

# MongoDB 自动增长

MongoDB 没有像 SQL 一样有自动增长的功能，MongoDB 的 `_id` 是系统自动生成的12字节唯一标识。

但在某些情况下，我们可能需要实现 ObjectId 自动增长功能。

由于 MongoDB 没有实现这个功能，我们可以通过编程的方式来实现，以下我们将在 `counters` 集合中实现 `_id` 字段自动增长。

## 使用 counters 集合

考虑以下 `products` 文档。我们希望 `_id` 字段实现从 1,2,3,4 到 n 的自动增长功能。

```
{
  "_id":1,
  "product_name": "Apple iPhone",
  "category": "mobiles"
}
```

为此，创建 `counters` 集合，序列字段值可以实现自动长：

```
>db.createCollection("counters")
```

现在我们向 `counters` 集合中插入以下文档，使用 `productid` 作为 key:

```
{
  "_id":"productid",
  "sequence_value": 0
}
```

`sequence_value` 字段是序列通过自动增长后的一个值。

使用以下命令插入 `counters` 集合的序列文档中：

```
>db.counters.insert({_id:"productid",sequence_value:0})
```

## 创建 Javascript 函数

现在，我们创建函数 `getNextSequenceValue` 来作为序列名的输入，指定的序列会自动增长 1 并返回最新序列值。在本文的实例中序列名为 `productid`。

```
>function getNextSequenceValue(sequenceName){
  var sequenceDocument = db.counters.findAndModify(
    {
```

```
        query:{_id: sequenceName },
        update: { $inc:{sequence_value:1}},
        "new":true
    });
    return sequenceDocument.sequence_value;
}
```

## 使用 Javascript 函数

接下来我们将使用 getNextSequenceValue 函数创建一个新的文档，并设置文档 \_id 自动为返回的序列值：

```
>db.products.insert({
  "_id":getNextSequenceValue("productid"),
  "product_name":"Apple iPhone",
  "category":"mobiles"})

>db.products.insert({
  "_id":getNextSequenceValue("productid"),
  "product_name":"Samsung S3",
  "category":"mobiles"})
```

就如你所看到的，我们使用 getNextSequenceValue 函数来设置 \_id 字段。

为了验证函数是否有效，我们可以使用以下命令读取文档：

```
>db.products.find()
```

以上命令将返回以下结果，我们发现 \_id 字段是自增长的：

```
{ "_id" : 1, "product_name" : "Apple iPhone", "category" : "mobiles" }

{ "_id" : 2, "product_name" : "Samsung S3", "category" : "mobiles" }
```

[← MongoDB 固定集合 \( Capped Collections \)](#)

[MongoDB 创建数据库 →](#)

[✎ 点我分享笔记](#)