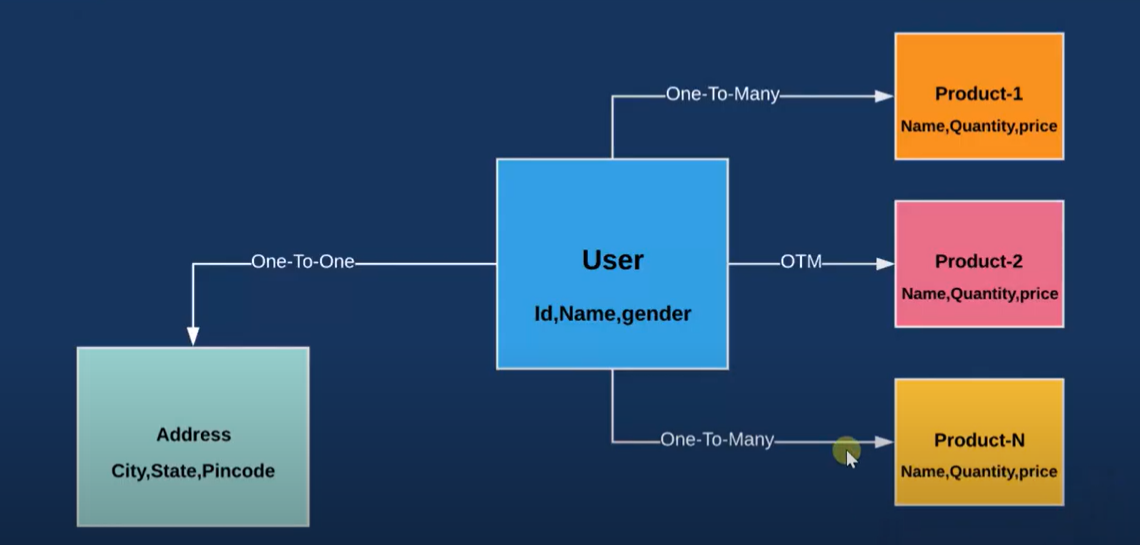
**MongoDB – Model One-to-One, One-to-Many Relationships Embedded Documents**

Application - **spring-mongo-embedded**

Dependencies – **Spring Web, Spring Data Mongo DB, Lombok, Springboot Dev Tool**

Embedded documents to perform association mapping.

Hi guys in this tutorial we are going to see how to perform association mapping in mongodb using embedded documents. So, with mongodb we can structure related data by embedded documents in general, embedding give a better performance for read operation only for search criteria so what we mean by embedded document style so let me show you one architectural diagram so that we will get some brief idea how to embedded the entity.



So, assume this one is something like flipkart purchase order service where a user can buy multiple products and a user can have only one address. So, here user and product having one too many associations and here user and address have one to one association.

So, in normal traditional mapping we are doing user as a separate entity product as a separate and address as a separate entity. So, in three entity, we need to annotate based on the business. Like in this, we need to annotate order.

It one too many. We need to pass the target entity and we need to specify the join column similarly for one to one also we are writing ordered one to one and we are specifying the primary join key. And each table suppose in case of user and product in product table there will be one foreign key which should be the primary key of this user. Similarly in address also it maps so while performing search operation we need to write like bunch of joint statement.

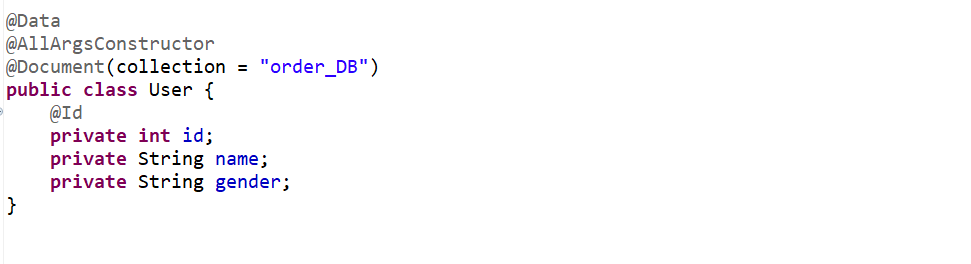
So, to ignore it the mongodb provided these embedded documents features so let's do a quick example on it then we will get to know. How we can embedded the documents. Documents is nothing like it is the model.

Embedded documents to perform association mapping.

