
MongoDB 使用手册-中文版

版本记录

日期	版本	原因	撰稿人	审核人
2010-7-15	1.0	初始化	万韬	

目录

第 1 章	MONGODB 简介.....	4
1.1	功能特点	4
1.2	适用范围	4
第 2 章	MONGODB 下载及安装.....	5
2.1	下载地址	5
2.2	安装方法	错误！未定义书签。
第 3 章	MONGODB 语法.....	6
3.1	基本命令	6
3.1.1.	启动 mongodb.....	6
3.1.2.	停止 mongodb.....	6
3.2	SQL 语法	7
3.2.1.	基本操作.....	7
3.2.2.	数据集操作.....	8
第 4 章	JAVA 操作 MONGODB	10
4.1	正在整理中.....	10
第 5 章	其它	10
5.1	正在整理中.....	10

第1章 MongoDB 简介

1.1 功能特点

官方网址: <http://www.mongodb.org/>

MongoDB 是一个基于**分布式**文件存储的数据库开源项目。由 C++语言编写, 旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。

它的特点是**可扩展**, **高性能**, **易使用**, **模式自由**, 存储数据非常方便等, 主要功能特性有:

- ◆ 面向文档存储: (类 JSON 数据模式简单而强大)。
- ◆ 高效的传统存储方式: 支持二进制数据及大型对象 (如照片和视频)。
- ◆ 复制及自动故障转移: Mongo 数据库支持服务器之间的数据复制, 支持主-从模式及服务器之间的相互复制。
- ◆ Auto-Sharding 自动分片支持云级扩展性 (处于早期 alpha 阶段): 自动分片功能支持水平的数据库集群, 可动态添加额外的机器。
- ◆ 动态查询: 它支持丰富的查询表达式。查询指令使用 JSON 形式的标记, 可轻易查询文档中内嵌的对象及数组。
- ◆ 全索引支持: 包括文档内嵌对象及数组。Mongo 的查询优化器会分析查询表达式, 并生成一个高效的查询计划。
- ◆ 支持 RUBY, PYTHON, JAVA, C++, PHP 等多种语言。

1.2 适用范围

适用场景:

- ◆ 适合实时的插入, 更新与查询, 并具备应用程序实时数据存储所需的复制及高度伸缩性。
- ◆ 适合作为信息基础设施的持久化缓存层。
- ◆ 适合由数十或数百台服务器组成的数据库。因为 Mongo 已经包含对 MapReduce 引擎的内置支持。
- ◆ Mongo 的 BSON 数据格式非常适合文档化格式的存储及查询。

不适用场景:

- ◆ 高度事务性的系统。
- ◆ 传统的商业智能应用。
- ◆ 极为复杂的 SQL 查询。

第2章 MongoDB 下载及安装

2.1 下载地址

<http://www.mongodb.org/downloads>

选择一个稳定的版本 **v1.4.5**，如下图：

	OS X 32-bit	OS X 64-bit	Linux 32-bit	Linux 64-bit	Windows 32-bit	Windows 64-bit	Solaris i86pc	Solaris 64	Source
Production Release (Recommended)									
1.4.4									
6/29/2010	OS X 10.5+	download	download	download	download	download	download	download	tgz
Changelog	OS X 10.4		*legacy-static	*legacy-static					zip
Release Notes									
Nightly									
Changelog	OS X 10.5+	download	download	download	download	download	download	download	tgz
	OS X 10.4		*legacy-static	*legacy-static	download				zip

2.2 启动

通过 `mongod --install` 命令把 mongodb 注册成为 window service。

- 1) 创建数据库存储目录；例如：d:\data\db
- 2) 通过命令行执行：

```
mongod --logpath D:\mongodb\logs\mongodb.log --logappend --dbpath D:\mongodb\data  
--directoryperdb --install
```

```
./mongod --fork --logpath /opt/mongodb/logs/mongodb.log --logappend --dbpath  
/opt/mongodb/data --directoryperdb
```

【注：将 mongodb 安装成服务，装该服务绑定到 IP127.0.0.1，日志文件为 d:\data\logs，以及添加方式记录。数据目录为 d:\data\db。并且每个数据库将储存在一个单独的目录（--directoryperdb）】

