**Spring Boot + MongoDB Auto-Generated @Id Example**

we already know how we can autogenerate ID in Spring Data JPA. So, to do that, usually we annotate @**Id** and @**GeneratedValue** in entity class. So, this approach is straightforward, right? But can you do the same in mongodb to autogenerate ID or sequence? No, in mongodb it's a bit tricky approach. And also, this is one of the most frequently asked interview questions.

So, in this tutorial, we'll understand how we can autogenerate sequence or ID in spring data mongo. Okay, so let's get started.

Application - **mongo-sequence-id-example**

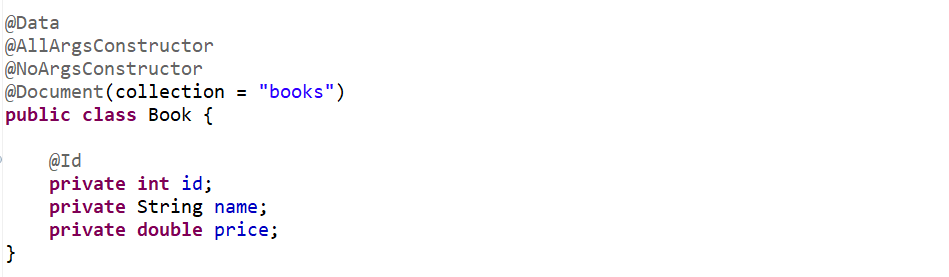
Dependencies – **Lombok, Spring Web, Spring Data Mongo DB**

Let’s create a package called service and entity. Let’s create a Book Entity and add some fields.

Since this is an Entity and we are going to use MongoDB so let’s annotate this class with @**Document**.

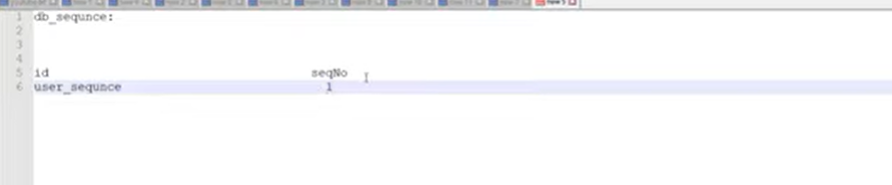
Bcz in mongo the data will be structured as a Document in key value pair. Let’s give the collection name as **books**.

I need to annotate **@ID** if I type here @**GeneratedValue** we are not getting that annotation, right? Because we are not using spring data Jpa. So, then how I can do that in Spring Data Mongo. So, for that we need to do some different logic.



Let's go to the notepad. Basically, we need to create one more document called **db\_sequence** or something like that you can give any name. So, in the **db\_sequence** we need to define two attribute id and sequence number.

So basically, while inserting the document to MongoDB we need to verify this table what is the current sequence number? Get that then increment it to one. So, for example, first time let's say the name of my id is **user\_sequence** the sequence number is first time it is 1.



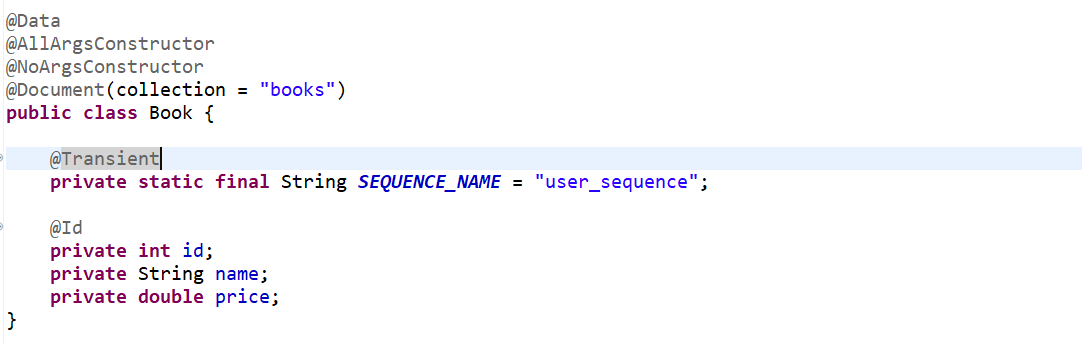
So, second time when I will try to insert any document to mongo first, I need to fetch the sequence number from this table. So, I got the sequence number is one. Now I just need to increment it to two. So that second time when I will insert the record again, I will fetch the sequence number from this table.

Now then I need to increment it. Okay, so this is how we can do that. So, for that, let me create one more document called DB sequence or something like that. So let me create a new entity.

I will name it **DbSequence**, I need to annotate with @**Document**(collection = “**db\_sequence**”) . now I need to define 2 fields **id** as sequenceName and **sequenceNumber**. So, this is another document so let’s annotate with @**Id**, so this id would be your name of your sequence number.

