

原mongodb和springMVC整合

2017年08月30日 15:57:37

阅读数：1967

在对mongodb和springmvc的整合是十分简单的，以maven项目为例。具体分为以下几步：

maven配置

在整合之前需要导入jar包，在这里只需在pom.xml中加入依赖即可。

```
1 <!-- mongodb 开始 -->
2     <dependency>
3         <groupId>org.mongodb</groupId>
4         <artifactId>mongo-java-driver</artifactId>
5         <version>3.3.0</version>
6     </dependency>
7     <dependency>
8         <groupId>org.springframework.data</groupId>
9         <artifactId>spring-data-mongodb</artifactId>
10        <version>1.7.1.RELEASE</version>
11    </dependency>
12 <!-- mongodb 结束 -->
```

集群配置

applicationContext.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <beans xmlns="http://www.springframework.org/schema/beans"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:jee="http://www.springframework.org/schema/jee"
5     xmlns:tx="http://www.springframework.org/schema/tx" xmlns:util="http://www.springframework.org/schema/util"
6     xmlns:context="http://www.springframework.org/schema/context"
7     xmlns:mongo="http://www.springframework.org/schema/data/mongo"
8     xsi:schemaLocation="http://www.springframework.org/schema/beans
9     http://www.springframework.org/schema/beans/spring-beans-4.1.xsd
10    http://www.springframework.org/schema/tx
11    http://www.springframework.org/schema/tx/spring-tx-4.1.xsd
12    http://www.springframework.org/schema/jee
13    http://www.springframework.org/schema/jee/spring-jee-4.1.xsd
14    http://www.springframework.org/schema/context
15    http://www.springframework.org/schema/context/spring-context-4.1.xsd
16    http://www.springframework.org/schema/schema/util
17    http://www.springframework.org/schema/schema/util/spring-util-4.1.xsd
18    http://www.springframework.org/schema/data/mongo
19    http://www.springframework.org/schema/data/mongo/spring-mongo-1.5.xsd
20    http://www.springframework.org/schema/data/repository
21    http://www.springframework.org/schema/data/repository/spring-repository-1.5.xsd
22    ">
23
24 <!-- 自动扫描 -->
25 <context:annotation-config />
26 <!-- 设置Spring使用自动扫描功能，使"base-package"中的对象可以使用Annotation配置 -->
27 <context:component-scan base-package="com.**.*)" />
28
29 <mongo:mongo id="mongo" replica-set="${mongodb.replicaSet}" />
30 <mongo:db-factory id="mongoDbFactory" dbname="${mongodb.database}"
31     mongo-ref="mongo" />
32
33 <!-- mongo 模板操作对象 -->
34 <bean id="mongoTemplate" class="org.springframework.data.mongodb.core.MongoTemplate">
35     <constructor-arg name="mongoDbFactory" ref="mongoDbFactory" />
36     <!-- 配置读写分离的模式 -->
37     <property name="readPreference" ref="secondaryPreferredReadPreference"></property>
38 </bean>
39
40 </beans>
```

配置文件config.properties

```
1
2 #mongodb 配置文件
```

3		mongodb.database=bookStore	4		mongodb.replicaSet = 10.0.11.28:27017,10.0.11.29:27017,10.0.11.30:27017
5					

mongodb实例

mongodb对象模型

Book.java , 具体代码如下

```
1 package com.book.book.model;
2
3 import java.io.Serializable;
4
5 import org.springframework.data.annotation.Id;
6 import org.springframework.data.mongodb.core.mapping.Document;
7
8 @Document(collection = "book")
9 public class Book implements Serializable{
10
11     @Id
12     private String id;
13     // 书名
14     private String name;
15     // 作者
16     private String author;
17     // 价格
18     private float price;
19     // 图书封面
20     private String picture;
21     // 图书存储量
22     private Integer stock;
23     // 售出量
24     private int sale;
25     // 图书简介
26     private String message;
27
28     public String getId() {
29         return id;
30     }
31
32     public void setId(String id) {
33         this.id = id;
34     }
35
36     public String getName() {
37         return name;
38     }
39
40     public void setName(String name) {
41         this.name = name;
42     }
43
44     public String getAuthor() {
45         return author;
46     }
47
48     public void setAuthor(String author) {
49         this.author = author;
50     }
51
52     public float getPrice() {
53         return price;
54     }
55
56     public void setPrice(float price) {
57         this.price = price;
58     }
59
60     public String getPicture() {
61         return picture;
62     }
63
64     public void setPicture(String picture) {
65         this.picture = picture;
66     }
67
68     public Integer getStock() {
69         return stock;
70     }
71
72     public void setStock(Integer stock) {
73         this.stock = stock;
74     }
75 }
```

