Spring Boot集成MongoDB Demo

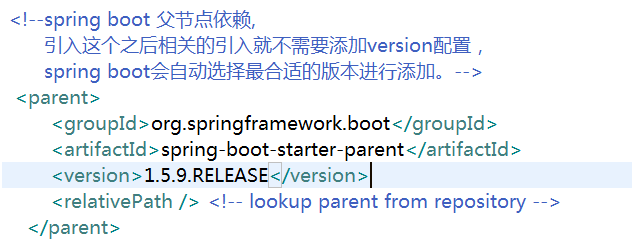
(1) 准备工作；

安装mongodb 服务,版本为mongoDB3.2 64位，

下载安装完，把mongodb服务启动起来以备之后进行操作使用。

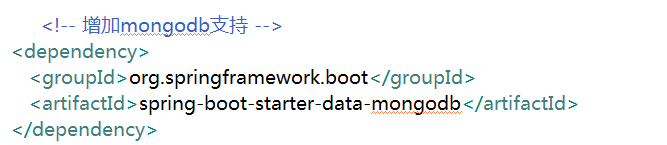
(2) 新建一个maven java project:

在这里我们新建一个spring-boot-mongodb项目，spring boot版本为1.5.9.RELEASE。



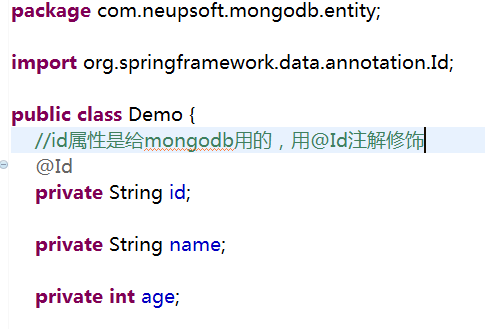
(3) 引入相关依赖；

引入mongodb的依赖，如下pom.xml文件：



(4) 编写测试代码；

在这里我们需要编写一个简单的例子进行测试下，我们需要创建一个Demo实体类，需要创建一个操作MongoDB的repository类，再者就是一个控制类DemoController。具体看如下代码：

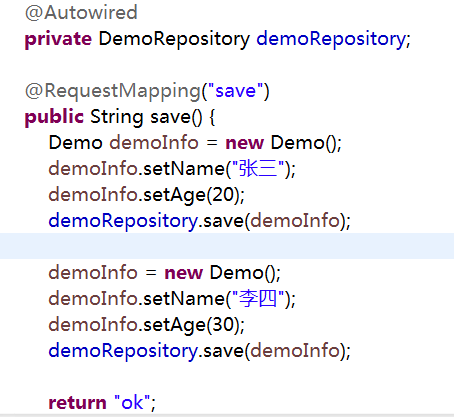


接下来编写一个操作mongodb的repository代码，它继承MongoRepository接口；MongoRepository接口包含了常用的CRUD操作，例如：save,insert,fillAll等。我们也可以定义我们自己的操作接口，具体看如下代码：

