Spring Framework: Spring Data MongoDB

CONNECTING TO MONGODB

Dan Geabunea
SENIOR SOFTWARE DEVELOPER

@romaniancoder www.romaniancoder.com

Overview

- Set up the development environment
- Basic Mongo concepts
- The dependencies needed for working with MongoDB in Spring
- Mongo data annotations & connection properties
- Demo: Create a Spring application and connect it to Mongo DB

Course Pre-requisites

Comfortable writing Java applications

Basic experience using Spring and Spring Boot

Some familiarity with MongoDB

Get up and Running

Prepare development environment

Create an empty Spring Boot application

Add the Spring Data MongoDB starter dependency

Define the connection properties

Apply Mongo data annotations to control persistence

The Development Environment

Java 8+

Mongo DB Server

IDE

Mongo GUI

Mongo Compass

https://www.mongodb.com/products/compass

Robo 3T

https://robomongo.org/download

Basic Mongo DB Concepts

Relational Database

Id	Name	Age
1	Dan	31
2	Anna	45
3	John	23

Table

Rows

Columns

Primary Keys

Mongo BSON Documents

```
{
    “_id”: “343fgrregrgregr”,
    “name”: “Dan”,
    “age”: 34
},
{
    “_id”:”ffdjwqiu83gndsoue,
    “name”: “Anna”,
    “age”: 24
}
```

Relational vs. Mongo

Relational	Mongo
Database	Database
Tables	Collections
Rows	Documents
Columns	Fields
Primary key	Id

Spring Mongo Dependencies

```
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter</artifactId>
    </dependency>
</dependencies>
```

Start with an Empty Spring Boot Application

This will bring in the basic default Spring Boot dependencies

```
<dependencies>
    …
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-mongodb</artifactId>
    </dependency>
</dependencies>
```

Add the Spring Data Mongo Dependency

Update the pom.xml file

Spring Boot Starter Mongo DB

spring-boot-starter-data-mongodb

- mongodb-driver
  - mongodb-driver:bson
  - mongodb-driver:driver-core
- spring-data-mongodb
  - spring-data-commons

Spring Mongo Data Annotations

Mongo Data Annotations

Metadata that we add on our classes and fields to let Spring Data know how to handle persistence in a Mongo database

Most Used Annotations

@Document

@Id

@Field

@Transient

Most Used Annotations

@Indexed

@TextIndexed

@CompoundIndex

@DbRef

@Document

```
@Document
public class Aircraft {
    private String id;
    private String model;
    private int nbSeats;
    …
}
```

@Document

```
@Document(collection="airplanes")
public class Aircraft {
    private String id;
    private String model;
    private int nbSeats;
    …
}
```

@Id

```
@Document
public class Aircraft {
    @Id private String id;
    private String model;
    private int nbSeats;
    …
}
```

@Field

```
@Document
public class Aircraft {
    @Id private String id;
    private String model;
    @Field private int nbSeats;
    …
}
```

@Field

```
@Document
public class Aircraft {
    @Id private String id;
    private String model;
    @Field(name="seats") private int nbSeats;
    …
}
```

@Transient

```
@Document
public class Aircraft {
    @Id private String id;
    private String model;
    @Transient private int nbSeats;
    …
}
```

@Indexed

```
@Document
public class Aircraft {
    @Id private String id;
    @Indexed private String model;
    private int nbSeats;
    …
}
```

@Indexed

```
@Document
public class Aircraft {
    @Id private String id;
    @Indexed(direction=IndexDirection.ASCENDING, unique=false)
    private String model;
    private int nbSeats;
    …
}
```

@TextIndexed

```
@Document
public class Aircraft {
    @Id private String id;
    @TextIndexed
    private String model;
    private int nbSeats;
    …
}
```

@CompoundIndex

```
@Document
@CompoundIndex(def="{'model':1, 'nbSeats':-1}")
public class Aircraft {
    @Id private String id;
    private String model;
    private int nbSeats;
    …
}
```

@DbRef

Aircraft

```
@Document
public class Aircraft {
    @Id private String id;
    private String model;
    @DbRef
    private Manufacturer man;
}
```

Manufacturer

```
@Document
public class Manufacturer{
    @Id private String id;
    private String name;
    …
}
```

Mongo Connection Properties

Basic Connection Parameters

Mongo server IP address

Mongo server port

Database name

Credentials [optional]

Via Properties File

```
spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=airportmanagement
spring.data.mongodb.username=admin
spring.data.mongodb.password==admin1234
```

Via Properties File

```
spring.data.mongodb.uri=mongodb://amin:pass123@localhost:27017/airportdb
```

At Runtime

```
java -jar ./my-app.jar
--spring.data.mongodb.uri=mongodb://localhost:27017/airportdb
```

Demo

Connect a Spring application to a Mongo database

- Import needed dependencies
- Add connection parameters
- Apply annotations

Summary

Recapped some fundamental Mongo terminology

Learned the 3 steps to get started with Spring and MongoDB

- Configure dependencies
- How to efficiently annotations
- Configure connection properties

We have a simple but functional application