安装成功后，如下图：



-
- 3) 启动服务后, 尝试是否可用, 通过命令行进入%MONGODB_HOME%\bin下执行 mongo.exe 命令后出现如下图所示信息表示连接成功:

```
D:\mongodb\bin>mongo.exe
MongoDB shell version: 1.4.5-pre-
url: test
connecting to: test
type "exit" to exit
type "help" for help
> _
```

第3章 MongoDB 语法

3.1 基本命令

3.1.1. 启动 mongod

run 直接启动 (net start “MongoDB”):

例如: mongod run

```
C:\Documents and Settings\ce>d:
D:\>cd mongodb\bin
D:\mongodb\bin>mongod run
Fri Jul 16 10:49:54 Mongo DB : starting : pid = 0 port = 27017 dbpath = /data/db
/ master = 0 slave = 0 32-bit
```

--dbpath 指定存储目录启动:

例如: mongod - dbpath = d:\ db

```
D:\mongodb\bin>mongod --dbpath=d:\DB
Fri Jul 16 10:52:21 Mongo DB : starting : pid = 0 port = 27017 dbpath = d:\DB ma
ster = 0 slave = 0 32-bit
```

--port 指定端口启动: (默认端口是:27017)

例如: mongod --port 12345。

3.1.2. 停止 mongod

在窗口模式中, 可以直接使用 Ctrl+C 停止服务。

3.2 SQL 语法

3.2.1. 基本操作

<code>db.AddUser(username, password)</code>	添加用户
<code>db.auth(usrename, password)</code>	设置数据库连接验证
<code>db.cloneDataBase(fromhost)</code>	从目标服务器克隆一个数据库
<code>db.commandHelp(name)</code>	returns the help for the command
<code>db.copyDatabase(fromdb, todb, fromhost)</code>	复制数据库 fromdb---源数据库名称, todb---目标数据库名称, fromhost---源数据库服务器地址
<code>db.createCollection({name, {size:3333, capped:333, max:88888}})</code>	创建一个数据集, 相当于一个表
<code>db.currentOp()</code>	取消当前库的当前操作
<code>db.dropDataBase()</code>	删除当前数据库
<code>db.eval(func, args)</code>	run code server-side
<code>db.getCollection(cname)</code>	取得一个数据集合, 同用法: <code>db['cname']</code> or
<code>db.getCollenctionNames()</code>	取得所有数据集合的名称列表
<code>db.getLastError()</code>	返回最后一个错误的提示消息
<code>db.getLastErrorObj()</code>	返回最后一个错误的对象
<code>db.getMongo()</code>	取得当前服务器的连接对象 get the server
<code>db.getMondo().setSlaveOk()</code>	allow this connection to read from then nonmaster membr of a replica pair
<code>db.getName()</code>	返回当操作数据库的名称
<code>db.getPrevError()</code>	返回上一个错误对象
<code>db.getProfilingLevel()</code>	
<code>db.getReplicationInfo()</code>	获得重复的数据
<code>db.getSisterDB(name)</code>	get the db at the same server as this onew
<code>db.killOp()</code>	停止(杀死)在当前库的当前操作
<code>db.printCollectionStats()</code>	返回当前库的数据集状态
<code>db.printReplicationInfo()</code>	

<code>db.printSlaveReplicationInfo()</code>	
<code>db.printShardingStatus()</code>	返回当前数据库是否为共享数据库
<code>db.removeUser(username)</code>	删除用户
<code>db.repairDatabase()</code>	修复当前数据库
<code>db.resetError()</code>	
<code>db.runCommand(cmdObj)</code>	run a database command. if cmdObj is a string, turns it into {cmdObj:1}
<code>db.setProfilingLevel(level)</code>	0=off, 1=slow, 2=all
<code>db.shutdownServer()</code>	关闭当前服务程序
<code>db.version()</code>	返回当前程序的版本信息