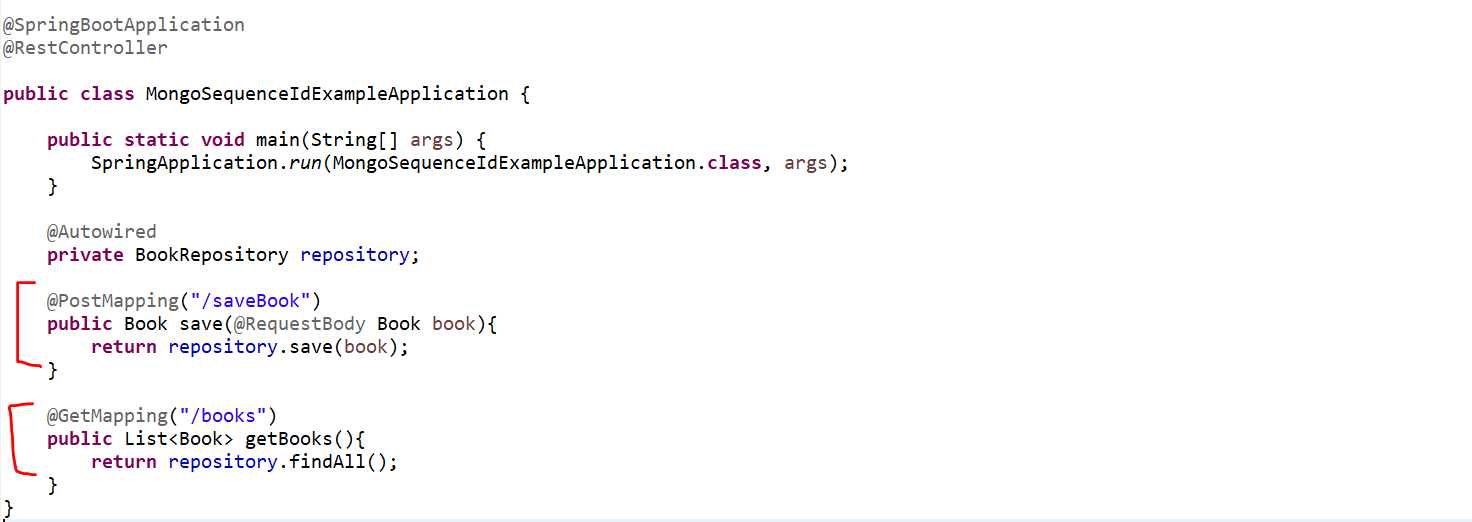


So, we need to define this sequence in our Entity Document Book. And I don’t want this to persist into Mongo so let’s annotate with @Transient.



So, we have created 2 entity one is Book which we are going to persist into our mongo. And another is DbSequence for autogenerated Id.

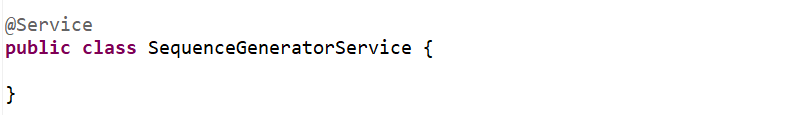
Let’s create a controller with 2 endpoints…



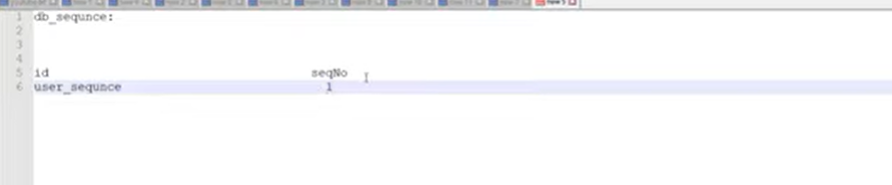
But if you observed here while we are persisting the book, we are not generating the Id. Still, we didn’t write the logic to generate the sequence. So before persisting we need to write a logic to generate sequence.



Let’s create a service called **SequenceGeneratorService**.

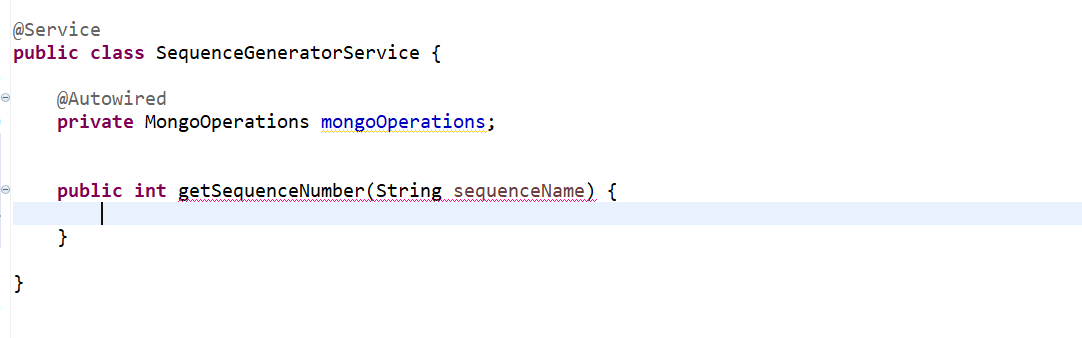


As I already explained we need to check in this db\_sequence collection what is the current sequence number then get it and increment it. So this logic I am going to write in this **SequenceGeneratorService**.

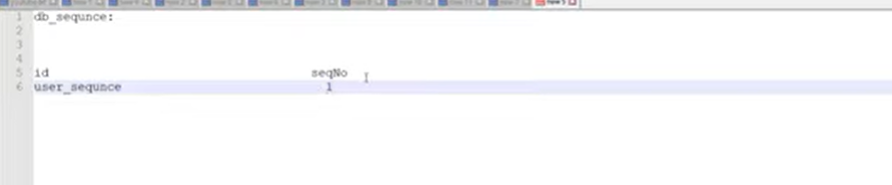


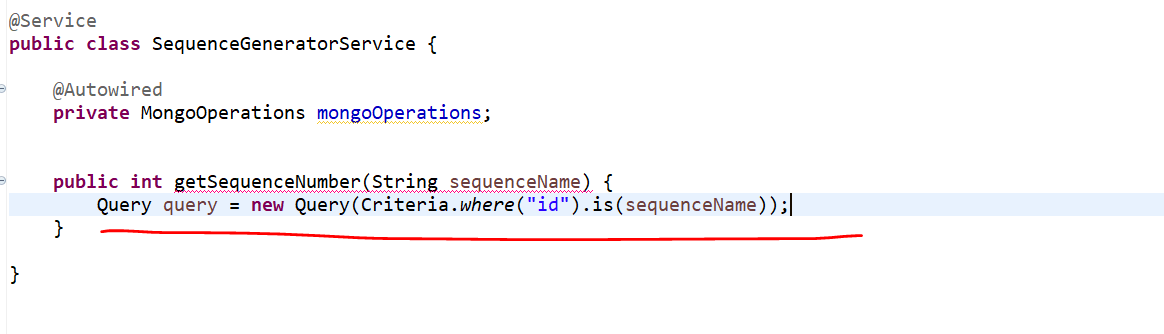
So, for that I need to Inject our **MongoOperations** even u can go for **MongoTemplate**.