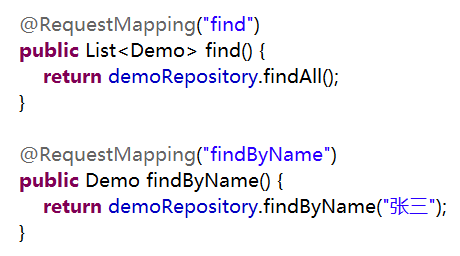


接下来就是访问控制类，这个类中很简单，都是最基本的代码，看如下：

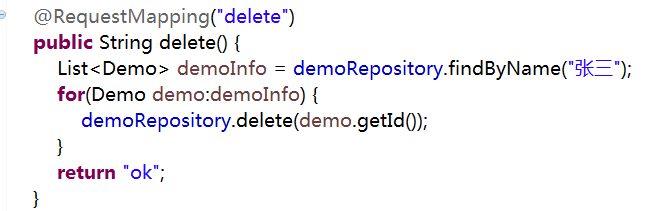
新增：



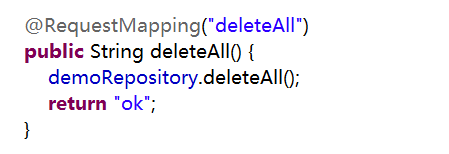
查询：



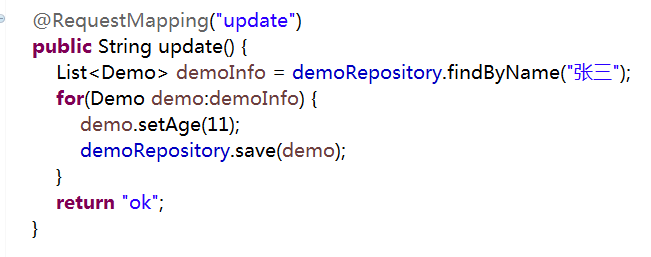
删除：



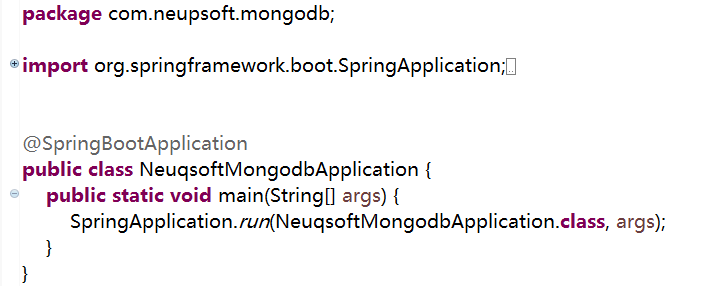
删除全部：



修改：



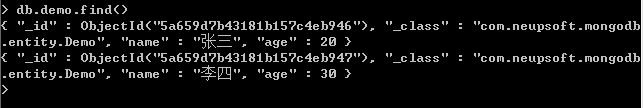
最后编写App.java启动类，启动应用程序：



运行程序，依次访问如下地址：

http://localhost:8080/save ：保存两条数据；

查询数据库中信息已经存入：



对应关系：

实体类名：集合名

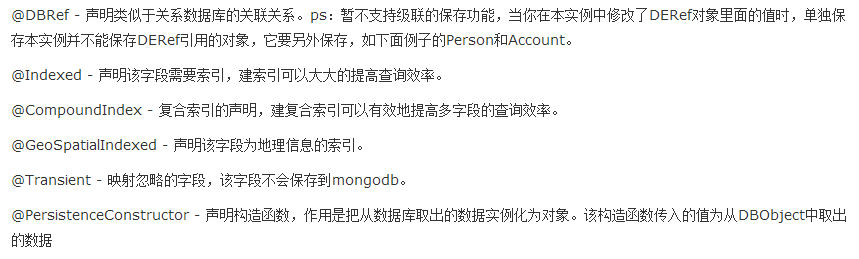
属性名：数据库中字段

如果想改变存储在mongodb中对应的集合名称，可以通过注解@Document(collection="mongodb对应集合名") 来设置。

同样，我们也可以通过@Field(values="mongodb对应字段名")来改变数据库中对应的字段名称。



Mongodb中其他常见注解：



http://localhost:8080/find ：查询所有数据；



http://localhost:8080/findByName ：通过名称查找；



http://localhost:8080/update ：修改数据；

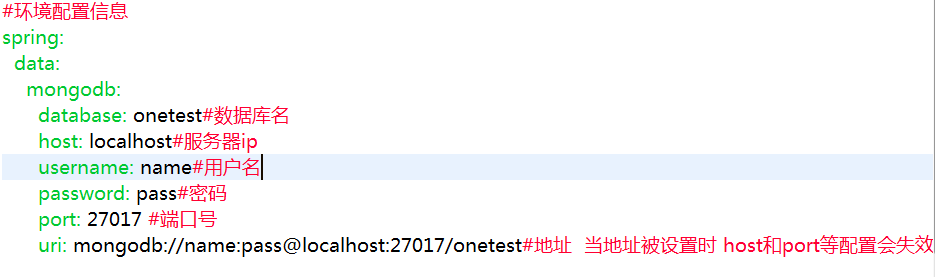
http://localhost:8080/delete ：删除数据；

http://localhost:8080/deleteAll ：删除所有数据；

只要在浏览器中能看到以上数据，基本的使用算是成功了。

(5) 配置文件

我们会发现我们根本没有做什么配置，因为spring boot提供了默认的一些配置，默认将数据存到test数据库中，集合名称为实体类名，默认端口号为27017。Spring boot 也给我们配置的地方application.yml文件，常见的配置如下



