# mongoDB索引篇

上文中对索引页开了一个简单的小头，这篇教程就来详细叙述一下mongoDB的索引

## 索引的作用，含义和特点

索引是对数据库表中一列或多列的值进行排序的一种结构，使用索引可快速访问数据库表中的特定信息。

数据库索引好比是一本书前面的目录，能加快数据库的查询速度。

建立索引的目的是加快对表中记录的查找或排序。为表设置索引要付出代价的：一是增加了数据库的存储空间，二是在插入和修改数据时要花费较多的时间(因为索引也要随之变动)。数据库索引就是为了提高表的搜索效率而对某些字段中的值建立的目录 。

## 索引的建立

在增删改查篇的末尾其实提及了建立单个索引的语句

db.collectionname.ensureIndex({字段名:1/-1}) //其中1为升序索引，-1为降序索引

## 索引的分类

1：\_id索引

2：单键索引

3：多键索引

4：复合索引

5：过期索引

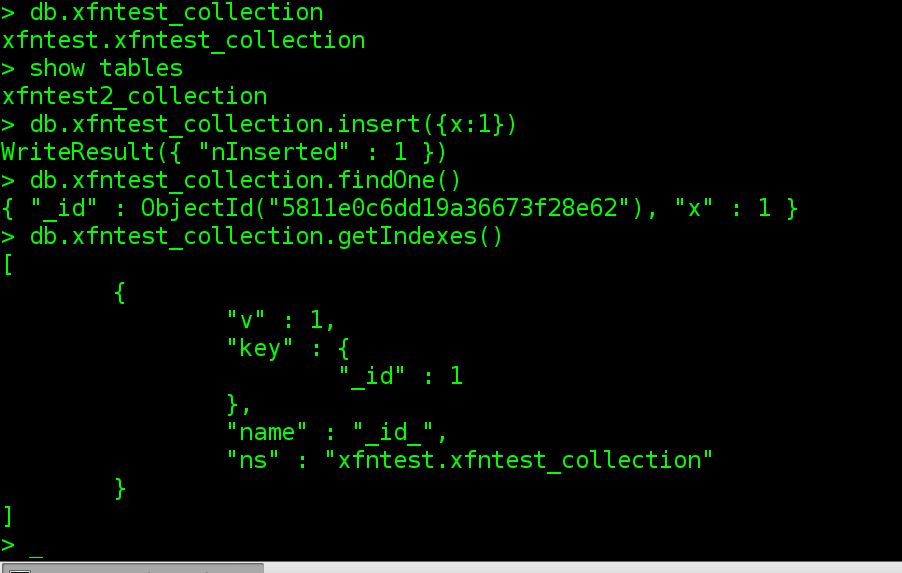
6：全文索引

7：地理位置索引

### \_id索引

\_id索引：绝大多数集合默认建立的索引，对于每个插入的数据，MongoDB都会生成一条唯一的\_id字段。

只要我建立一个空表，插入一条数据，\_id字段（索引）就会自动建立



如上图

建立了xfntest\_collection的表，然后插入一条json型数据{x:1}

然后查看表的索引，发现key已经变成了\_id字段

### 单键索引

1.单键索引是最普通的索引

2.与\_id索引不同，单键索引不会自动创建

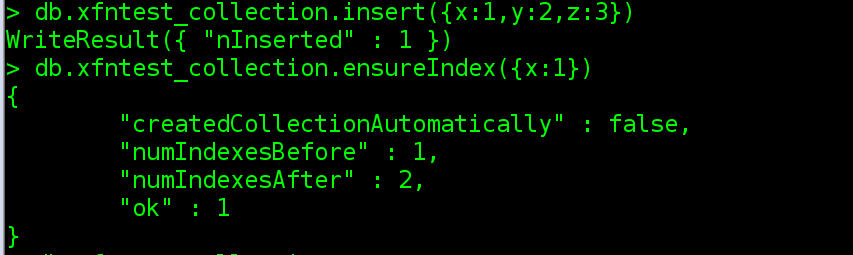
单键索引的建立在上方已经提及

db.collectionname.ensureIndex({字段名:1/-1}) //其中1为升序索引，-1为降序索引

索引可以重复创建，若创建已经存在的索引，则会直接返回成功。

比如我数据库中的数据形式类似于{x:1,y:2,z:3}，我想为x字段建立一个升序索引

db.collectionname.ensureIndex({x:1})