Now write one logic which will return us the Sequence Number. So, lets create a method and pass a parameter as Sequence Name.

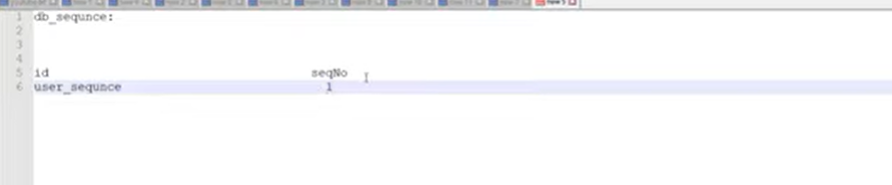


So, this is the sequence name we need to pass this sequence name to this below table then get the sequence number and then increment it. So, this logic is very simple.

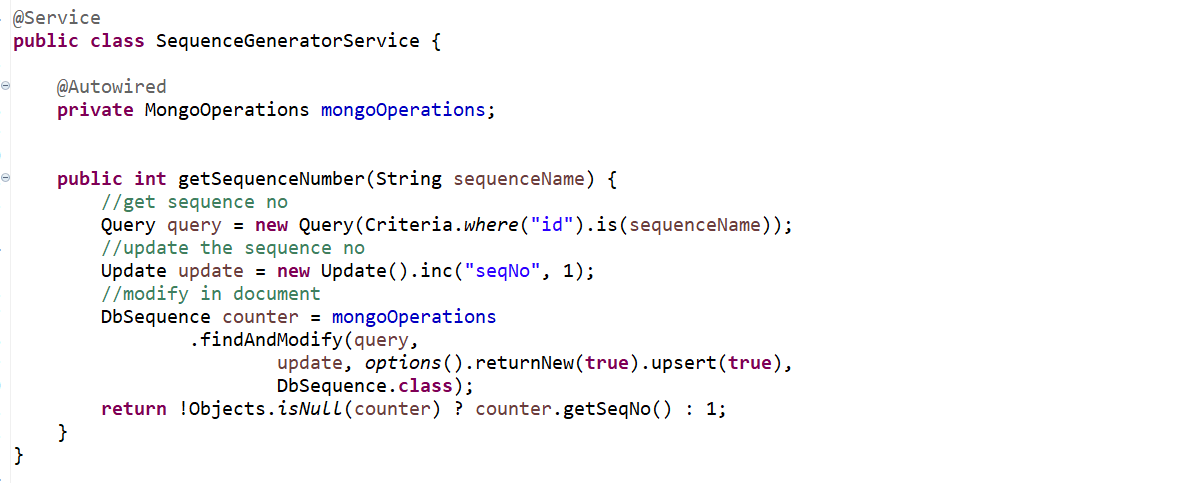




So, just do the filter where id is equal to sequence name. so, if u observed our id is our sequence name which is **user\_sequence** that is what we mentioned in our Book Entity or in **Book** Document so it will just get the corresponding **seqNo** and increment it to 1 using **Update** class.



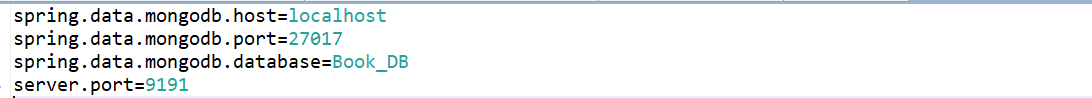
Now we have MongoOperations with us, we already get the query, and we already update it with sequence number. Now we just need to find and modify the table.



We are just using MongoOperations **findAndModify**() we just pass the query, the query which we get the sequence number then we pass the update which we pass for increment then we specify the class type and also, we are adding if its counter is not null then return the **sequence** else return 1 if it’s a first time then get the **1.**

