Spring-data-mongodb使用说明

## MongoDB安装部署

本文档使用MongoDB 4.0.5，Spring Boot 1.5.13，spring-data-mongodb 1.10.11 版本。

注：Spring boot 1.5.13如果不指定spring-data-mongodb版本时，默认会导入1.10.12版本，该版本会导致使用MongoTemplate时出现异常，所以需要降版本，最高支持到1.10.11。

此处是在docker容器中安装部署。

### 下载MongoDB

创建mongo目录

mkdir -p ~/mongo

进入mongo目录

cd ~/mongo

下载MongoDB

docker pull mongo

### 运行MongoDB

docker run --name mongodb-server -p 27017:27017 -v $PWD/db:/data/db -d mongo:latest –auth

命令说明：

-p 27017:27017：将容器的27017端口映射到主机的27017端口

-v $PWD/db:/data/db：将当前目录下db挂在到容器的/data/db，作为mongo的数据库目录

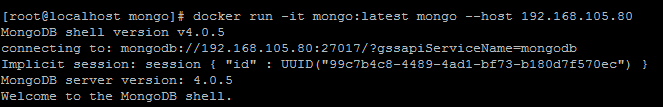
–auth：表示连接mongo需要授权

### 连接mongo容器

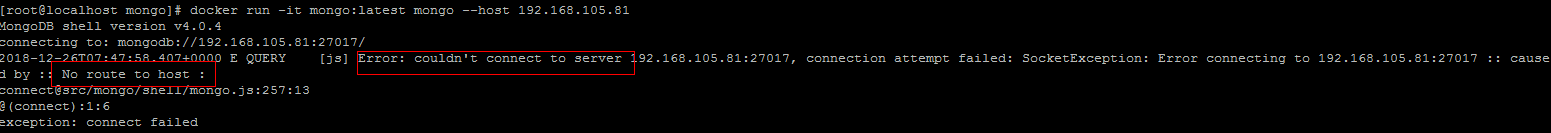
docker run -it mongo:latest mongo --host 192.168.105.80

IP为当前安装MongoDB的主机IP地址。

连接成功会返回如下信息：



如果出现异常如下图所示：



关掉防火墙，或放行27017端口即可。

### 进入mongo容器

docker exec -it mongodb-server /bin/bash

### 进入MongoDB

mongo

成功进入MongoDB会出现如下信息：



### 设置权限添加用户

使用MongoDB内置数据库admin

use admin



添加一个管理员账号用户名为root，密码为123456

db.createUser(

{

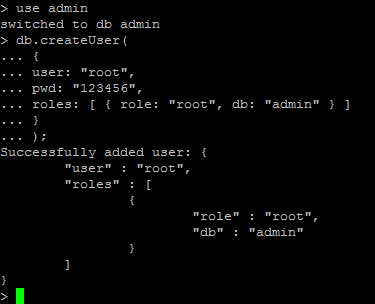
user: "root",

pwd: "123456",

roles: [ { role: "root", db: "admin" } ]

}

);



重新进入admin库，并进行root账号权限验证

db.auth("root","123456");



返回1则表示鉴权成功。

使用用户自定义业务库demo（此时并未创建该库，命令执行不受影响）

use demo

添加一个普通用户用户名为demo，密码为123456，只具备操作该demo库的读写权限

db.createUser(

{

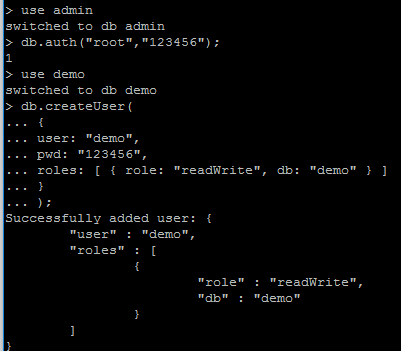
user: "demo",

pwd: "123456",

roles: [ { role: "readWrite", db: "demo" } ]