（6）springBoot怎么连接多个副本集模式的mongodb

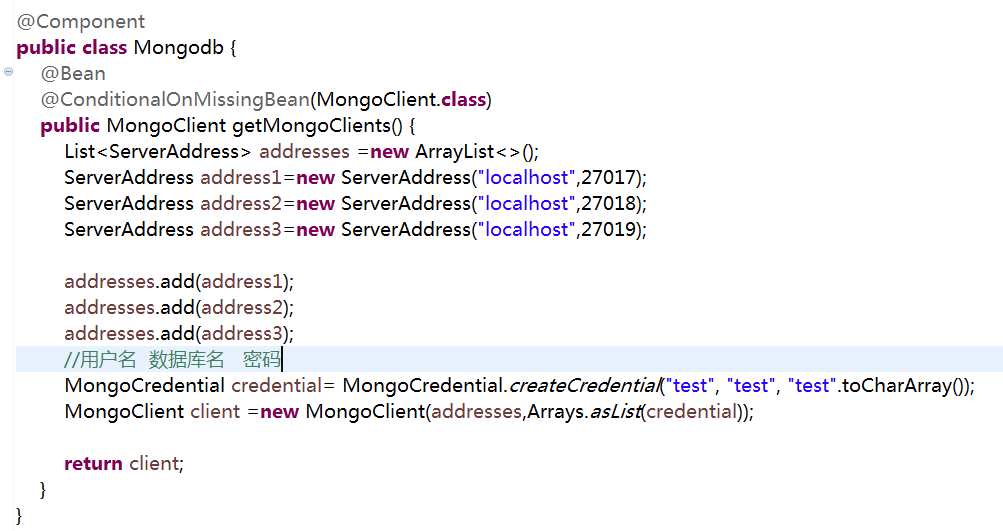
首先先创建一个mongodb副本集

mongod --port 27017 --dbpath d:/mongodb/data --replSet "rs0"

mongod --port 27018 --dbpath d:/mongodb/data2 --replSet "rs0"

mongod --port 27019 --dbpath d:/mongodb/data3 --replSet "rs0"

可以采用覆盖spring boot bean的方法覆盖掉系统默认的mongodb client的处理方法，由于前面spring boot的配置中仅仅能配置一个ip地址， 因此对上面的多地址的客户端连接不支持， 因此spring boot的默认mongodb方法不能满足需求



@Bean 这个声明了一个bean

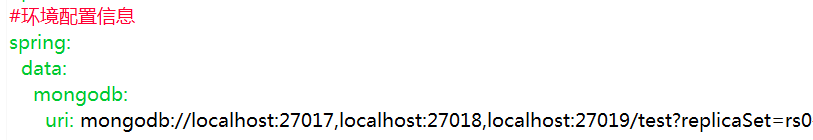
@ConditionalOnMissingBean(MongoClient.class) 这个非常重要，

这个是告诉系统，下面的bean是要替换系统默认的那个bean的（mongodb）

最后@Component这个是告诉spring boot 要扫描着类， 否则，前面的注解都没用。

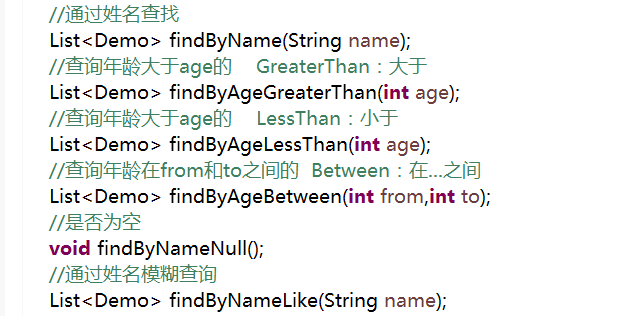
还有一种简单的方法：

直接通过配置文件中的spring.data.mongodb.uri配置多个地址来实现副本集的链接，如下：



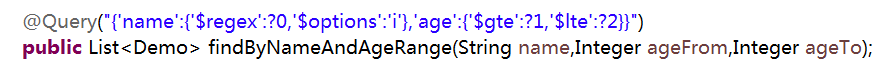
mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][replicaSet=副本集名]]

我们还可以在MongoRepository接口中做一些复杂的查询功能。以按照以下规则定义接口的方法，自定义查询方法，格式为“findBy+字段名+方法后缀”，方法传进的参数即字段的值



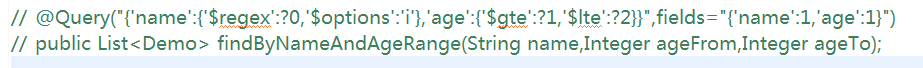
尽管以上查询功能已经很丰富，但如果还不能满足使用情况的话可以用一下方法---基于mongodb原本查询语句的查询方式。

