Setting up MongoDB

How to prepare MongoDB for an Exponential site - the requirements, a user created in the site's own database with readWrite and dbAdmin, why a user is required even without access control, the settings block, the...

Source file: `doc/eng-us/help/database-administrators/mongodb-setup.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/database-administrators/mongodb-setup.md

This page is for database administrators who prepare MongoDB for an Exponential site. MongoDB is supported through an adapter, `expMongoDB` (`lib/ezdb/classes/expmongodb.php`), that translates the kernel's SQL-shaped calls into MongoDB operations; it arrived with 6.0.14. This page answers: what the adapter needs, how to create the user, what the settings look like, how to install and which indexes to create. Read MongoDB: limits and operations before you choose MongoDB for production. Source: 9.5 MongoDB.

Contents

1. Requirements
2. Create the user
3. The settings
4. Install
5. Create the indexes
6. Build the search index
7. Checklist

Requirements

	Minimum	Tested
PHP	8.2	8.5
MongoDB server	6.0 (the wizard checks 4.0)	8.3
PHP extension `mongodb`	1.18	2.3.3
Composer library `mongodb/mongodb`	installed by you (`composer.json` only suggests it)	`^1.21 || ^2.0` in CI

Without the library the error is `Class "MongoDB\Client" not found` (PHP extensions per engine).

Create the user

The adapter connects with the URI `mongodb://<user>:<password>@<server>:<port>/<database>`. MongoDB uses the database in the URI's path as the authentication database when no `authSource` is given, so create the user **in the site's database**:

Shell

```
mongosh --eval '
  db.getSiblingDB("exponential").createUser({
    user: "exponential",
    pwd: passwordPrompt(),
    roles: [ { role: "readWrite", db: "exponential" }, { role: "dbAdmin", db: "exponential" } ]
  })'
```

`passwordPrompt()` asks for the password instead of putting it in the shell history. `readWrite` covers the site's work; `dbAdmin` lets the schema handler (`expMongoSchema`) create collections and indexes. The database itself is created by the first write; there is no `CREATE DATABASE`.

The user is required, also on a test server without access control: the adapter always writes `<user>:<password>@` into the URI, and with an empty `User` the driver refuses it with "Failed to parse MongoDB URI: ... 'default' authentication mechanism requires a username".

The settings

INI

```
[DatabaseSettings]
DatabaseImplementation=mongodb
Server=localhost
# empty: 27017
Port=27017
User=exponential
Password=<the password>
Database=exponential
```

User name and password are URL-encoded by the adapter, so special characters are safe. One client per connection URI is kept for the life of the process, so a Velocity worker reuses its connections.

Install

On MongoDB the setup wizard's configuration step can take one to three minutes. Raise PHP-FPM's `request_terminate_timeout` before you start, or install from the console:

Shell

```
EXP_INSTALL_DB_PASSWORD='<the password>' php bin/php/console exp:install --apply --db=mongodb \
    --db-host=127.0.0.1 --db-name=exponential --db-user=exponential --allow-root-user
```

`--db=mongo` is accepted too; the kickstarter takes `[database_choice] Type=mongodb`.

Create the indexes

The kernel's queries rely on indexes that the install does not create. Without them subtree queries and URL alias lookups scan whole collections:

Shell

```
mongosh "mongodb://exponential@localhost:27017/exponential" --file bin/mongodb/create_indexes.js
```

`mongosh` asks for the password. The script holds 45 `createIndex` calls on the content, tree, URL alias and other collections, several of them unique (for example `ezcontentobject.id`,

`ezcontentobject.remote_id`, `ezcontentobject_tree.node_id`). Run it again after an upgrade that adds collections; existing indexes are left as they are by MongoDB.

Build the search index

Run the cronjobs once to build the built-in search index:

Shell

```
php runcronjobs.php --allow-root-user
```

Then schedule them like on any other engine (Housekeeping: cronjobs and queue tables).

Checklist

- ☐ PHP 8.2+, extension `mongodb` 1.18+, library `mongodb/mongodb` installed in every PHP the site uses.
- ☐ MongoDB 6.0 or later with access control enabled.
- ☐ The user exists in the site's database with `readWrite` and `dbAdmin` only.
- ☐ `bin/mongodb/create_indexes.js` was run.
- ☐ The cronjobs ran once and are scheduled.
- ☐ Backups with `mongodump` are scheduled (Backups per engine).

Related pages

- MongoDB: limits and operations
- MongoDB as the database
- MongoDB for hosts
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

Become an Exponential Partner

Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.

Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.