**MongoDB**

# Installation

**Datenstruktur anlegen**Damit MongoDB installiert werden kann, sind einige Verzeichnisse anzulegen. Folgende Ordnerstruktur stellt ein mögliches Beispiel dar:

C:\MongoAdmin  
C:\MongoAdmin\Data  
C:\MongoAdmin\Log  
C:\MongoAdmin\Config  
C:\MongoAdmin\Version

**Binaries downloaden**Die Binaries können bei der folgenden Website heruntergeladen werden:  
<http://www.mongodb.org> -> (Downloads -> All Version Binaries).

Hier im Beispiel wird die Version „win32/mongodb-win32-x86\_64-2008plus-ssl-v3.4-latest.zip“ heruntergeladen. Nach dem herunterladen der Zip-Datei kann diese entzippt und in das neue Verzeichnis „C:\MongoAdmin\Version\3.4\“ kopiert werden (Inhalt von bin).

**Systemvariable erstellen**Um den Start der MongoDB via Konsole zu erleichtern, ist es von Vorteil eine Systemvariable anzulegen. Folgende Systemvariable (Path) wird angelegt: „C:\Dev\MongoAdmin\Versions\3.4\bin“.

**Erster Test**  
Nachdem die oberen Schritte durchgeführt wurden, kann nun ein erster Test der Lauffähigkeit der MongoDB durchgeführt werden. Dazu ein Command-Prompt öffnen und folgenden Befehl eintippen:

*mongod --dbpath C:\MongoAdmin\Data*

In einem zweiten Command-Prompt kann nun die Verbindung zur MongoDB geprüft werden. Dazu folgenden Befehl eingeben:

*mongo*

**mongod.conf erstellen (YAML based)**Damit in einem späteren Schritt ein Windows-Service erstellt werden kann, ist eine „mongod.conf“ Datei notwendig. Diese im Verzeichnis „C:\MongoAdmin\Config „ erstellen und mit folgendem Inhalt befüllen:

storage:  
 dbPath: C:\Dev\MongoAdmin\Data

systemLog:  
 destination: file  
 path: C:\Dev\MongoAdmin\Log\mongod.log

**Windows-Service erstellen**  
Um einen Windows-Service zu erstellen, sollte am besten ein Windows-User angelegt (zB. mongodb) sowie darauf geachtet werden, dass dieser User Schreibberechtigungen auf das „C:\MongoAdmin“ Verzeichnis hat. Danach kann ein Command-Prompt (Admin-Modus) geöffnet werden. Mit dem Befehl

mongod --install --config „C:\MongoAdmin\Config\mongod.conf --serviceUser mongodb --servicePassword <password>

wird der Windows-Service angelegt.

**Test des Windows Service**  
sc qc mongodb

**Service starten**  
net start mongodb

**Service stoppen**  
net stop mongodb

**Service deinstallieren**  
mongod --remove

**MongoDB shutdown**  
use admin  
db.shutdownServer()

# Configuration

db.serverStatus()  
db.serverStatus().dur

**Storage Engines**  
systemLog:  
 logRotate: rename  
 logAppend: true  
 engine: mmapv1|wiredTiger

**Journal Flush Interval (Standard 100 ms)**- 2ms to 300ms

storage:  
 dbPath: “”  
 journal:  
 enabled: true  
 mmapv1:  
 journal:  
 commitIntervalMs: 144

**Directory per DB**  
storage:  
 directoryPerDB: true

**IP and Port**net:  
 bindIp: address[,address2…] (optional)  
 port: 27017

# Query Crash Course

**Save**db (Abfrage welche DB)  
use demo (switch to demo)  
db (Kontrolle)  
show databases (Anzeige der Datenbanken)  
show collections (Anzeige der Collections)  
  
Collections können Documents enthalten

**Insert (First document)**  
db.goo.insert({\_id:4})  
db.goo.find()  
show collections  
db.goo.stats()

**Save (document)**  
db.goo.save({\_id:4, x:1}) (Update)  
db.goo.find()

db.goo.save({\_id:4, x:1, y: true});  
db.goo.find()

db.goo.save({\_id:4, y:false}) (x wird herausgelöscht)

**Find**db.goo.find({\_id:4})  
db.goo.find({y:false})  
db.goo.find({\_id:4, y:false}  
db.goo.find({\_id: {$gt: 1, $lt: 7}}) ($gt->greater than, $lt-> lower than)

**Update**  
db.goo.update({\_id:4}, {$set:{x:1}})  
db.goo.update({\_id:4}, {$set:{y:true}})  
db.goo.update({y:true}, {$inc:{x:1}})  
db.goo.update({x:1}, {$inc:{x:1}}

**Delete**db.goo.remove({\_id:4})  
db.goo.find()  
db.goo.drop() -> drop collection  
show collections  
db.dropDatabases()  
show dbs

**BSON (Binary Serialization Object Notation)**use demo  
var p = {name: ‘dude’, born: ISODate(“1984-04-01”), likes: [‘naps’, ‘cake’], points: 1}  
db.peeps.save(p)  
db.peeps.find()

**Sort**db.s.find()  
db.s.find().sort({y:-1}) (descending)  
db.s.find().sort({y:1, x:1}) (ascending first y then x)  
db.s.find().sort({y:1, x:-1}) (ascending first y then x descending)

**Limit**db.s.find().sort({\_id:1}).limit(3)

**Cursor**