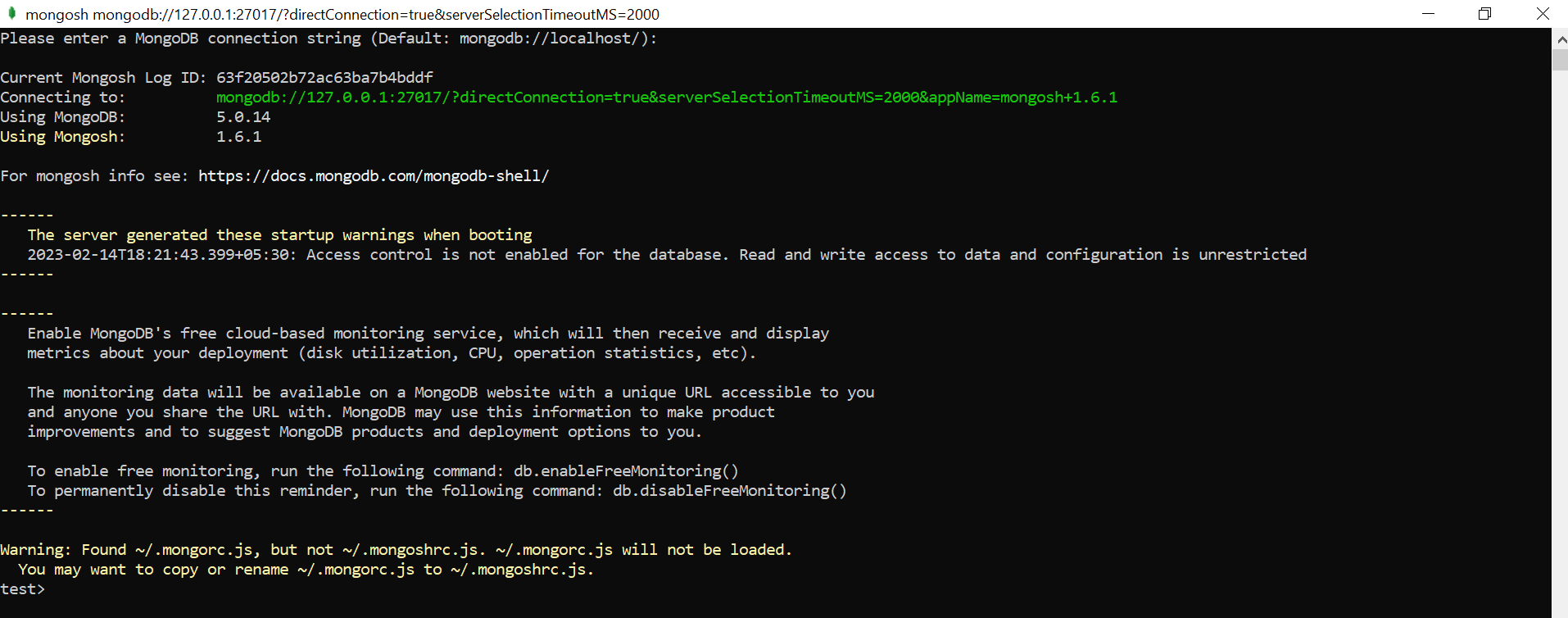
**Now go to controller and before persist get the sequence and update it into the Book document.**