Application - **spring-mongo-embedded**

Dependencies – **Spring Web, Spring Data Mongo DB, Lombok, Springboot Dev Tool**

As for diagram, we need to create the three-entity user address and product. But here user is only the entity. These are the embedded user entity. Okay, so let's create the entity user. First declare the field private and ID private string name then private something like string gender.



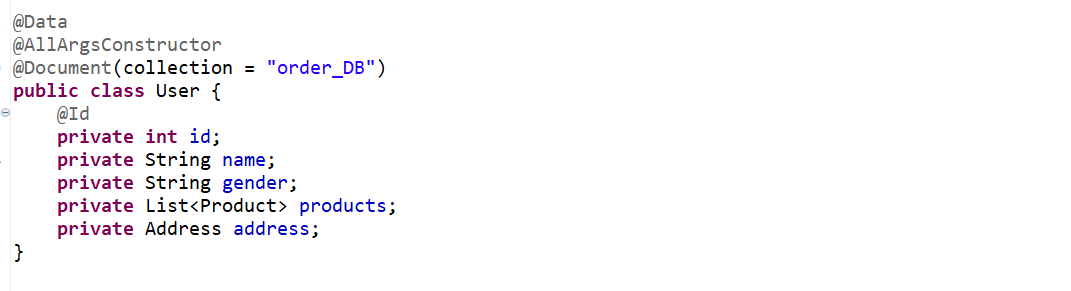
Let’s annotate User as @Document and pass collection as order\_DB.





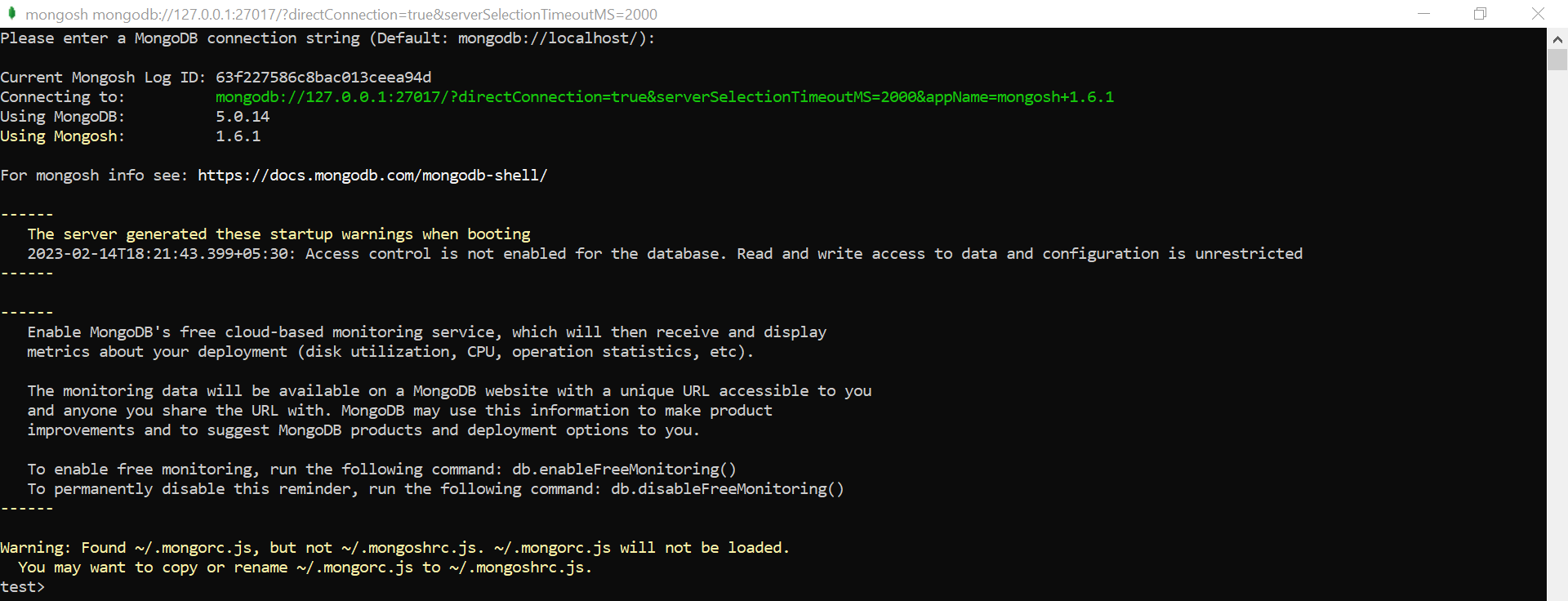
So, a User can buy a List of Products and a User can have one Address.

