

关于_id和Objectid

MongoDB中存储的文档必须有一个"_id"键。这个键的值可以是任意类型，默认是个ObjectId对象。在一个集合里，每个文档都有唯一的“_id”，确保集合里每个文档都能被唯一标识。不同集合"_id"的值可以重复，但同一集合内"_id"的值必须唯一

1. ObjectId

ObjectId是"_id"的默认类型。因为设计MongoDb的初衷就是用作分布式数据库，所以能够在分片环境中生成

唯一的标识符非常重要，而常规的做法：在多个服务器上同步自动增加主键既费时又费力，这就是MongoDB采用ObjectId的原因。

ObjectId采用12字节的存储空间，是一个由24个十六进制数字组成的字符串

0,1,2,3	4,5,6	7,8	9,10,11
时间戳	机器	PID	计数器

如果快速创建多个ObjectId，会发现每次只有最后几位有变化。另外，中间的几位数字也会变化（要是在创建过程中停顿几秒）。

这是ObjectId的创建方式导致的，如上图

时间戳单位为秒，与随后5个字节组合起来，提供了秒级的唯一性。这个4个字节隐藏了文档的创建时间，绝大多数驱动程序都会提供一个方法，用于从ObjectId中获取这些信息。

因为使用的是当前时间，很多用户担心要对服务器进行时钟同步。其实没必要，因为时间戳的实际值并不重要，只要它总是不停增加就好。

接下来3个字节是所在主机的唯一标识符。通常是机器主机名的散列值。这样就可以保证不同主机生成不同的ObjectId，不产生冲突

接下来连个字节确保了在同一台机器上并发的多个进程产生的ObjectId是唯一的

前9个字节确保了同一秒钟不同机器不同进程产生的ObjectId是唯一的。最后3个字节是一个自动增加的计数器。确保相同进程的同一秒产生的ObjectId也是不一样的。

2.自动生成_id

如果插入文档时没有"_id"键，系统会自帮你创建一个。可以由MongoDb服务器来做这件事。但通常会在客户端由驱动程序完成。这一做法非常好地体现了MongoDb的哲学：能交给客户端驱动程序来做的事情就不要交给服务器来做。这种理念背后的原因是：即便是像MongoDB这样扩展性非常好的数据库，扩展应用层也要比扩展数据库层容易的多。将工作交给客户端做就减轻了数据库扩展的负担。