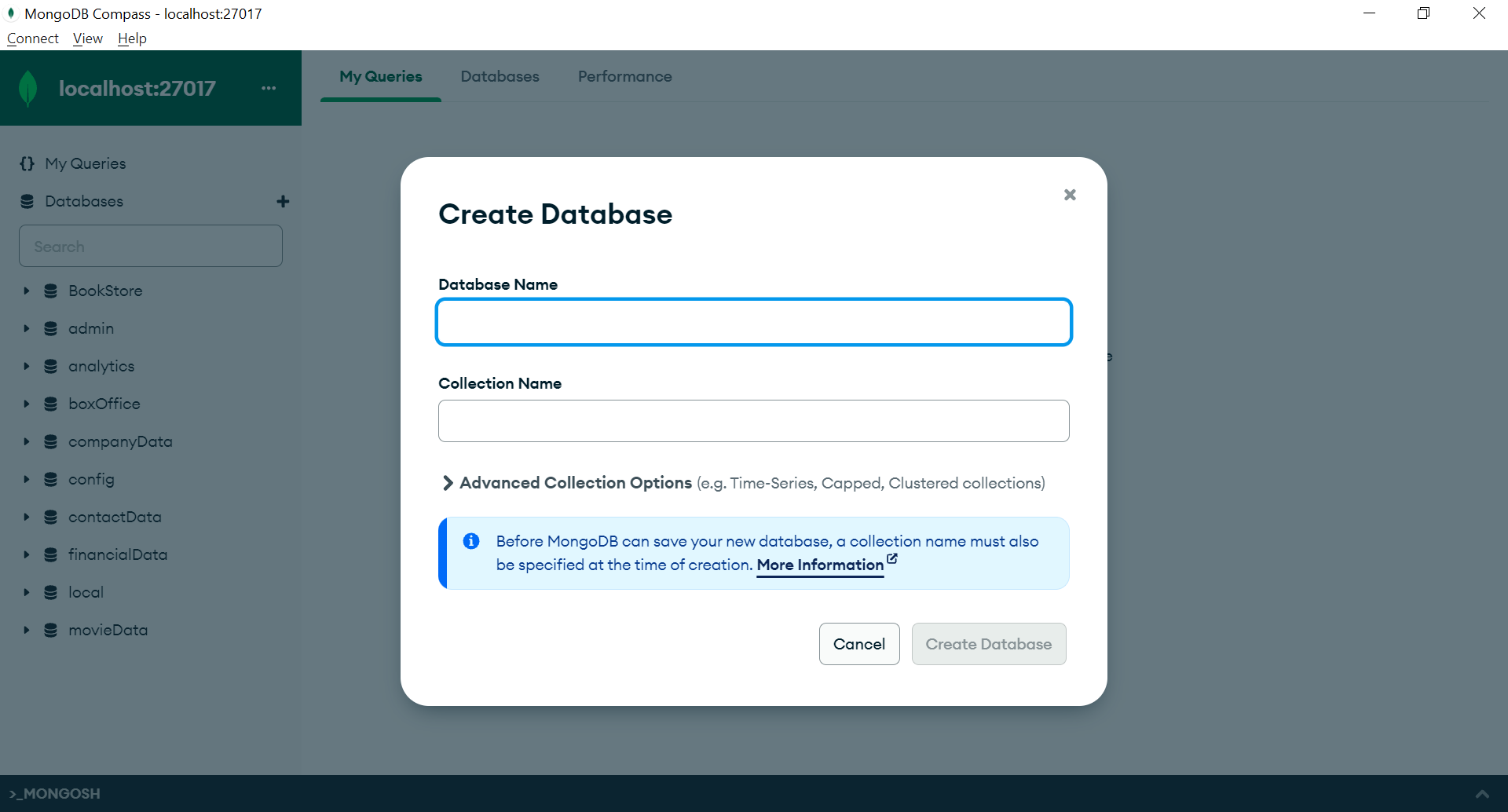
Go to application.properties file---



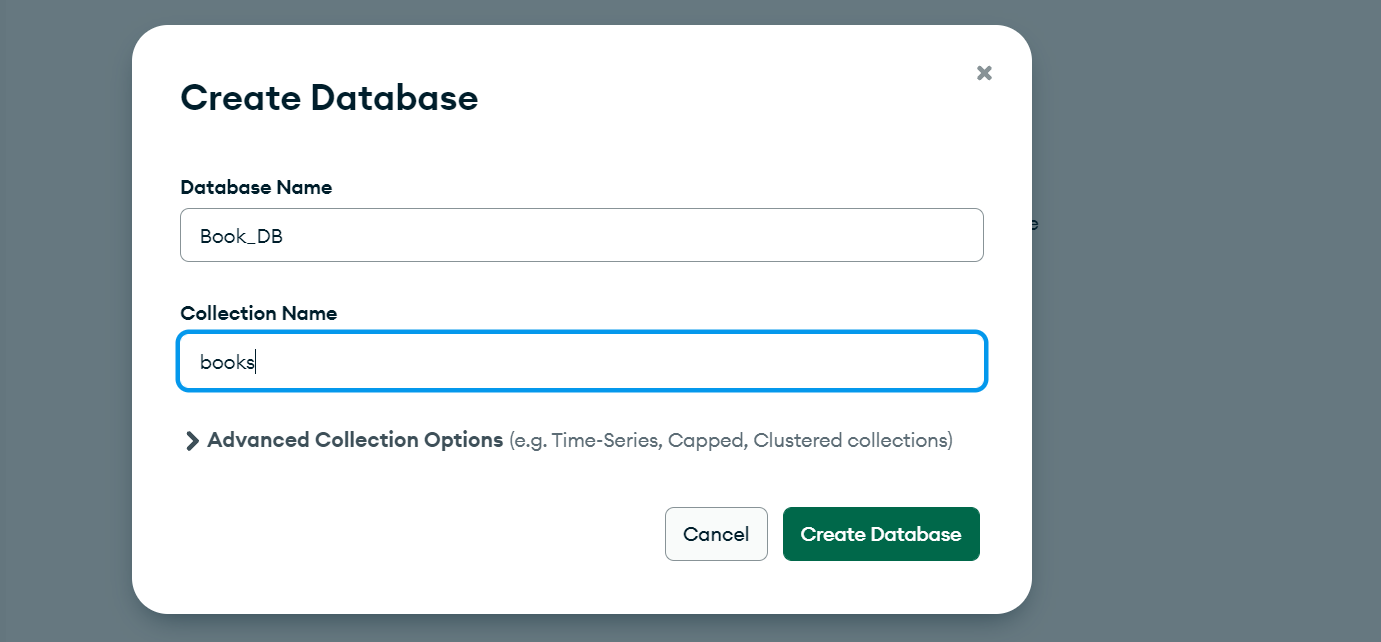
Now go to your mongodb server and start and then go to your Mongo DB Compass Community…

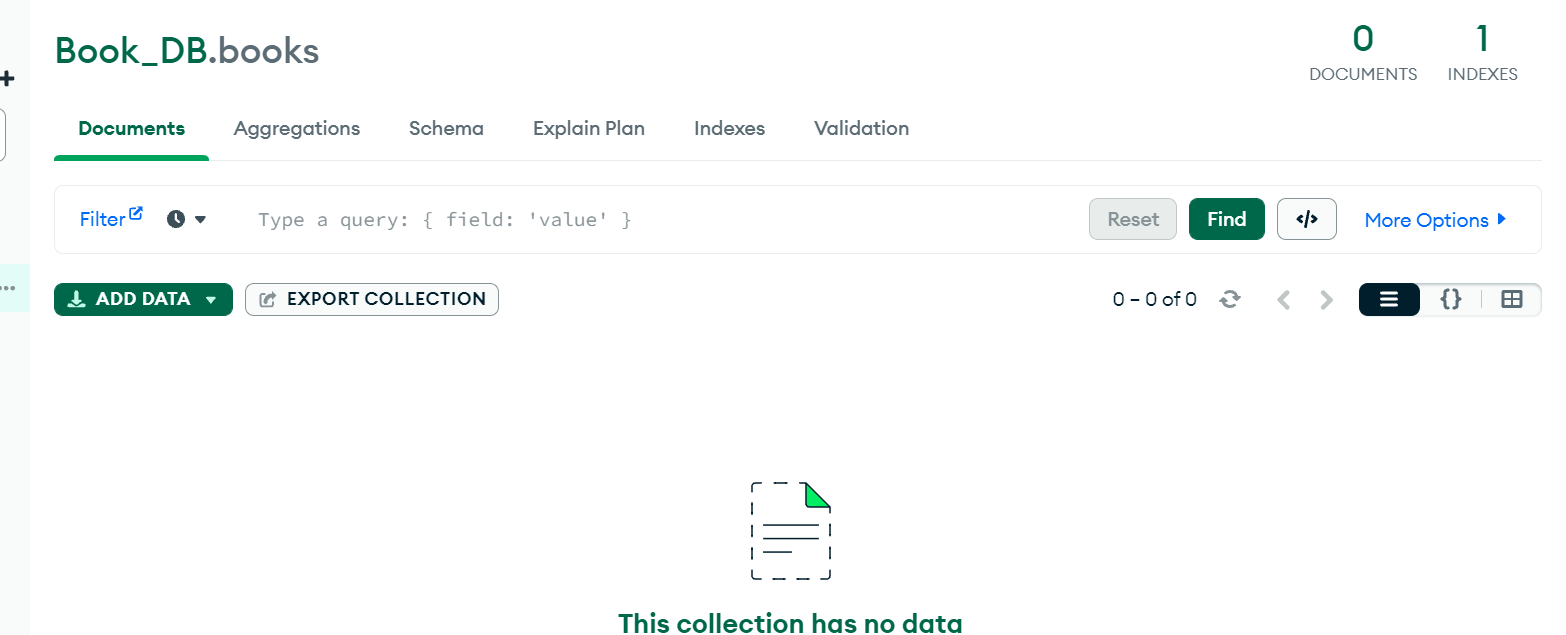
Open mongosh





Now let’s create the database with the same name **Book\_DB**.