So, here no need to use mapping annotation like **@One-To-One, @One-To-Many, @Many-To-Many** annotations.

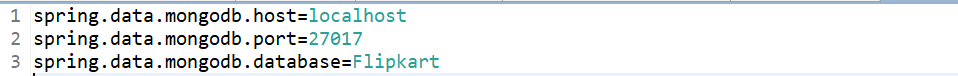


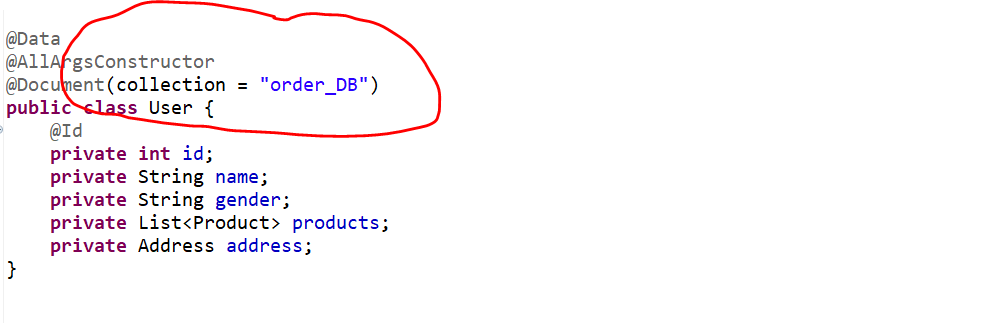
So, now let me create a database name and collection name.

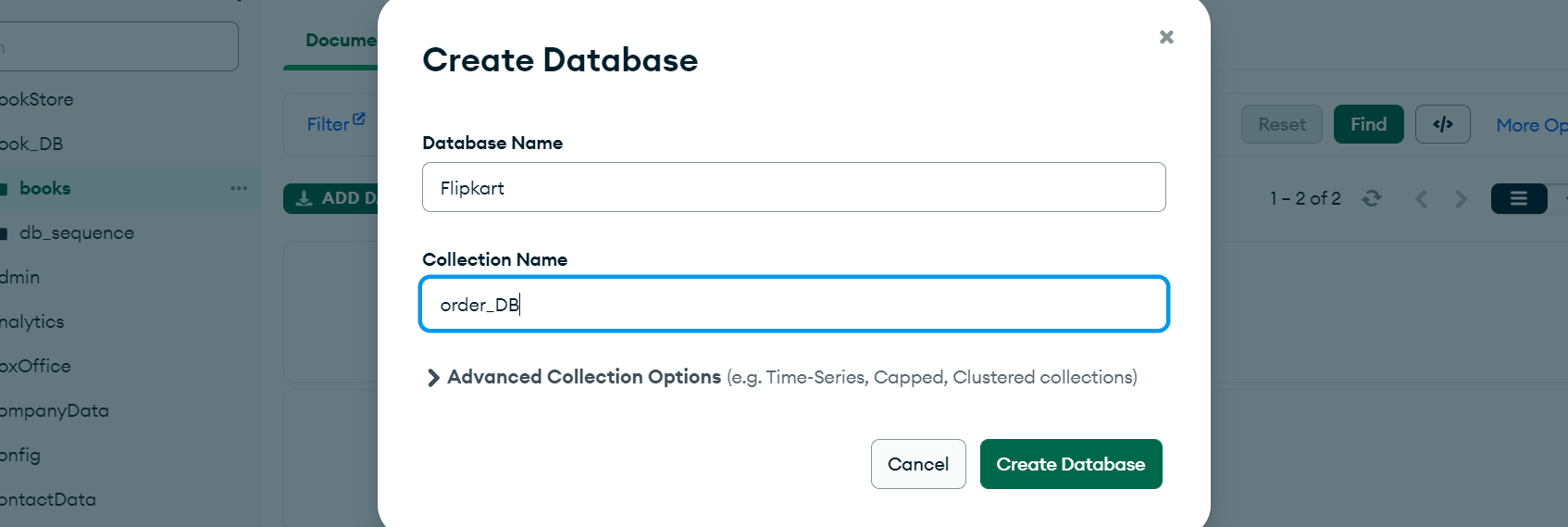
Let’s connect with mongo db server and open the Mongo Compass Community…

