

Date : 09/04/2021

Spring Boot 9AM

Mr. RAGHU

MongoDB - Spring Boot

Primitive Data: -

Variable in class --- key that holds
single value in Document

Collection Data:-

List/Set/Array

variable in class - key that holds multiple values

key : [val, val, val,]

Map:-

```
key : {  
    mapKey : mapVal,  
    mapKey : mapVal  
}
```

Association Mapping:- ClassType

In this case on inner JSON Document is created.

That looks format:

hasAVariable behaves like a key and child object
behaves as like Inner JSON.

```
hasAVariable : {  
    variable : value  
}
```

*) INNER JSON also called as Embedded Document.

Finally one JSON Document that contains all child
JSONS is created. They are not re-usable, repeated
data.

Employee---<>Address

eid	aid
ename	lod
addr	

```
{  
  "eid"    : 10,  
  "ename"  : "SAM",  
  "addr"   : {  
    "aid"   : 100,  
    "loc"   : "HYD"  
  }  
}
```

Download Link:

<https://robomongo.org/>

```
> Click on Download Studio 3T Only
> Extract Zip to Folder > Double click to install
> next > next > Finish
> Start Robo Studio 3T > Click on Connect > new Connection
> Enter name > save > Click on Connect Button
> Expand Database and choose collection > press enter
> use Query Builder tool for easy search
```

====Example=====

1. models

```
package in.nareshit.raghu.model;
```

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
```

```
@Data
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
public class Department {
```

```
    private String deptCode;
```

```
    private String deptName;
```

```
}
```

```
--
```

```
package in.nareshit.raghu.model;
```

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
```

```
@Data
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
public class Address {
```

```
    private String hno;
```

```
    private String loc;
```

```
    private Long pinCode;
```

```
}
```

```
--
```

```
package in.nareshit.raghu.model;
```

```
import java.util.List;
import java.util.Map;
import java.util.Set;
```

```
//ctrl+shift+O
```

```
import org.springframework.data.annotation.Id;
```

```
import org.springframework.data.mongodb.core.mapping.Document;
```

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
```

```

@Data
@Document
@NoArgsConstructor
@AllArgsConstructor
public class Employee {

    @Id
    private Integer empId;
    private String empName;
    private Double empSal;

    private Set<String> empPrjs;
    private List<String> empPrjVer;
    private String[] empGrades;

    private Map<String,String> empClient;

    private Address addr;//HAS-A
    private List<Department> dobs; //HAS-A
}

```

```

}
-----
2. Repository Inteface
package in.nareshit.raghu.repo;

import org.springframework.data.mongodb.repository.MongoRepository;

import in.nareshit.raghu.model.Employee;

public interface EmployeeRepository
    extends MongoRepository<Employee, Integer> {

}

```

```

-----
3. Runner class
package in.nareshit.raghu.runner;

import java.util.List;
import java.util.Map;
import java.util.Set;

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

import in.nareshit.raghu.model.Address;
import in.nareshit.raghu.model.Department;
import in.nareshit.raghu.model.Employee;
import in.nareshit.raghu.repo.EmployeeRepository;

```

```

@Component
public class DataInsertRunner implements CommandLineRunner {

    @Autowired
    private EmployeeRepository repo;
}

```

```

        public void run(String... args) throws Exception {
            repo.deleteAll();
            repo.save(
                new Employee(
                    10, "SAM", 200.0,
                    Set.of("HTC", "NIT", "ORCL"),
                    List.of("3.2GA", "6.5
RELEASE", "0.1 ALPHA"),
                    new String[]
{"A+", "GR-T", "UI-NEW"},
                    Map.of("C1", "TEC-
N", "C2", "US-ARMY", "C3", "JANSON & JANSON"),
                    new Address("8-9/A",
"HYD", 500032L),
                    List.of(
                        new Department("D1", "DEV-AB"),
                        new Department("D2", "QA-RB"),
                        new Department("D3", "SUPRT-MN")
                    )
                )
            );
        }
    }
}

```

```

-----
4. application.properties
spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=nit

```

5. Output:

```

{
  "_id" : 10,
  "empName" : "SAM",
  "empSal" : 200,
  "empPrjs" : [
    "ORCL",
    "NIT",
    "HTC"
  ],
  "empPrjVer" : [
    "3.2GA",
    "6.5 RELEASE",
    "0.1 ALPHA"
  ],
  "empGrades" : [
    "A+",
    "GR-T",
    "UI-NEW"
  ],
  "empClient" : {

```

```
        "C3" : "JANSON & JANSON",
        "C2" : "US-ARMY",
        "C1" : "TEC-N"
    },
    "addr" : {
        "hno" : "8-9/A",
        "loc" : "HYD",
        "pinCode" : NumberLong(500032)
    },
    "dobs" : [
        {
            "deptCode" : "D1",
            "deptName" : "DEV-AB"
        },
        {
            "deptCode" : "D2",
            "deptName" : "QA-RB"
        },
        {
            "deptCode" : "D3",
            "deptName" : "SUPRT-MN"
        }
    ],
    "_class" : "in.nareshit.raghu.model.Employee"
}
```