Drivers

http://docs.mongodb.org/ecosystem/drivers/

Outline

- Role of the driver and implementation
- Some internals
- Shell tips & tricks!

Role of the driver

One reason why MongoDB was successfully adopted by developers was the investment in almost a dozen drivers up front.

This allowed developers to work with objects that were already familiar to them:

- Python -> dictionaries
- Ruby -> hashes
- Perl -> associative arrays

Drivers provide the ability to persist known objects in the DB without having to learn too many new things.

Drivers and MongoDB 3.0

- 2 releases around the same time in Q1-2015
  A. Needed to support the new SCRAM authentication scheme for MongoDB 3.0
     - Users must upgrade drivers to use 3.0 if *–auth* is enabled
  B. Drivers are all supporting the same specifications
     - https://github.com/mongodb/specifications/tree/master/source
     - Before, they implemented functionality differently

Common Specifications

- CRUD
  - https://github.com/mongodb/specifications/blob/master/source/crud/crud.rst
- Authentication
  - https://github.com/mongodb/specifications/blob/master/source/auth/auth.rst
- Server Discovery and Monitoring
  - https://github.com/mongodb/specifications/blob/master/source/server-discovery-and-monitoring/server-discovery-and-monitoring-summary.rst
- Server Selection
  - https://github.com/mongodb/specifications/blob/master/source/server-selection/server-selection.rst
- Index Management
  - https://github.com/mongodb/specifications/blob/master/source/index-management.rst

What does a driver need to do

- BSON support
- Wire Protocol
- Keeping track of the cluster status
- Implements the following commands:
  - Find
  - getMore
  - killCursors

BSON

- BSON is the language of the DB; it's what's sent over the wire, stored in the DB…
- bsonspec.org

API for CRUD+ operations

- Essentials
  - Basic operations: query, insert, update, remove, ensureIndex, findOne, limit, sort
  - Fetch more data from a cursor (dbGetMore)
  - Sending the KillCursors operation

- Nice to haves
  - Database $cmd support and helpers
  - getLastError(), count(), eval(), explain(), hint(), $where

Wire Protocol

- Clients communicate with the database server through a regular TCP/IP socket
- There are two types of messages, client requests and database responses, each having a slightly different structure.
- Commands from the driver:
  - OP_QUERY
  - OP_GET_MORE
  - OP_KILL_CURSORS
  - OP_INSERT
  - OP_UPDATE
  - OP_DELETE
  - OP_COMMAND
  - OP_COMMANDREPLY
- Wire Protocol Versions:
  - 2: write commands
  - 3: auth protocol
  - 4: read concern

- Spec: https://docs.mongodb.org/manual/reference/mongodb-wire-protocol/

Standard Message Header

```
struct MsgHeader {
    int32   messageLength; // total message size, including this
    int32   requestID;     // identifier for this message
    int32   responseTo;    // requestID from the original request
                           //   (used in reponses from db)
    int32   opCode;        // request type - see table below
}
```

Opcode Name	opCode value	Comment
OP_REPLY	1	Reply to a client request. responseTo is set
OP_MSG	1000	generic msg command followed by a string
OP_UPDATE	2001	update document
OP_INSERT	2002	insert new document
RESERVED	2003	formerly used for OP_GET_BY_OID
OP_QUERY	2004	query a collection
OP_GET_MORE	2005	Get more data from a query. See Cursors
OP_DELETE	2006	Delete documents
OP_KILL_CURSORS	2007	Tell database client is done with a cursor

getLastError

- App sends directives and then getLastError (GLE) to find out what happened
- Resulted in strange behaviors since you had to keep connections open to send GLE without sending other operations first
- Drivers abstracted this away, but still made two calls under the hood: one for the op and one for GLE
- Since 2.6, most write methods are returning a status at the end of command.
  - Op_Command => 2010
    Op_CommandReply => 2011

How about all the other commands?

```
> db.runCommand
function ( obj, extra ){
    if ( typeof( obj ) == "string" ){
        var n = {};
        n[obj] = 1;
        obj = n;
        if ( extra && typeof( extra ) == "object" ) {
            for ( var x in extra ) {
                n[x] = extra[x];
            }
        }
    }
    return this.getCollection( "$cmd" ).findOne( obj );
}
```

A command is just a findOne on some special $cmd collection

Look at example messages

- http://docs.mongodb.org/meta-driver/latest/legacy/mongodb-wire-protocol/#client-request-messages

- The types used in these documents (cstring, int32, etc.) are the same as those defined in the BSON specification.
- Integer constants are in capitals (e.g. ZERO for the integer value of 0).

Cursors

- Cursor id returned in query response, driver calls *getmore*
- Tailable cursors, e.g. *OpLog*
- Cursors timeout after 10 minutes by default
  - In Sharded Clusters, chunk migration cleaning is not completed until cursors are closed
- But cursors are dumb, so you must be smart!