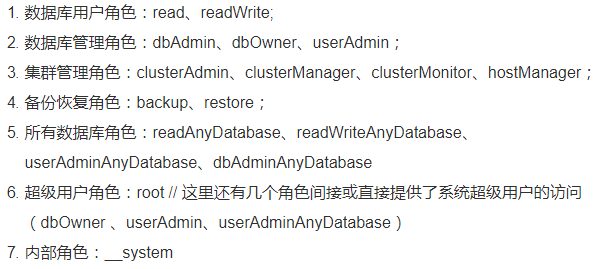
}

);



## MongoDB内置角色权限说明

### 内置的角色



### 角色的权限



## Spring-data-mongodb开发详解

MongoDB概念解释：

数据库（database）同传统关系数据库一致

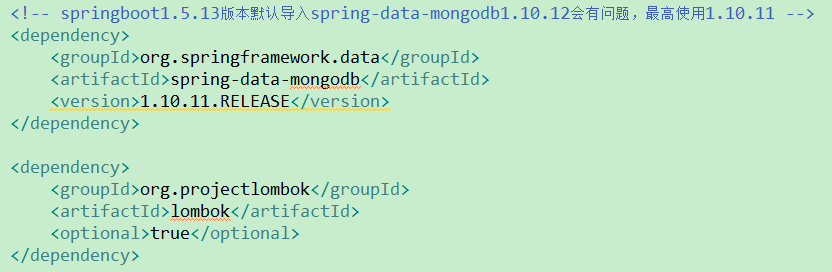
集合（collection）等同于数据库表

文档（document）等同于表内一条记录

使用MongoDB，同传统关系数据库一样，也是建库建集合，再做相应的文档增删改查。

### pom.xml依赖

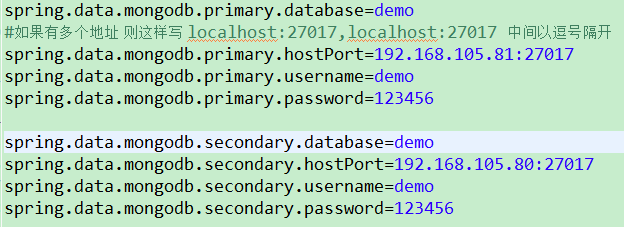
在pom.xml加入spring-data-mongodb依赖，如下图



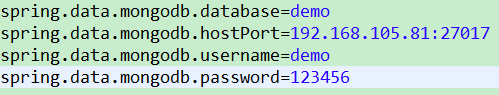
注：此处使用的是springboot1.5.13，如果不指定spring-data-mongodb版本，默认会导入1.10.12，该版本在使用某些接口时会出现异常，降低到1.10.11版本则无问题。

### application.properties配置

在application.properties文件中增加mongo配置，这里实现了连接多数据源，如下图：



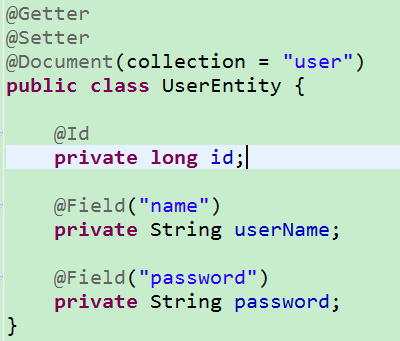
单数据源配置，如下图：



database数据库如果未创建，会自动创建。

### 创建数据实体类

创建数据实体类，会在程序对该实体进行数据操作时，自动在MongoDB数据库中创建对应的集合，如下图：



此处创建一个UserEntity的实体类，会在数据库生成名为user的集合。

@Document注解表示这是一个文档，collection表示集合的名称，不指定该属性默认集合名称为实体类名

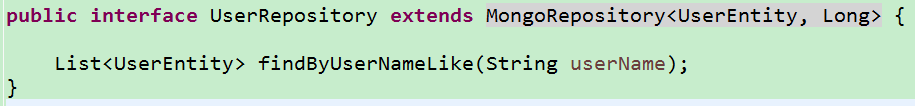
@Id注解为主键标识，无特殊需求，建议不使用该注解指定主键，因为不指定主键时，MongoDB会自动生成\_id主键字段，性能上更好

