

Unit Testing MongoDB*

* Actually: testing applications *that use* MongoDB.

What do we unit test?

- WriteConcern set correctly
- Unique indexes
- Documents persisted correctly
- Queries return the right objects
- Versioning strategy implemented correctly

Mocking?

- Don't use a mock Driver implementation
- Use the real Driver against a test database

Setup

- Execute a full DDL script before tests
- Tables, indices, foreign keys
- Specific insertion order
- Data deletion needs specific order

Setup

- ~~Execute a full DDL script before tests~~
- ~~Tables, indices, foreign keys~~
- ~~Specific insertion order~~
- ~~Data deletion needs specific order~~

Actual setup

- Take advantage of lazy creation
- Make index setup part of your app
 - Execute during test setup phase
- Delete collections (or contents) after
- Feasible to do this once per test

Test isolation

- Create a whole DB for each test
 - simple, but slow
 - MongoDB writes 100+MB to disk for each new database
 - We can do better

Test isolation

- New collection for each test
 - Not IO intensive to create a collection
- Inject collection into your class
- Delete collection after test

Random tips

- Shared MongoDB server for unit tests
 - Use `--smallfiles` for a unit test server
 - Increase file handles for connections
- Everyone gets their own server
 - Use the MongoDB plugin for Jenkins