MONGODB - AUTO-INCREMENT SEQUENCE

MongoDB does not have out-of-the-box auto-increment functionality like SQL databases. By default, it uses the 12-byte ObjectId for the **_id** field as primary key to uniquely identify the documents. However, there may be scenarios where we may want the _id field to have some auto-incremented value other than the ObjectId.

Since this is not a default feature in MongoDB, we will programmatically achieve this functionality by using a **counters** collection as suggested by the MongoDB documentation.

Using counter Collection

Consider the following **products** document. We want the _id field to be an **auto-incremented integer sequence** starting from 1,2,3,4 upto n.

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

For this, create a **counters** collection which will keep track of the last sequence value for all the sequence fields.

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

Now, we will insert the following document in the counters collection with **productid** as its key:

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

The field **sequence_value** keeps track of the last value of the sequence.

Use the following code to insert this sequence document in the counters collection:

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

Creating Javascript Function

Now, we will create a function **getNextSequenceValue** which will take the sequence name as its input, increment the sequence number by 1 and return the updated sequence number. In our case, the sequence name is **productid**.

```
>function getNextSequenceValue(sequenceName){
   var sequenceDocument = db.counters.findAndModify(
      {
         query:{_id: sequenceName },
         update: {$inc:{sequence_value:1}},
         new:true
      });
   return sequenceDocument.sequence_value;
}
```

Using the Javascript Function:

We will now use the function getNextSequenceValue while creating a new document and assigning the returned sequence value as document's _id field.

Insert two sample documents using the following code:

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

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

As you can see, we have used the getNextSequenceValue function to set value for the _id field.

To verify the functionality, let us fetch the documents using find command:

```
>db.prodcuts.find()
```

The above query returned the following documents having the auto-incremented _id field:

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

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