MongoDB介绍

## 基础篇

## 一：简介

MongoDB 是一个高性能，开源，无模式的文档型数据库，是当前NoSQL 数据库产品中最热

门的一种。它在许多场景下可用于替代传统的关系型数据库或键/值存储方式，MongoDB 使

用C++开发。MongoDB 的官方网站地址是：http://www.mongodb.org/。

## 二：安装、配置、启动

1.windows安装。

下载安装包https://www.mongodb.org/downloads



双击安装，一步一步来就可以了。这里不说明。

## 三：概念

1.文档

mongoDB的基本单位，相当于关系数据库中的行。它是一组有序的key/value键值对，

使用json格式，如：{"foo" : 3, "greeting" : "Hello, world!"}。

注意以下几点

* 不能包含\0字符，用于标志key的技术
* .和$字符在mangodb中有特殊含义，如$被用于修饰符($inc表示更新修饰符)，应该考虑保留，以免被驱动解析
* 以\_开始的key也应该保留，比如\_id是mangodb中的关键字
* mangodb中key是不能重复的
* mangodb是类型和大小写敏感的，如{"foo" : 3}和{"foo" : "3"}是两个不同的document，{"foo" : 3}和{"Foo" : 3}类似

2.集合

相当于关系数据库的表，不过没有数据结构的定义。它有多个document组成。

因为是无结构定义的，所以你可以把任何document存入一个collection里。每个

collection用一个名字标识，需要注意以下几点：

* 名字不允许是空字符串""
* 名字不能包含\0字符，因为它表示名字的结束
* 不能创建以system.开头的

3.数据库

一个mongoDB的实例可以运行多个database，database之间是完全独立的，每个database有自己的权限，每个database存储于磁盘的不同文件。

同collections的name一样，databases的name可以是任意的UTF-8字符串。但是有以下限制：

* 空字符串””是非法的
* 不允许出现’’,.,$,/,\,\0字符
* 建议名称都是小写
* 不能超过64个字节
* 保留的名字不能用在databases的name，比如local

## 四：MongoDB Shell

MongoDB Shell 是MongoDB 自带的交互式Javascript shell，用来对MongoDB 进行操作和管理

的交互式环境。

如：使用 "mongod --help" 可查看相关连接参数

下面简单介绍常用的shell操作，这里以windows下的操作为例：

1.启动数据库

 *mongod --dbpath E:\MongoDB\standalone\data\db*

mongod的重要参数

--dbpath:

数据文件存放路径，每个数据库会在其中创建一个子目录，用于防止同一个实例多次运

行的mongod.lock 也保存在此目录中。

--port

指定端口

-- logpath

错误日志文件

--logappend

错误日志采用追加模式（默认是覆写模式

还有一些其他参数，利用mongod --help查询

2.停止数据库

1. Control+C

2. db.shutdownServer()

3. linux下直接kill掉响应的pid

 ps -ef|grep mongod来查找pid

 kill -2 查找到的pid (最好不要kill -9,可能导致mongodb数据损坏)

3.连接数据库

前提最好将bin目录加入到path中



默认连接本地host:27017

若连接非本地： mongo host:port

 4、查询本地所有数据库名称

> show dbs;

 5、切换至指定数据库环境（若无指定的数据库，则创建新的库）

> use mydb;

6、查询当前库下的所有聚集集合collection（相当于table）

> show collections;

7、创建聚集集合

> db.createCollection('mycollection');

## 五：数据类型

MongoDB的文件存储格式为BSON,同JSON一样支持往其它文档对象和数组中再插入文档对象和数组，同时扩展了JSON的数据类型.与数据库打交道的那些应用。例如，JSON没有日期类型，这会使得处理本来简单的日期问题变得非常繁琐。只有一种数字类型，没法区分浮点数和整数，更不能区分32位和64位数字。也没有办法表示其他常用类型，如正则表达式或函数。

下面是MongoDB的支持的数据类型：

* null    null用于表示空值或者不存在的字段。 {"x":null}
* 布尔   布尔类型有两个值'true'和'false1'. {"X":true}
* 32位整数  类型不可用。JavaScript仅支持64位浮点数，所以32位整数会被自动转换。
* 64位整数  不支持这个类型。shell会使用一个特殊的内嵌文档来显示64位整数，
* 64位浮点数  shell中的数字都是这种类型。下面的表示都是浮点数： {"X" : 3.1415926} {"X" : 3}
* 字符串   UTF-8字符串都可表示为字符串类型的数据： {"x" : "foobar"}
* 符号  不支持这种类型。shell将数据库里的符号类型转换成字符串。
* 对象id  对象id是文档的12字节的唯一 ID, {"X" :ObjectId() }
* 日期  日期类型存储的是从标准纪元开始的毫秒数。不存储时区： {"X" ： new Date()}
* 正则表达式  文档中可以包含正则表达式，采用JavaScript的正则表达式语法: {"x" ： /foobar/i}
* 代码  文档中还可以包含JavaScript代码：{"x" ： function() { /\* …… \*/ }}
* 二进制数据  二进制数据可以由任意字节的串组成。不过shell中无法使用。
* 最大值  BSON包括一个特殊类型，表示可能的最大值。shell中没有这个类型。
* 最小值  BSON包括一个特殊类型，表示可能的最小值。shell中没有这个类型。
* 未定义  文档中也可以使用未定义类型:{"x":undefined}
* 数组  值的集合或者列表可以表示成数组：{"x" ： ["a", "b", "c"]}
* 内嵌文档  文档可以包含别的文档，也可以作为值嵌入到父文档中，数据可以组织得更自然些，不用非得存成扁平结构的：{"x" ： {"food" ： "noodle"}}

## 六：客户端工具

MongoVUE