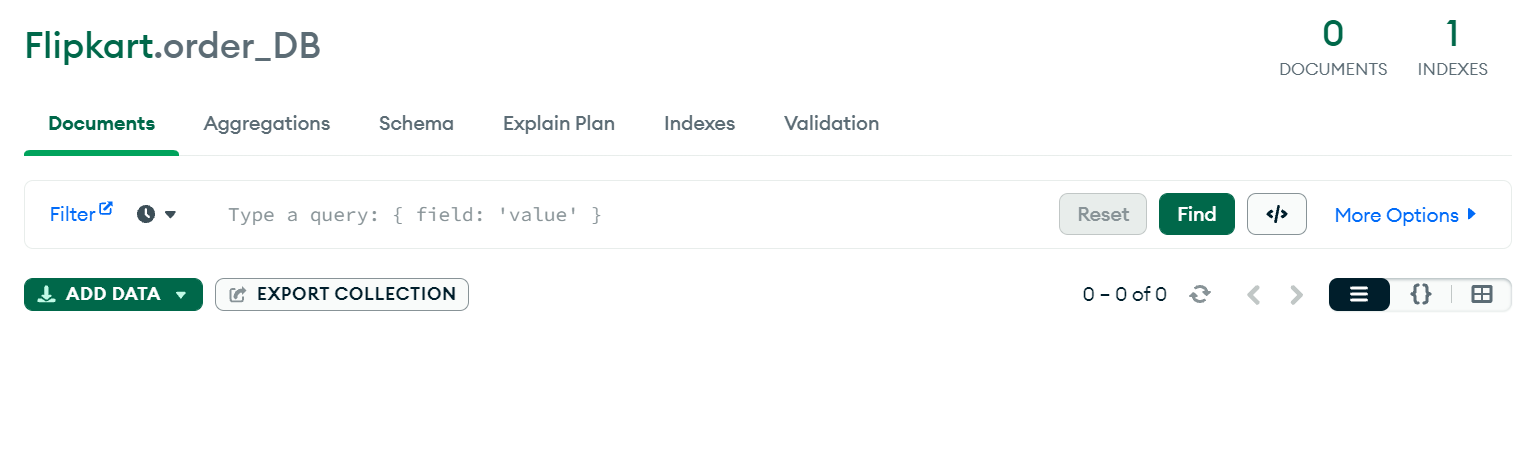


**Application.properties**

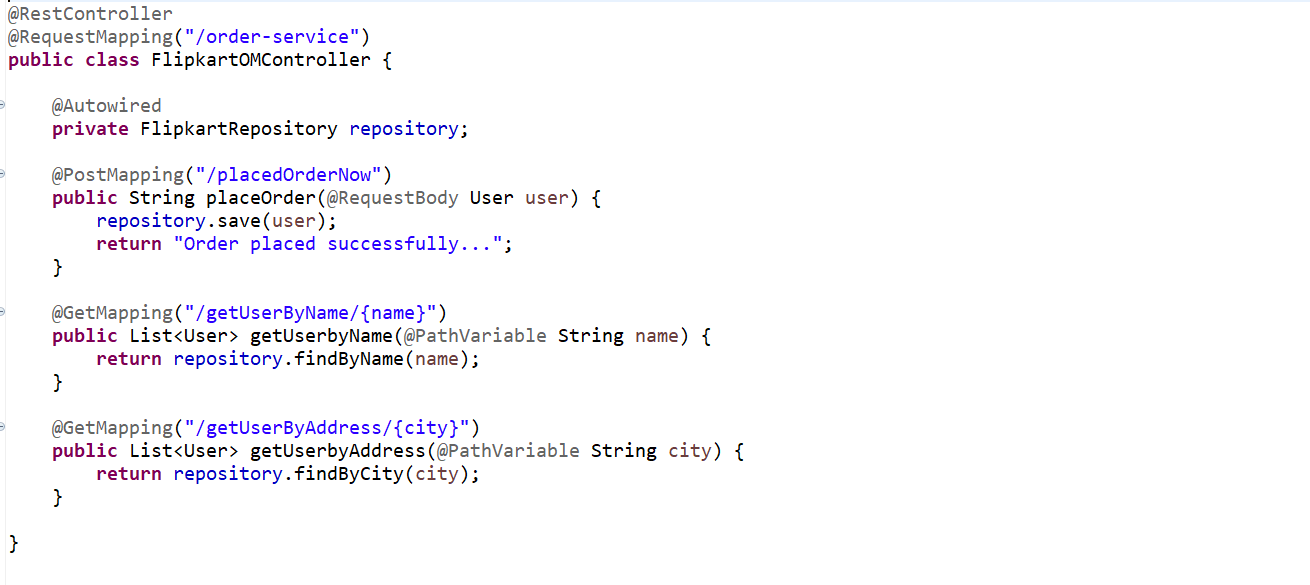




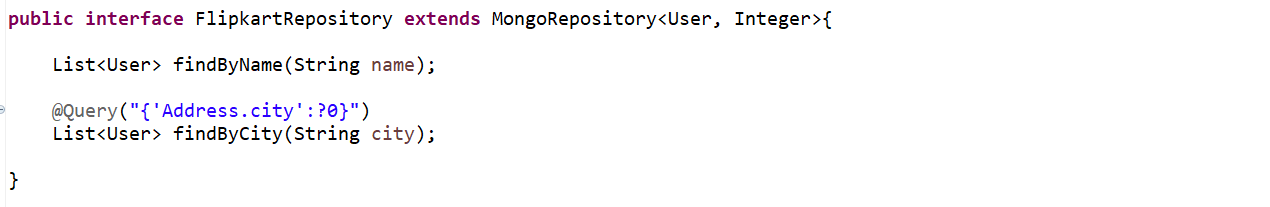




Now let’s create a controller…



Now we want to get the User by Address based on city. In our Address Embedded document there will be one field city.



Now let’s start the application…

Now let’s add some data into our database…

Now let’s add the Body or Request Payload for Post Mapping…first is our User Object having id, name ,gender and then it contains list of products. In products pass the product object just see what all are fields. Similarly add one more product. Now the next one is address.