此时出现了两个key(即两个索引)，一个是\_id字段，另外一个就是x字段

### 多键索引

多键索引与单键索引创建形式相同，区别在于字段的值。

1）单键索引：值为一个单一的值，如字符串，数字或日期。

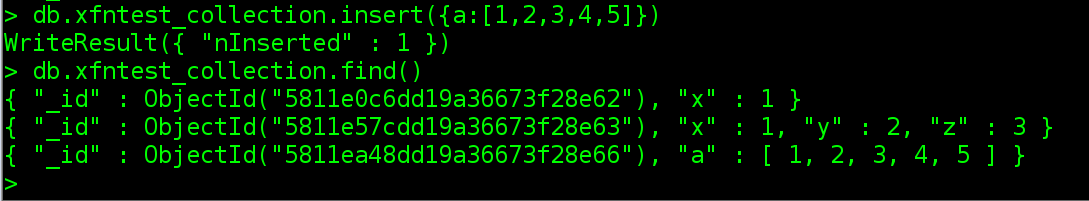
2）多键索引：值具有多个记录，如数组。

先插入一条数组型数据吧

db.collectionname.insert({a:[1,2,3]})

此时已经添加了一个多键索引(插入的同时已经建好索引)

自行来测试db.collectionname.insert({a:[1,2,3]})，然后getIndexes()一下吧，同样没有key为a的索引，但是其实a的索引已经建立，也就是说，如果mongoDB中如果插入数组类型的多键数据，索引是自动建立的，证据就是我们能很方便的对插入的多键类型进行快速的查找和排序，详见<http://blog.csdn.net/xinghebuluo/article/details/7221642>



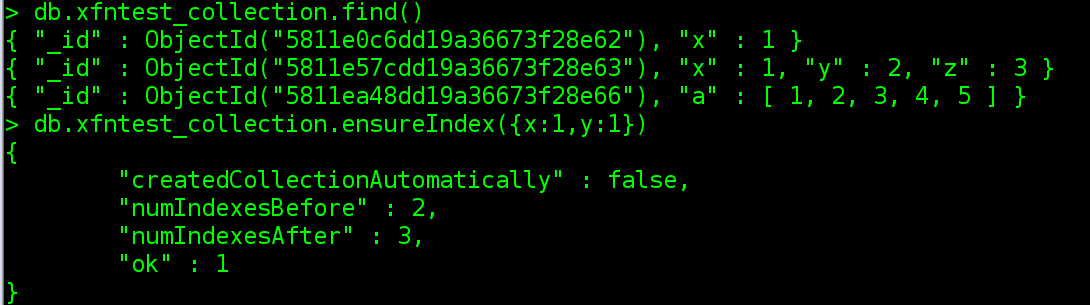
由于多键索引不太好表示，所以此时getIndexes()是找不到多键索引的，但是它确实存在

### 复合索引

有时我们查询的条件不只一个，比如我想查找x为1且y为2的数据

此时需要建立符合索引，其实就是在建立单键索引的基础上添加别的字段再标明一下升序或者降序就可以了

db.collectionname.ensureIndex({字段名a:1/-1, 字段名b:1/-1,…})



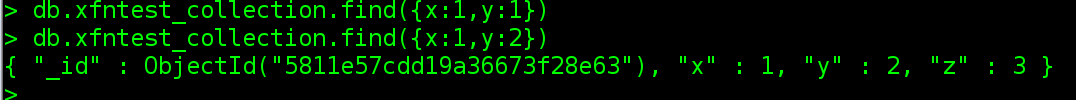
比如我现在想建立关于x和y两个字段的升序索引

db.xfntest\_collection.ensureIndex({x:1,y:1})

