config\_path*: *Optional[str] = None*, *db\_replica\_set\_name*: *Optional[str] = None*) → subprocess.Popen

Launch an on machine database with authentication.

**Parameters**

- **db\_ip** (*list*) – List of IPs to accept connections from.
- **db\_port** (*string*) – Port desired for database.
- **db\_path** (*string*) – Path to parent dir of database.
- **db\_log\_path** (*string*) – Path to parent dir of log files.
- **db\_log\_name** (*string*) – Desired base name for log files.
- **db\_cursor\_timeout** (*integer*) – Set timeout time for unused cursors.
- **db\_path** – Config file path to pass to MongoDB.

**Return type** subprocess.Popen

**Returns** Subprocess of running MongoDB.

**stop**(*db\_path=None*) → subprocess.Popen

Stop a running local database.

**Parameters** **db\_path** (*string*) – The path to the database to shut down.

**Returns** Subprocess of database closer.

**Return type** subprocess.Popen



## DISAMBIGUATION

### 4.1 General

#### 4.1.1 IP (address)

An Internet Protocol (**IP**) address can be thought of as a computer postcode to help identify where a machine can be found, for the purposes of communicating to it; it is difficult to mail a letter without a postcode.

**local *IP (address)* example:** 192.168.1.1

#### 4.1.2 port

A computers **port** is usually associated with an *IP (address)*. This **port** further specifies where specifically to communicate to. Following the post analogy the post code is the *IP (address)*, but the house number is the **port**. This analogy breaks down when we consider that each computer can and usually does have multiple **ports** in the thousands, so maybe the computer is a property tycoon with multiple homes that can be contacted.

***port* example:** 192.168.1.1 :22

### 4.2 Authentication

### 4.3 TLS Related

### 4.4 Shard/ Replica Related



## TROUBLESHOOTING

## 5.1 MongoDB/ Serving Issues

**Error: not master and slaveOk=false** This error means you have attempted to read from a replica set that is not the master. If you would like to read from SECONDARY-ies/ slaves (anything thats not the PRIMARY) you can:

**Mongo shell:**

```
rs.slaveOk()
```

**pymongo.errors.OperationFailure: Authentication failed** This error means likely means that your authentication credentials are incorrect, you will want to check the values you are passing to pymongo via Nemesyst to ensure they are what you are expecting. In particular pay special attention to `Mongo().connect()` as it is the life blood of all connections but since the driver is a lazy driver it wont fail until you attempt to use the connection.

**pymongo.errors.ServerSelectionTimeoutError: 192.168.1.10:27017: [SSL: CERTIFICATE\_VERIFY\_FAILED] certificate**  
This error is a implementation quirk of pymongo not converting between ip addresses and hostname strings implicitly even if the certificate stipulates the desired IP address correctly for other things such as the mongo client. My only recommendation is to either use hostnames even if that only be explicit in `/etc/hosts` or disabling TLS but both are bad options for anything more than testing.



## INDEX

### C

`connect()` (*ezdb.mongo.Mongo method*), 10

### D

`debug()` (*ezdb.mongo.Mongo method*), 11

`dump()` (*ezdb.mongo.Mongo method*), 11

### G

`getBatches()` (*ezdb.mongo.Mongo method*), 11

`getCursor()` (*ezdb.mongo.Mongo method*), 11

`getFiles()` (*ezdb.mongo.Mongo method*), 12

### I

`init()` (*ezdb.mongo.Mongo method*), 12

### L

`login()` (*ezdb.mongo.Mongo method*), 12

### M

`Mongo` (*class in ezdb.mongo*), 10

### S

`start()` (*ezdb.mongo.Mongo method*), 13

`stop()` (*ezdb.mongo.Mongo method*), 13