If you go inside, it you can see there is no entries yet. So, let’s start our application and let’s try to add from postman. Now in payload we need to add only 2 fields name and price and id is autogenerated.

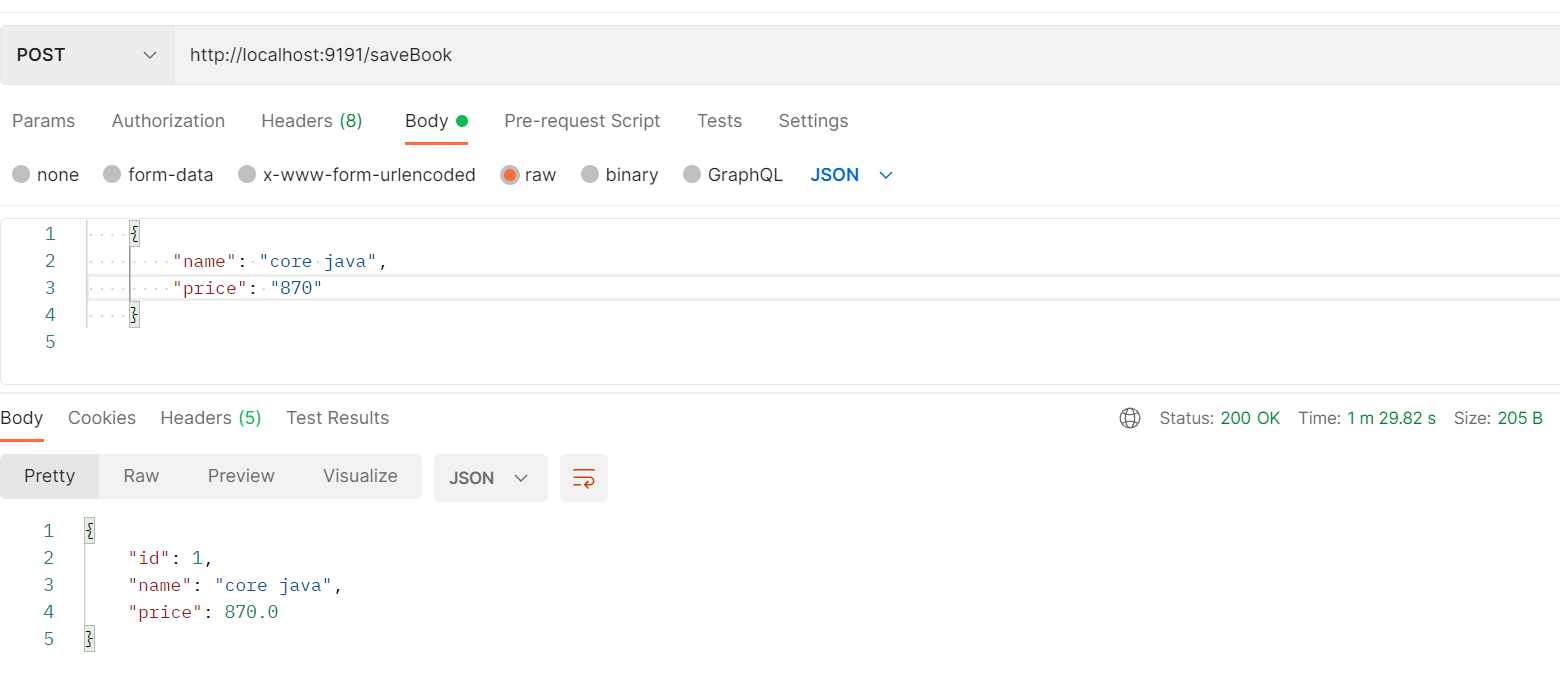
**POST** [**http://localhost:9191/saveBook**](http://localhost:9191/saveBook)