Connection handling/pools

- Pool of connections that can be shared by many many threads
  - Replica Set vs Sharded Cluster
- Automatically connect to proper server, and failover, when connecting to a replica set
- Driver does *NOT* retry a failed operation
  - “Shit will happen” – Hannes Magnuson

MongoDB supported drivers

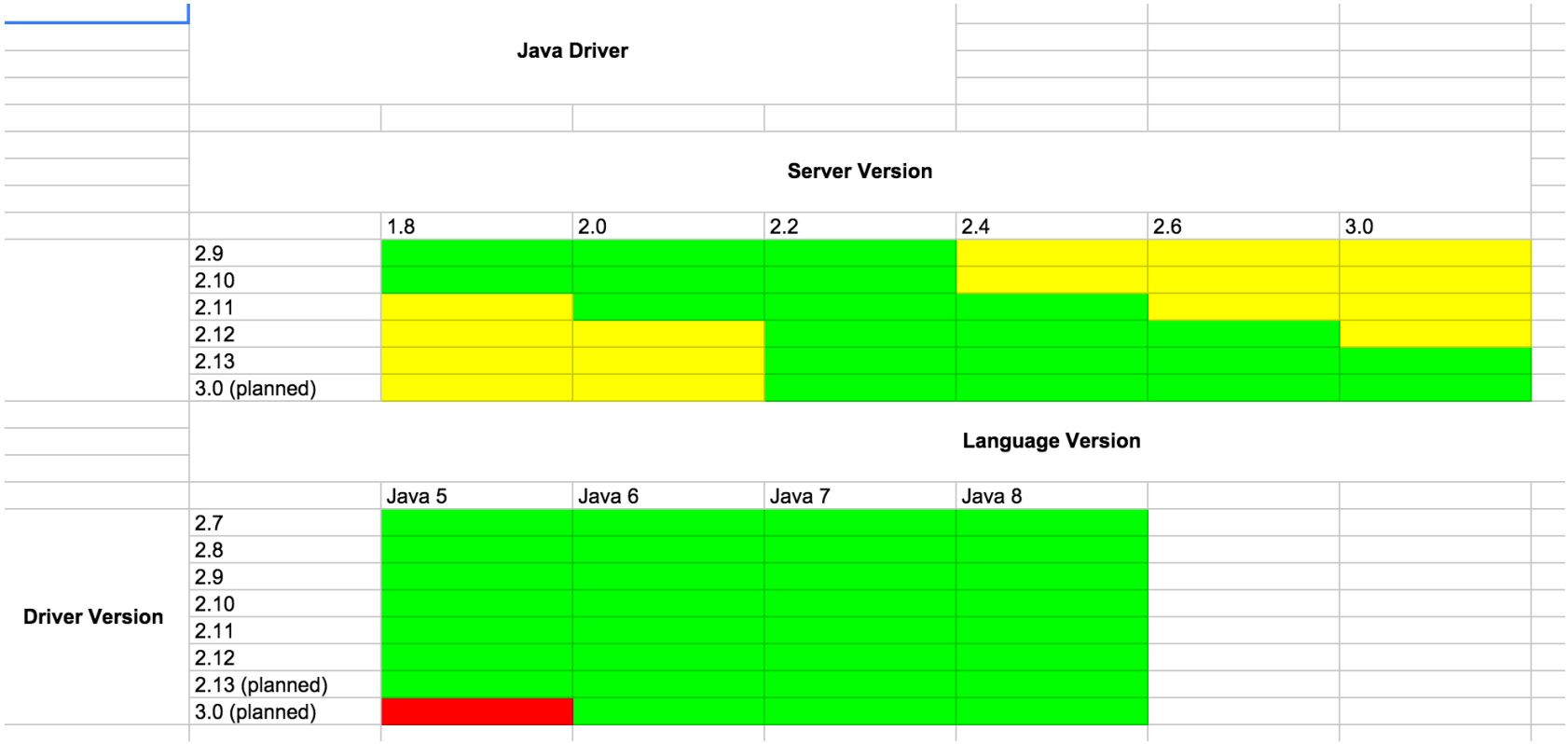
Documentation	Releases	Source	API	JIRA	Online Course
C	Releases	Source	API	JIRA	
C++11	Releases	Source	API	JIRA	
C++ (legacy)	Releases	Source	API	JIRA	
C#	Releases	Source	API	JIRA	Course
Java	Releases	Source	API	JIRA	Course
Node.js	Releases	Source	API	JIRA	Course
Perl	Releases	Source	API	JIRA	
PHP	Releases	Source	API	JIRA	
Python	Releases	Source	API	JIRA	Course
Motor	Releases	Source	API	JIRA	
Ruby	Releases	Source	API	JIRA	
Scala	Releases	Source	API	JIRA	

Object Mappers

- Names
  - ORM – Object Relational Mapping tool
  - ODM – Object Data-store Mapping tool
  - OM – Object Mapping tool
  - DRM – Document Resource Mapping tool
- Some examples per language
  - Java: Morphia*, Spring MongoDB
  - Node.js: Mongoose

Driver/Server compatibility

- https://docs.google.com/a/10gen.com/spreadsheet/ccc?key=0AkOLPg1iZn3qdFc0T25hd0Y3VGZyTkR2aE9JQmw2UXc&usp=drive_web#gid=1



Shell tips & tricks!

- The shell is a special case of the driver after all ;)

Internal - did we cover?

- Replica Set vs Sharded Cluster pool of connections
- BSON sorting