例：在原接口中加入



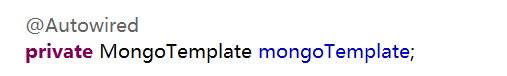
注释Query里面的就是mongodb原来的查询语法，我们可以定义传进来的查询参数，通过坐标定义方法的参数。

还可以在后面指定要返回的数据字段，如上面的例子修改如下，则只通过demo表里面的name和age字段构建demo对象。



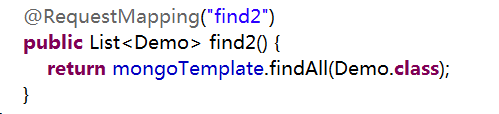
(7) 使用MongoTemplate进行操作;

首先在DemoController中引入MongoTemplate ：

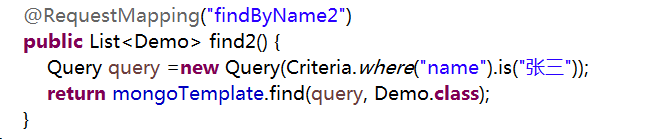


编写访问方法：

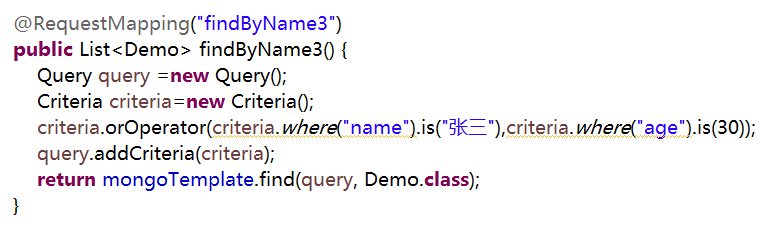
查询全部：



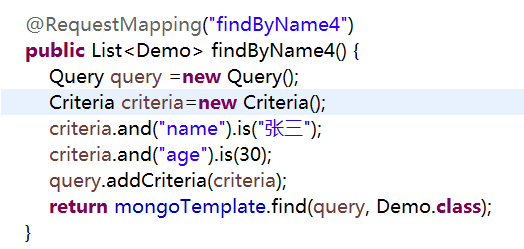
按姓名查找：



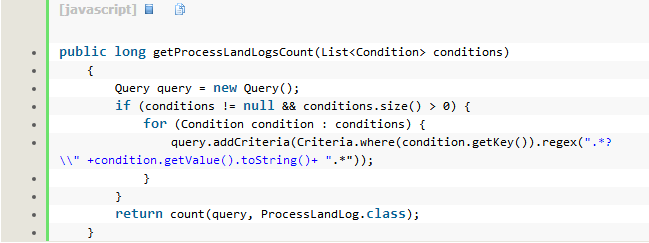
或者条件：



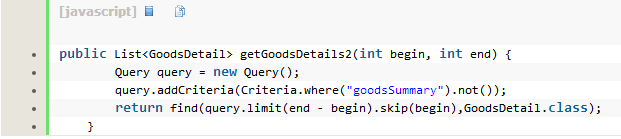
并且条件：



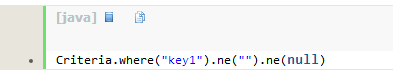
模糊查询：



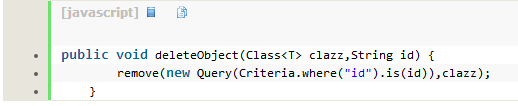
查询字段不存在的数据：



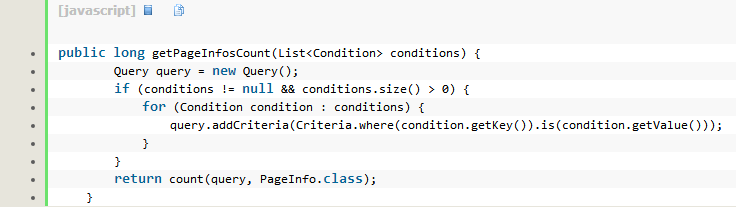
查询字段不为空的数据：



删除数据：



查询数量：



更新一条数据的一个字段：

1. **public** WriteResult updateTime(PageUrl pageUrl) {
2. String id = pageUrl.getId();
3. **return** updateFirst(**new** Query(Criteria.where("id").is(id)),Update.update("updateTime", pageUrl.getUpdateTime()), PageUrl.**class**);
4. }