

Monger, a Clojure client for MongoDB

- [Home](#)
- [All guides](#)
- [API reference](#)
- [Community](#)
- [Code](#)
- [Change log](#)
- [More Clojure libraries](#)
- [Donate](#)
- [Clojure Docs](#)
- [Guide list](#)
 - [Getting started](#)
 - [Connecting to MongoDB](#)
 - [Inserting Documents](#)
 - [Querying: finders and query DSL](#)
 - [Updating documents](#)
 - [Deleting documents](#)
 - [Indexing and other collection operations](#)
 - [Integration with other Clojure libraries](#)
 - [Using MongoDB Map/Reduce](#)
 - [Using GridFS](#)
 - [Using MongoDB Aggregation Framework \(MongoDB 2.2+ only\)](#)
 - [Using MongoDB commands](#)
 - [Miscellaneous](#)
- [Tell Us What You Think!](#)

Guide list

[Monger documentation](#) is organized as a number of guides, covering all kinds of topics.

We recommend that you read these guides, if possible, in this order:

Getting started

This guide combines an overview of Monger with a quick tutorial that helps you to get started with it. It should take about 10 minutes to read and study the provided code examples. This guide covers:

- Feature of Monger, why Monger was created
- Clojure and MongoDB version requirements
- How to add Monger dependency to your project
- Basic operations (create, read, update, delete)
- Overview of Monger Query DSL
- Overview of how Monger integrates with libraries like `clojure.data.json` and `JodaTime`.

Connecting to MongoDB

This guide covers:

- Connecting to MongoDB using connection options
- Connecting to MongoDB using connection URI
- Tuning database connection (concurrency level, automatic reconnection, timeouts, etc)
- Connecting in PaaS environments, for example, MongoHQ add-on on Heroku
- Choosing default database
- Working with multiple databases

Inserting Documents

This guide covers:

- Inserting documents
- Inserting batches of documents
- Checking database responses
- Validating data with Validateur, a [Clojure validation library](#)
- Setting default write concern
- Changing write concern for individual operations
- Working with multiple databases

Querying: finders and query DSL

This guide covers:

- Querying documents with Clojure
- Using query operators with Clojure
- Working with database cursors
- Using Monger Query DSL
- Working with multiple databases

Updating documents

This guide covers:

- Updating documents with Clojure
- Using atomic operations with Clojure
- Upserting documents
- Updating a single document vs multiple documents
- Overriding default write concern for individual operations
- Working with multiple databases

Deleting documents

This guide covers:

- Deleting documents with Clojure
- Deleting a single document vs multiple documents
- Working with multiple databases

Indexing and other collection operations

This guide covers:

- Creating indexes on MongoDB collections
- Dropping indexes
- Creating a capped collection
- Using MongoDB [TTL collections](#) (MongoDB 2.2+)
- Reindexing a collection
- Dropping a collection

Integration with other Clojure libraries

This guide covers:

- Monger's philosophy of "having batteries included"
- Integration with `clojure.data.json`
- Integration with `clj-time` and Joda Time
- Integration with `clojure.core.cache`: MongoDB-based Clojure cache protocol implementation
- Using MongoDB-backed Ring session stores
- Basic [Noir](#) integration example

Using MongoDB Map/Reduce

This guide covers:

- Using Map/Reduce with Clojure
- Storing and loading JavaScript functions from classpath

Using GridFS

This guide covers:

- Storing files on GridFS from Clojure
- Loading files from GridFS with Clojure

Using MongoDB Aggregation Framework (MongoDB 2.2+ only)

This guide covers:

- Using MongoDB 2.2 Aggregation Framework with Clojure

Using MongoDB commands

This guide covers:

- Performing MongoDB commands with Clojure

Miscellaneous

This guide covers miscellaneous topics:

- [Logging configuration](#)

Tell Us What You Think!

Please take a moment to tell us what you think about this guide on Twitter or the [Monger mailing list](#)

Let us know what was unclear or what has not been covered. Maybe you do not like the guide style or grammar or discover spelling mistakes. Reader feedback is key to making the documentation better.

[comments powered by Disqus](#)

This website was developed by [ClojureWerkz team](#).

Follow us on Twitter: [ClojureWerkz](#), [Michael Klishin](#), [Alex P](#)

Artwork by [zuk13](#)