

CheatSheet: Mongodb

TOOLS

- PDF Link: [cheatsheet-mongodb-A4.pdf](#), Category: kubernetes
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-mongodb-A4>
- Related posts: Prometheus CheatSheet, Nagios CheatSheet, #denny-cheatsheets

File me Issues or star this repo.

1.1 Mongodb Operation

Name	Summary
Connect to mongodb	<code>mongo 192.168.75.108:12345/d_2003 -u u_2003 -p XXXX</code>
Start mongo	<code>mongod --config /etc/mongod.conf</code>
Stop mongo	<code>kill -15 <pid>; mongo --eval "db.getSiblingDB('admin').shutdownServer()"</code>
Mongo script get parameters	<code>mongo --eval "var day='2013-07-26' localhost:27017/shopex mongojs/update-index.js</code>
Mongo data folder	<code>/var/lib/mongodb/journal</code>
Export to json file	<code>mongoexport --db shopex --collection order --out collection.json</code>
Import json file	<code>mongoimport --upsert --db shopex --collection order --file collection.json</code>

1.2 Mongodb Basic

Name	Summary
Remove table	<code>db.kvstore.remove()</code>
define stored procedure	<code>db.system.js.save({_id:"addNumbers", value:function(x, y){ return x + y; }})</code>
view stored procedure	<code>db.system.js.find()</code>
call stored procedure	<code>db.eval('addNumbers(3, 4.2)')</code>
Create record	<code>db.users.insert({"name":{"first":"refactor","last":"refactor2"},"age":24})</code>
Update record	<code>db.users.update({"age":30}, {\$set:{"age":31}});</code>
Delete by filtering	<code>db.users.remove({"age":30})</code>
Loop cursor	<code>while(s.hasNext()) printjson(s.next());</code>
Loop array	<code>i=0; while(i < s.length) printjson(s[i++]);</code>

1.3 Mongodb Query

Name	Summary
Basic query	<code>db.users.find({"name":{"first":"refactor","last":"refactor2"}})</code>
Basic query	<code>db.users.find({"name.first":"refactor"})</code>
great than / or equal	<code>db.users.find({"age":{"\$gte":20}})</code>
not equal	<code>db.users.find({"age":{"\$ne":31}})</code>
exists	<code>db.users.find({"age":{"\$exists":true}})</code>
in	<code>db.users.find({"age":{"\$in":[21,24]}})</code>
not in	<code>db.users.find({"age":{"\$nin":[21,23]}})</code>
or	<code>db.users.find({"\$or":[{"name.last":"another"}, {"age":31}]})</code>
Count matched records	<code>db.order.find({"created":{"\$regex":"2013-07-12.*"}}).count()</code>
query by offset	<code>db.users.find({"age":{"\$nin":[21,23]}}).count()</code>
sort	<code>db.users.find({"age":{"\$nin":[21,23]}}).skip(1).limit(2)</code>
reverse order	<code>db.users.find().sort({"age":24});</code>
mod 20	<code>db.users.find().sort({"name.last": -1})</code>
Parse string to float	<code>parseFloat(v.availableConfirmFee, 10);</code>
filter by "like"	<code>db.taobao.find({"created":"/2013-07-12.*"/})</code>
	<code>db.order.find({"created":"/2013-07-14.*/", "num":{"\$exists":false}}).forEach(function(u) { printjson(u.t</code>
	<code>db.order.find({"created":{"\$regex":"day"}}).count();</code>
	<code>db.order.distinct("buyerNick", {"created":{"\$regex":"day"}}).length</code>

1.4 More Resources

License: Code is licensed under MIT License.