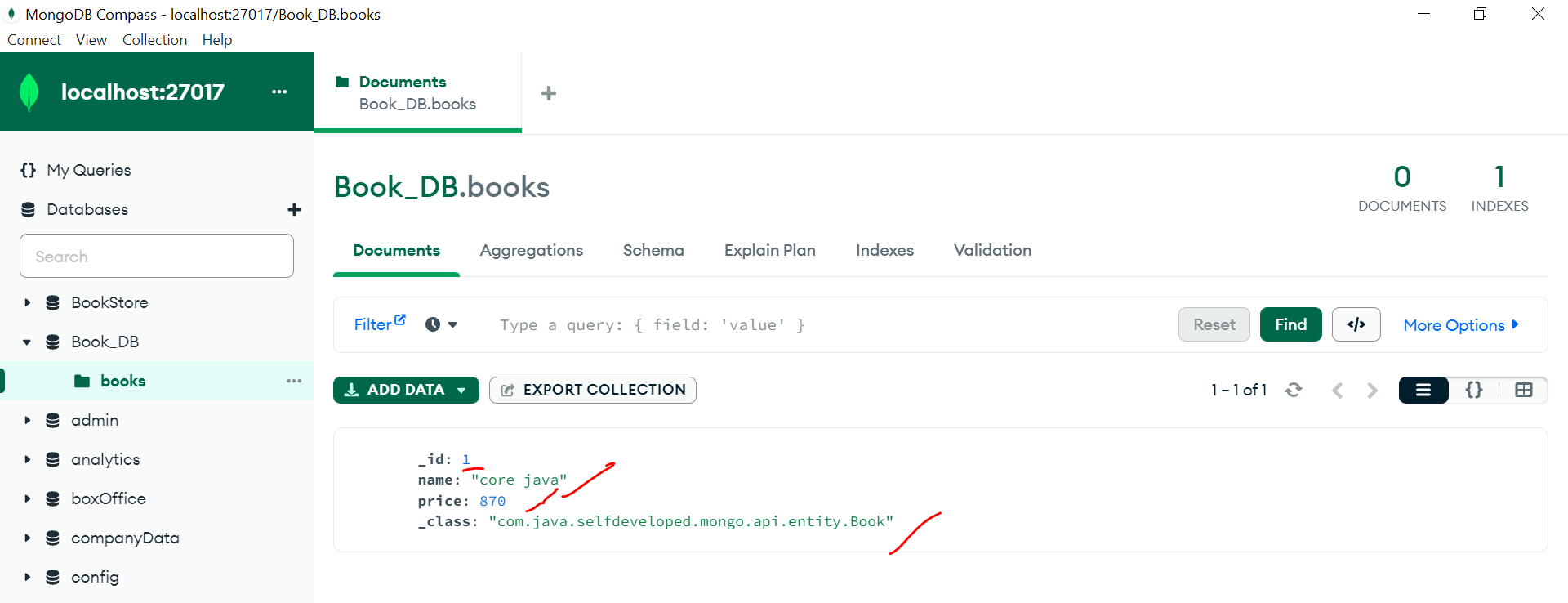
**{**

**"name": "core java",**

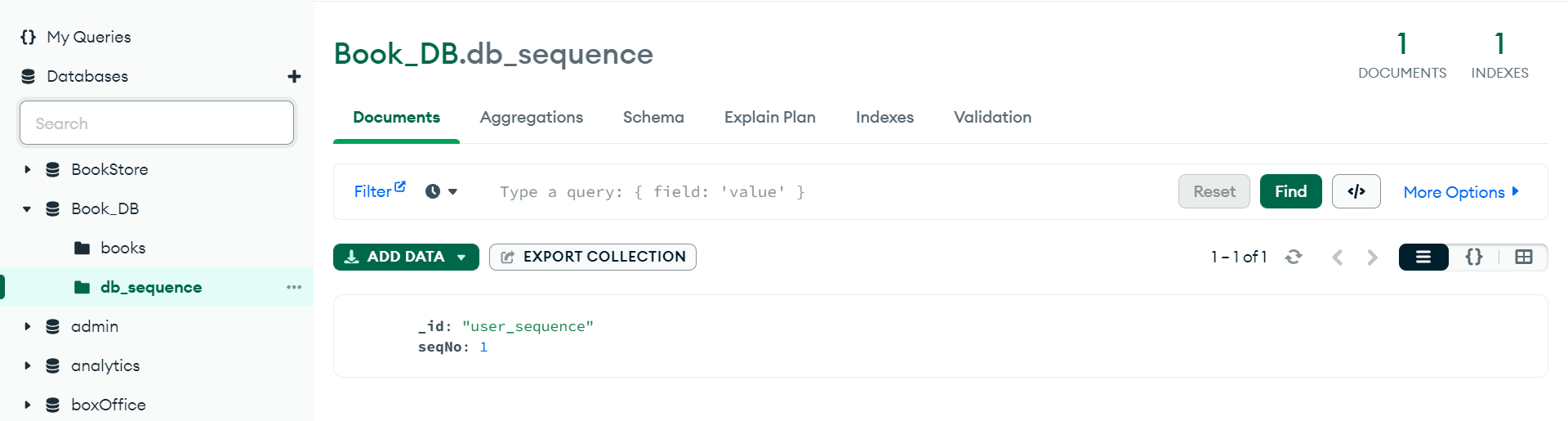
**"price": "870"**

**}**





Also, if you go to your database **Book\_DB** you will find one more document called **db\_sequence**



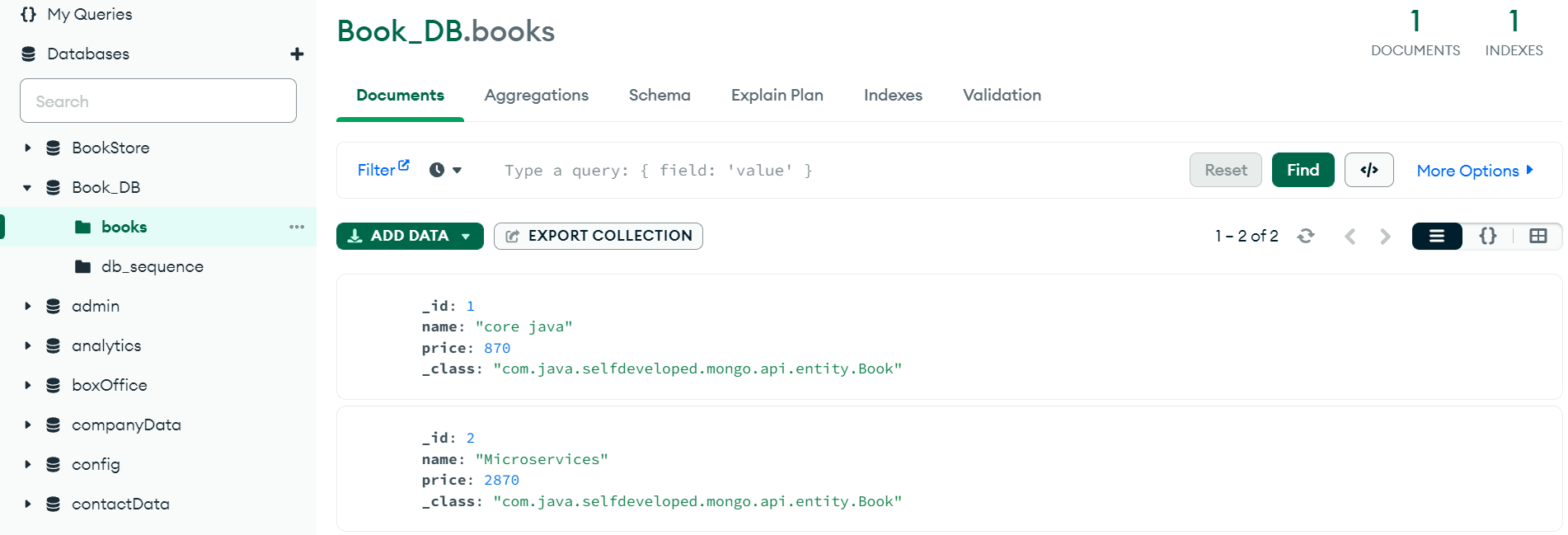
Now let me add one more document…then id will get updated to 2.