@Field注解为数据库中集合的字段名称，不使用该注解时，默认字段名为属性名。建议使用该注解自定义字段名，同时名字尽量短，可以节省存储空间

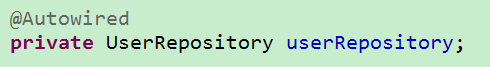
## 单数据源CRUD

### 继承MongoRepository

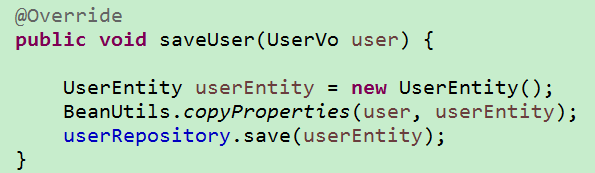
直接继承MongoRepository，默认已实现基本的CRUD方法，如下图：



具体使用，如下图：



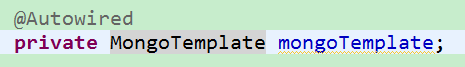
添加数据，如下图：



其它默认方法，delete，findAll等不一一介绍，使用方式跟JPA一致。

### 使用MongoTemplate

直接注入MongoTemplate，如下图：



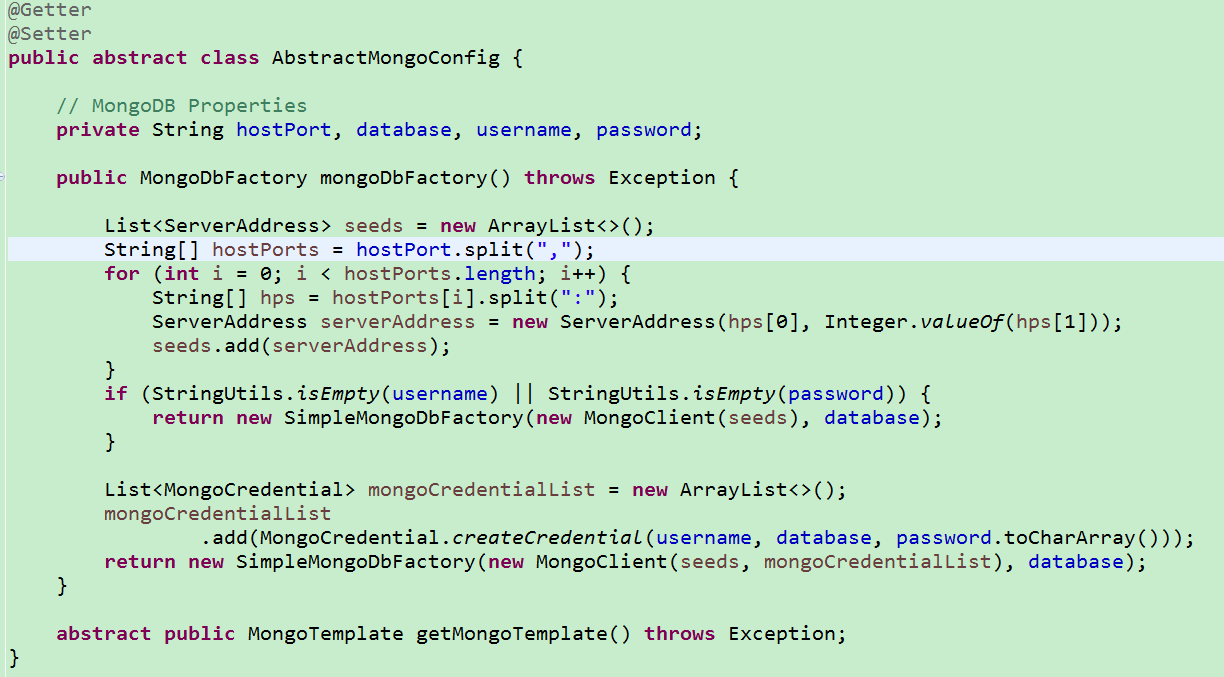
添加数据，如下图：



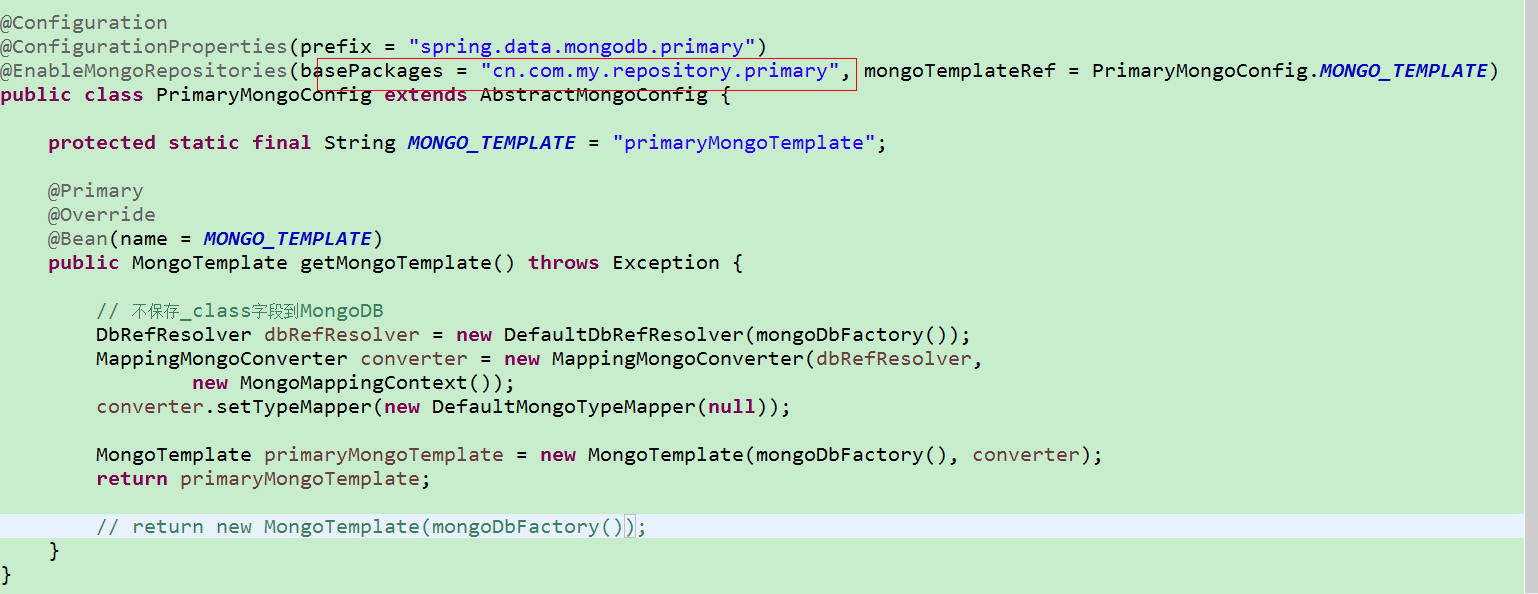
注：默认保存方法还有insert，区别在于save时，如果记录已存在，会覆盖插入，insert时，如果记录已存在，会抛异常。