简单查看一下索引发现都有了

查询一下x为1且y为1的数据（没有）

查询一下x为1且y为2的数据（发现了）



### 过期索引

定义：是指在一段时间后会过期的索引

1、索引过期后，相应的数据会被删除

2、适合存储一些在一段时间之后会失效的数据，比如用户的登录信息，存储的日志

3、建立方法

db.collection.ensureIndex({字段名:1},{expireAfterSeconds:10}) //创建过期索引，time-字段，expireAfterSeconds在多少秒后过期，单位：秒（注意expireAfterSeconds遵循驼峰法则，不要拼错，否则不会删除）

a.存储在过期索引字段的值必须是指定的时间类型；

必须是ISODate或者ISODate数组，不能使用时间戳，除标准时间之外的过期索引都不能被自动删除，比如指定time:22

比如使用new Date()创建的时间类型就是ISODate

b.如果指定了ISODate数组，则按照快到期的时间进行删除

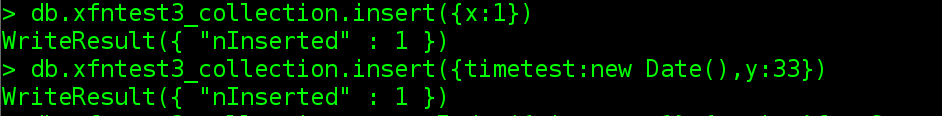
c.过期索引不能是复合索引，因为不能指定两个字段均为时间字段，否则容易冲突

d.删除时间是不精确的

删除过程是由后台程序每60s跑一次，而且删除也需要一些时间，所以存在误差

如果设置时间很小，则会在60秒内才可以删除，而不是在规定时间内删除

也就是说expireAfterseconds指定小于60的值意义不大，因为最少也要60s之后执行

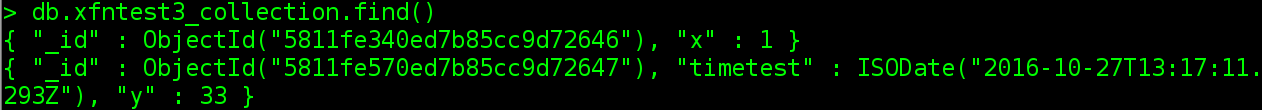


添加两条数据，一条普通，一条名字为我们想设置的过期索引



过期索引建立完成了

此时看一下数据

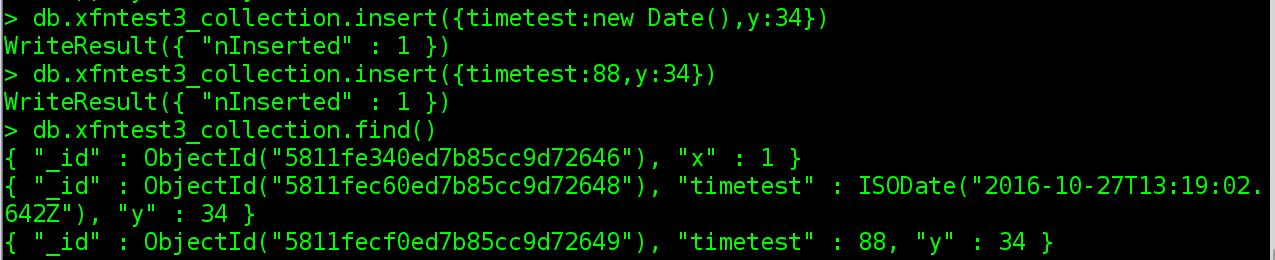


一条普通数据和一条包含过期索引时间为13:17:11的数据

过了1分钟

再插入两条数据包含过期索引，但是timetest值一个为ISODate一个为普通数字

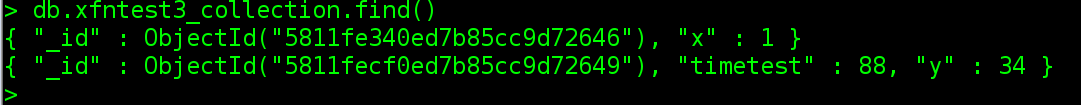
再看一下数据



包含过期索引时间为13:17:11的数据被自动删除

添加了包含过期索引时间为13:19:02的数据和为88的数据

又过了1分钟



包含过期索引时间为13:19:02的数据被删除，包含88的普通数据还在

说明两点

1. 在过期索引建立后，无论数据是在过期索引建立前还是建立后插入，只要过期时间到了就会被删除
2. 想要数据过期被删除，那么插入数据的过期索引字段值就一定要是ISODate或者ISODate数组的形式