Date: 05/04/2021
Spring Boot 9AM
Mr. RAGHU

Spring Boot Data MongoDB

=> MongoDB is a NoSQL Database.
=> Stores Data in Collections(similer to tables)
=> Collections store data in JSON Format(Document)
=> Queries are called as commands/functions
=> Start MongoDB server 'mongod'
then start mongo client 'mongo'
=> Unstructural Database, no fixed schema given.
=> It supports easy scaling(both-Horizontal and Vertical)
=> MongoDB server runs on port : 27017 (default)

=> Primary Key type in MongoDB is String[must be],
with variable name id, then onlyDB generates
HexaDecimal Value (UUID).

*) If we modify type from String to any other
then MongoDB will not generate value, assign it
manually.

=> @Id (org.springframework.data.annotation)
Must be applied on any one variable.

If we do not provide then default variable
is taken as id with default type String
and mongodb generates Hexa Decimal value.

*) If we provide
```
@Id
private String id;
```
we can get generate ID back to Application,
else we can not see Id.

*) we can choose non-String type for primary key
variable creation, then
a. This time id is not generated
b. variable name(bookId) is mapped with _id key in DB

```
@Id
private Integer bookId;
```

=> @Document annotation is optional. To provide
collection details we can use it. If we do not
provide collection name, then defualt is
className (with lowercase).

=> TODO all Database operations in MongoDB,
Spring Data API has provided one interface
'MongoRepository<T,ID>'

T = Model class Name
ID = PrimaryKey DataType (ex: String)

```
  This interface internally extends two more
  CrudRepository and PagingAndSortingRepository

=> Do not use JPA Annotations and concepts
  (@Entity and show_sql, ddl-auto,
   dialect..etc)  It is NoSQL Database.

=================================================
  App#1 : Spring Boot MongoDB Crud App
  App#2 : Spring Boot MongoDB Security

-------------------------------------------------
  App#1 : Spring Boot MongoDB Crud App

*) If we add 'spring-boot-starter-data-mongodb'
    (Spring Data MongoDB) dependency then
   our project behaves as client(MongoDB Client)

=> By default it create connection with details
   default port = 27017, host=localhost, database=test


S#1 Create one Spring Starter Project
 Name: SpringBoot2MongoDBCurdEx
 Dep : Spring Data MongoDB, Lombok

S#2 application.properties
spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=nit
#spring.data.mongodb.username=
#spring.data.mongodb.password=

S#3 Model class
package in.nareshit.raghu.model;

import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.NonNull;
import lombok.RequiredArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
@RequiredArgsConstructor
//@Document(collection = "sample") //optional
public class Book {

        //auto-generated
        @Id
        private String id;
```

```
	@NonNull
	private Integer bookId;
	@NonNull
	private String bookName;
	@NonNull
	private Double bookCost;

}
```

S#4 Repository Interface

```
package in.nareshit.raghu.repo;
import org.springframework.data.mongodb.repository.MongoRepository;
import in.nareshit.raghu.model.Book;
public interface BookRepository
	extends MongoRepository<Book, String> {

}
```

S#5 Runner class

```
package in.nareshit.raghu.runner;

import java.util.Arrays;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;

import in.nareshit.raghu.model.Book;
import in.nareshit.raghu.repo.BookRepository;

@Component
public class BookTestRunner implements CommandLineRunner {
	@Autowired
	private BookRepository repo;

	public void run(String... args) throws Exception {
		//for ddl-auto=create
		repo.deleteAll();

		//1. save() - insert/update
		repo.save(new Book(2505, "Core Java", 500.0));
		repo.save(new Book(2506, "Adv Java", 600.0));
		Book b = repo.save(new Book(2507, "SpringBoot",
800.0));

		System.out.println(b.getId());

		repo.saveAll(
				Arrays.asList(
						new Book(2508,
"Angular", 600.0),
						new Book(2509, "MS",
800.0),
						new Book(2510, "HTML",
900.0)
						)
				);
```

```
                System.out.println("-----------DONE-------");

        }

}

cmd> mongod
cmd> mongo

> use nit;
switched to db nit
> show collections
book
> db.book.find().pretty();
==================================================
Q) Which datatype can be used to create Primarykey
   variable in Model class?
A) String (even any other Primitive Type
    Integer, Char, Boolean).

--------------------------------------------------
Q) How collection is created in MongoDB if there is
   no ddl-auto?
A) When we perform insert function/commands, then
   Database creates a collection first if not exist
   next insert data.


MongoDB Manual Link:
https://docs.mongodb.com/v3.6/tutorial/insert-documents/

1. insert/insertOne/insertMany

cmd> mongod
cmd> mongo
   > show databases;
   > use sample
   > show collections

# To insert one JSON Object
> db.student.insertOne({"sid":201,"sname":"SYED","sfee":300.2});
{
        "acknowledged" : true,
        "insertedId" : ObjectId("606a8f58c0109a43a3084785")
}

# To insert multiple JSON Objects
> db.student.insertMany([
  {"sid":202,"sname":"SAM","sfee":400.2},
  {"sid":203,"sname":"RAM","sfee":600.2},
  {"sid":204,"sname":"AJAY","sfee":800.2}
]);

*) insert can be used as either insertOne/insertMany

db.student.insert({"sid":205,"sname":"AA","sfee":300.2});
```

```
db.student.insert([
  {"sid":206,"sname":"BB","sfee":400.2},
  {"sid":207,"sname":"CC","sfee":600.2},
  {"sid":208,"sname":"DD","sfee":800.2}
]);
```

db.collection.insertOne() Inserts a single document into a collection.
db.collection.insertMany() Inserts multiple documents into a collection.
db.collection.insert() Inserts a single document or multiple documents into a collection.

*)Note: While running Application if we get
MongoSocketOpenException: Exception opening socket
Caused by: java.net.ConnectException: Connection refused: connect
then start mongodb server (mongod)

==========

Ex#1 (valid or not?)

```
class Person {
   @Id
   Integer pid;
   String pname
}
```

*) Valid, but id not generated. We need to provide

Ex#2 (valid or not?)

```
class Person {
   Integer pid;
   String pname
}
```

*) Valid, auto-generated id with String(hexa decimal)

Ex#3 (valid or not?)

```
class Person {
   @Id
   String id;
   Integer pid;
   String pname
}
```

*) Valid, we can even see generated id at Application.

*) we can define manual code for customization of id

```
package in.nareshit.raghu;
import java.util.Random;
public interface MyIdGen {

        public static int getId() {
                return new Random().nextInt(999999);
        }
}
```

==========