3.2.2. 数据集(表)操作

<code>db.test.find({id:10})</code>	返回 test 数据集 ID=10 的数据集
<code>db.test.find({id:10}).count()</code>	返回 test 数据集 ID=10 的数据总数
<code>db.test.find({id:10}).limit(2)</code>	返回 test 数据集 ID=10 的数据集从第二条开始的数据集
<code>db.test.find({id:10}).skip(8)</code>	返回 test 数据集 ID=10 的数据集从 0 到第八条的数据集
<code>db.test.find({id:10}).limit(2).skip(8)</code>	返回 test 数据集 ID=10 的数据集从第二条到第八条的数据
<code>db.test.find({id:10}).sort()</code>	返回 test 数据集 ID=10 的排序数据集
<code>db.test.findOne([query])</code>	返回符合条件的一条数据
<code>db.test.getDB()</code>	返回此数据集所属的数据库名称
<code>db.test.getIndexes()</code>	返回些数据集的索引信息
<code>db.test.group({key:...,initial:...,reduce:...[,cond:...]}))</code>	
<code>db.test.mapReduce(mapFunction,reduceFunction,<optional params>)</code>	
<code>db.test.remove(query)</code>	在数据集中删除一条数据
<code>db.test.renameCollection(newName)</code>	重命名些数据集名称
<code>db.test.save(obj)</code>	往数据集中插入一条数据
<code>db.test.stats()</code>	返回此数据集的状态
<code>db.test.storageSize()</code>	返回此数据集的存储大小
<code>db.test.totalIndexSize()</code>	返回此数据集的索引文件大小

<code>db.test.totalSize()</code>	返回些数据集的总大小
<code>db.test.update(query,object[,upsert_bool])</code>	在此数据集中更新一条数据
<code>db.test.validate()</code>	验证此数据集
<code>db.test.getShardVersion()</code>	返回数据集共享版本号

MongoDB 语法与现有关系型数据库 SQL 语法比较

MongoDB 语法

MySql 语法

<code>db.test.find({'name':'foobar'})</code>	<code><==> select * from test where name='foobar'</code>
<code>db.test.find()</code>	<code><==> select * from test</code>
<code>db.test.find({'ID':10}).count()</code>	<code><==> select count(*) from test where ID=10</code>
<code>db.test.find().skip(10).limit(20)</code>	<code><==> select * from test limit 10,20</code>
<code>db.test.find({'ID':{'\$in':[25,35,45]}})</code>	<code><==> select * from test where ID in (25,35,45)</code>
<code>db.test.find().sort({'ID':-1})</code>	<code><==> select * from test order by ID desc</code>
<code>db.test.distinct('name',{'ID':{'\$lt:20'}})</code>	<code><==> select distinct(name) from test where ID<20</code>
<code>db.test.group({key: {'name':true},cond: {'name':'foo'},reduce:function(obj,prev){prev.msum+=obj.marks;},initial:{msum:0}})</code>	<code><==> select name,sum(marks) from test group by name</code>
<code>db.test.find('this.ID<20',{name:1})</code>	<code><==> select name from test where ID<20</code>
<code>db.test.insert({'name':'foobar','age':25})</code>	<code><==> insert into test ('name','age') values('foobar',25)</code>
<code>db.test.remove({})</code>	<code><==> delete * from test</code>
<code>db.test.remove({'age':20})</code>	<code><==> delete test where age=20</code>
<code>db.test.remove({'age':{'\$lt:20'}})</code>	<code><==> delete test where age<20</code>
<code>db.test.remove({'age':{'\$lte:20'}})</code>	<code><==> delete test where age<=20</code>
<code>db.test.remove({'age':{'\$gt:20'}})</code>	<code><==> delete test where age>20</code>
<code>db.test.remove({'age':{'\$gte:20'}})</code>	<code><==> delete test where age>=20</code>
<code>db.test.remove({'age':{'\$ne:20'}})</code>	<code><==> delete test where age!=20</code>
<code>db.test.update({'name':'foobar'},{\$set:{'age':36}})</code>	<code><==> update test set age=36 where name='foobar'</code>
<code>db.test.update({'name':'foobar'},{\$inc:{'age':3}})</code>	<code><==> update test set age=age+3 where</code>

```
name='foobar'
```

第4章 JAVA 操作 MongoDB

4.1 正在整理中……

第5章 其它

5.1 正在整理中……