**POST** [**http://localhost:8080/order-service/placedOrderNow**](http://localhost:8080/order-service/placedOrderNow)

**Request -**

**{**

**"id" : 2878,**

**"name" : "Kaushal",**

**"gender" : "male",**

**"products": [**

**{**

**"name" : "mobile",**

**"quantity" : 1,**

**"price" : 8000**

**},**

**{**

**"name" : "cooler",**

**"quantity" : 1,**

**"price" : 5000**

**}**

**],**

**"address":{**

**"city" : "Bangalore",**

**"state" : "Karnataka",**

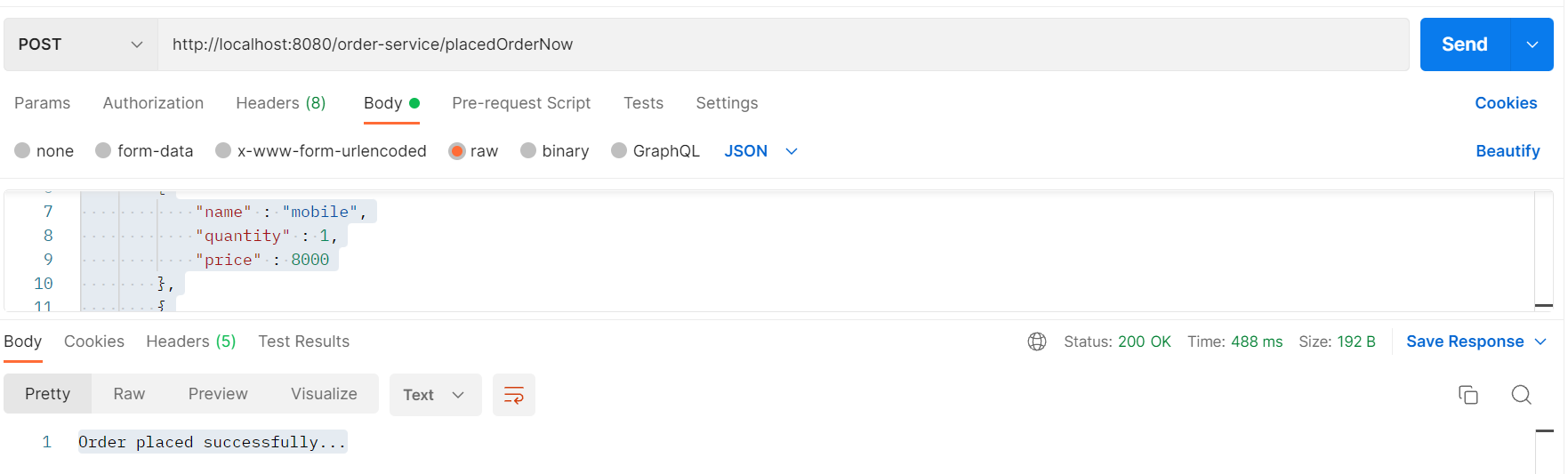
**"pincode" : "560037"**

**}**

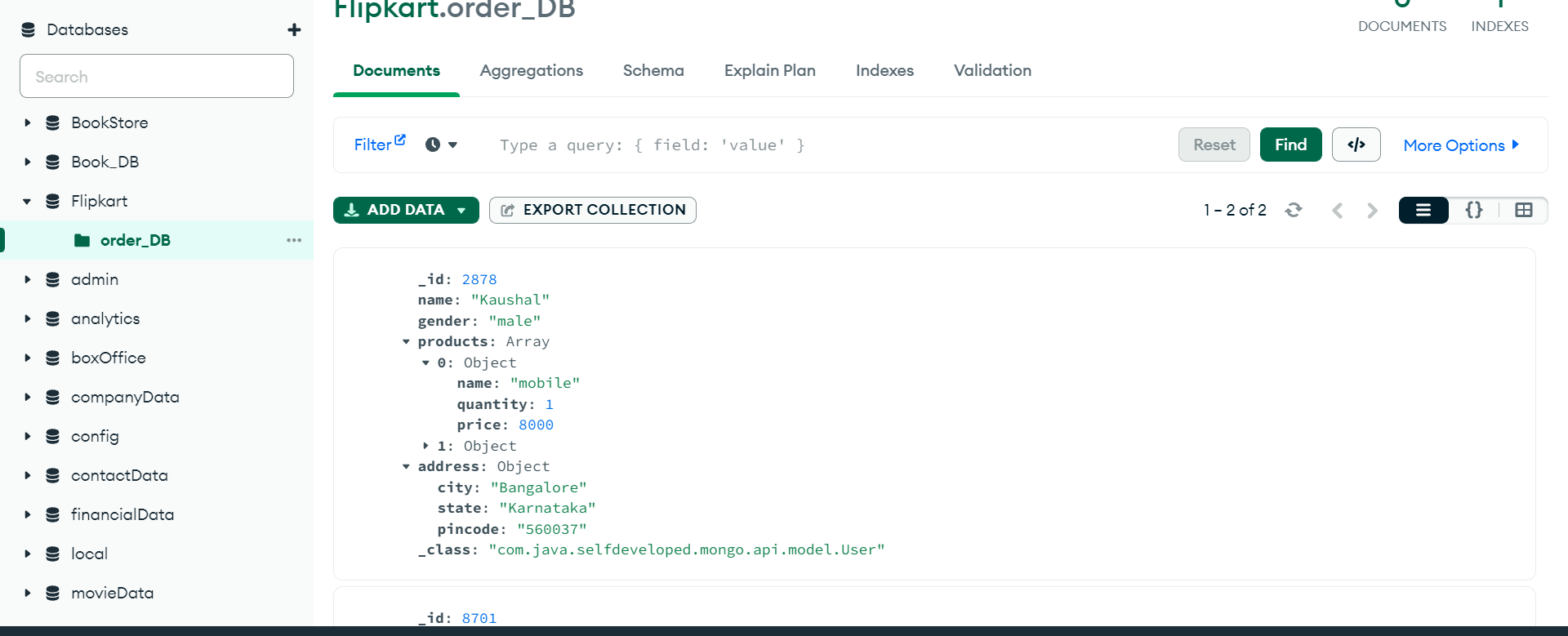
**}**

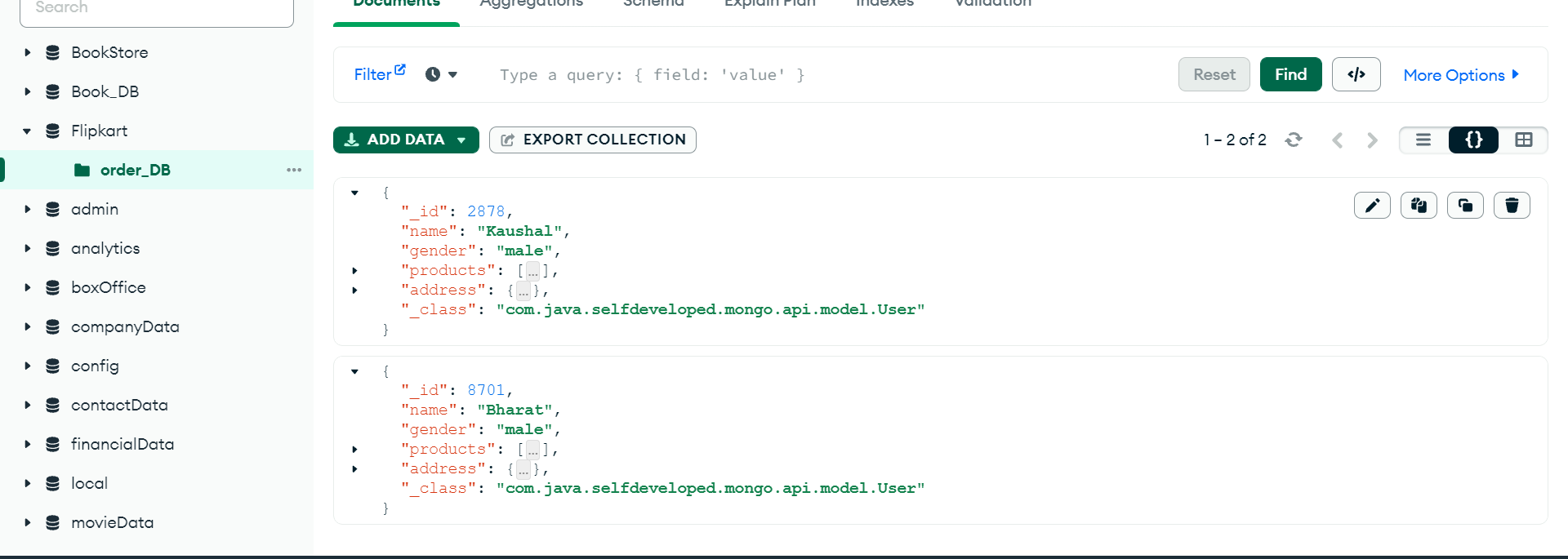
**Response-**

Order placed successfully...



Added one more embedded document…