## 多数据源CRUD

### 自定义抽象类实现MongoDbFactory方法



### 不同数据源配置实现（primary数据源）



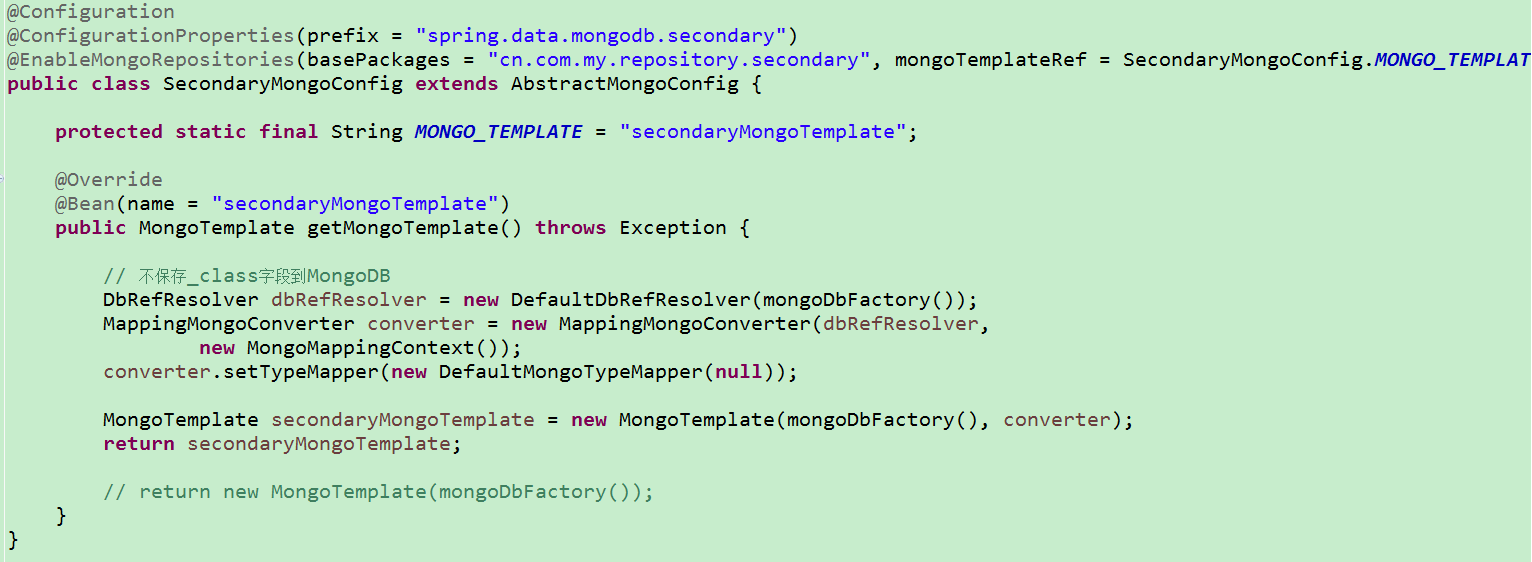
其中红框标识，如果通过继承MongoRepository方式实现数据操作，则该DAO接口类必须在此路径下。

该类同步实现了不在MongoDB数据库集合中保存\_class字段的效果。

默认插入数据时会产生数据实体类的路径，如下图：



### 不同数据源配置实现（secondary数据源）

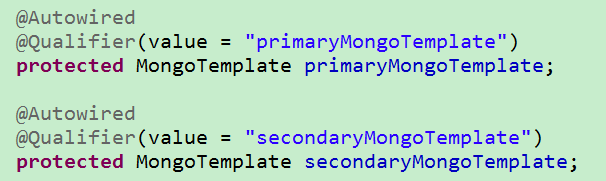


### 继承MongoRepository

不同数据源根据上面的配置类，指定的basePackages路径，继承MongoRepository，其它操作方式同单数据源一致。

### 使用MongoTemplate

注入时增加注解@Qualifier，如下图：



其它操作方式同单数据源时，使用MongoTemplate一致。

### 启动类排除Mongo配置

由于已自定义mongo配置，故需在启动类增加排除mongo默认的配置，如下图：



### 嵌套查询

当数据类型为嵌套类型，如下图：



需要根据嵌套的子对象属性做查询时，如根据storeId进行查询，则需要根据storeInfo.storeId进行字段匹配查询，如下图：



## 工程代码demo

## 工程代码打包上传至<https://lijingshan-1.ut.cn:8443/svn/SmartCloud/02document/技术预研/MongoDB>该目录下，详细实现和其它操作见该工程。