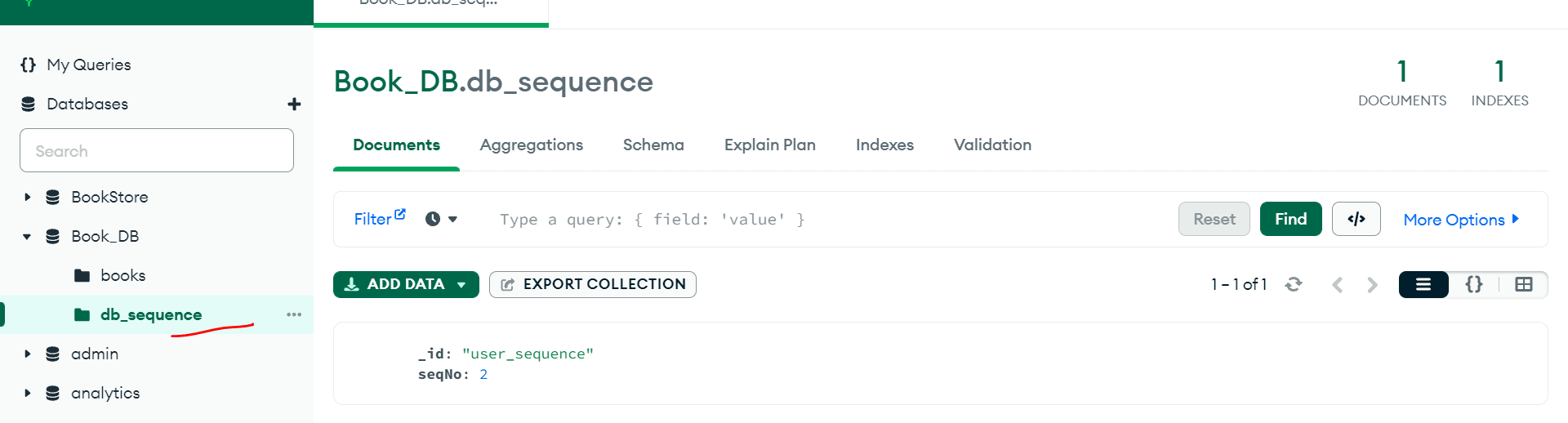
    {

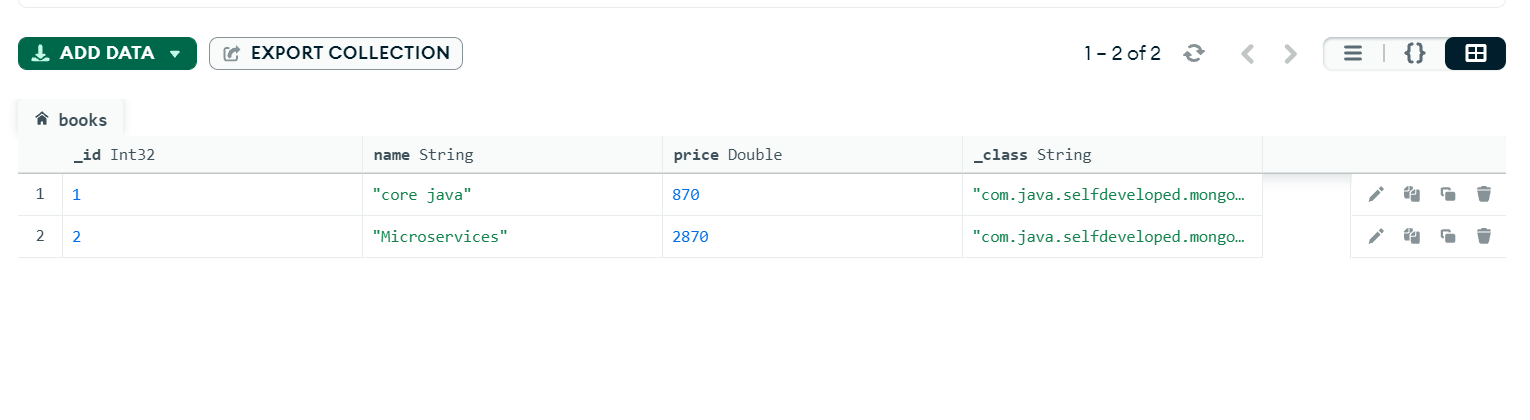
        "name": "Microservices",

        "price": "2870"

    }







This is how we can auto-generate sequence in mongodb and this is one of the most important common interview question.

GET <http://localhost:9191/books>

Response :-

**[**

**{**

**"id": 1,**

**"name": "core java",**

**"price": 870.0**

**},**

**{**

**"id": 2,**

**"name": "Microservices",**

**"price": 2870.0**

**}**

**]**