Let’s get User By name….

**GET** [**http://localhost:8080/order-service/getUserByName/Bharat**](http://localhost:8080/order-service/getUserByName/Bharat)

**[**

**{**

**"id": 8701,**

**"name": "Bharat",**

**"gender": "male",**

**"products": [**

**{**

**"name": "IPhone",**

**"quantity": 1,**

**"price": 80000**

**},**

**{**

**"name": "AC",**

**"quantity": 1,**

**"price": 50000**

**}**

**],**

**"address": {**

**"city": "Bangalore",**

**"state": "Karnataka",**

**"pincode": "560037"**

**}**

**}**

**]**

Let Get user by Address based on city….

**GET** [**http://localhost:8080/order-service/getUserByAddress/Bangalore**](http://localhost:8080/order-service/getUserByAddress/Bangalore)

**[**

**{**

**"id": 2878,**

**"name": "Kaushal",**

**"gender": "male",**

**"products": [**

**{**

**"name": "mobile",**

**"quantity": 1,**

**"price": 8000**

**},**

**{**

**"name": "cooler",**

**"quantity": 1,**

**"price": 5000**

**}**

**],**

**"address": {**

**"city": "Bangalore",**

**"state": "Karnataka",**

**"pincode": "560037"**

**}**

**},**

**{**

**"id": 8701,**

**"name": "Bharat",**

**"gender": "male",**

**"products": [**

**{**

**"name": "IPhone",**

**"quantity": 1,**

**"price": 80000**

**},**

**{**

**"name": "AC",**

**"quantity": 1,**

**"price": 50000**

**}**

**],**

**"address": {**

**"city": "Bangalore",**

**"state": "Karnataka",**

**"pincode": "560037"**

**}**

**}**

**]**