```
76 |         public int getSale() {
77 |             return sale;
78 |         }
79 |
80 |         public void setSale(int sale) {
81 |             this.sale = sale;
82 |         }
83 |
84 |         public String getMessage() {
85 |             return message;
86 |         }
87 |
88 |         public void setMessage(String message) {
89 |             this.message = message;
90 |         }
91 |     }
92 | }
```

接口BookService.java

```
1 | package com.book.book.service;
2 |
3 | /*
4 |  * 图书管理
5 |  */
6 | public interface BookService {
7 |
8 |     /*
9 |     * 查询所有图书
10 |    */
11 |    public String FindAll();
12 |
13 |    /*
14 |    * 根据图书名获取所有图书—模糊查询
15 |    *
16 |    * @name 图书名
17 |    *
18 |    * @return
19 |    */
20 |    public String getByName(String name);
21 |
22 |    /*
23 |    * 根据作者获取所有图书—模糊查询
24 |    *
25 |    * @author 作者
26 |    *
27 |    * @return
28 |    */
29 |    public String getByAuthor(String author);
30 |
31 |    /*
32 |    * 根据id获取所有图书
33 |    *
34 |    * @id 图书id
35 |    *
36 |    * @return
37 |    */
38 |    public String getId(String id);
39 | }
```

接口实现

```
1 | package com.book.book.service;
2 |
3 | import java.util.ArrayList;
4 | import java.util.List;
5 |
6 | import javax.ws.rs.Path;
7 |
8 | import org.springframework.beans.factory.annotation.Autowired;
9 | import org.springframework.beans.factory.annotation.Qualifier;
10 | import org.springframework.data.mongodb.core.MongoTemplate;
11 | import org.springframework.data.mongodb.core.query.Criteria;
12 | import org.springframework.data.mongodb.core.query.Query;
13 | import org.springframework.stereotype.Service;
14 |
15 | import com.book.book.model.Book;
16 |
17 | import net.sf.json.JSONObject;
18 |
19 | @Path("/Book")
20 | @Service
21 | public class BookServiceImpl implements BookService {
```

```
22 |         @Autowired
23 |         @Qualifier("mongoTemplate")
24 |         MongoTemplate mongoTemplate;
25 |
26 |
27 |     public String findAll() {
28 |         JSONObject jsonObject = new JSONObject();
29 |         try {
30 |             List<Book> bookList = new ArrayList<Book>();
31 |             bookList = mongoTemplate.findAll(Book.class);
32 |             jsonObject.put("result", 1);
33 |             jsonObject.put("msg", bookList);
34 |         } catch (Exception e) {
35 |             jsonObject.put("result", 0);
36 |             jsonObject.put("msg", "接口异常，请联系管理员！");
37 |         }
38 |
39 |         return jsonObject.toString();
40 |     }
41 |
42 |     public String getByName(String name) {
43 |         JSONObject jsonObject = new JSONObject();
44 |         try {
45 |             Criteria criteria = new Criteria();
46 |             criteria.andOperator(Criteria.where("name").regex(".*?\\" + name + ".*"));
47 |             List<Book> bookList = new ArrayList<Book>();
48 |             bookList = mongoTemplate.find(new Query(criteria), Book.class);
49 |             jsonObject.put("result", 1);
50 |             jsonObject.put("msg", bookList);
51 |         } catch (Exception e) {
52 |             jsonObject.put("result", 0);
53 |             jsonObject.put("msg", "接口异常，请联系管理员！");
54 |         }
55 |
56 |         return jsonObject.toString();
57 |     }
58 |
59 |     public String getByAuthor(String author) {
60 |         JSONObject jsonObject = new JSONObject();
61 |         try {
62 |             Criteria criteria = new Criteria();
63 |             criteria.andOperator(Criteria.where("author").regex(".*?\\" + author + ".*"));
64 |             List<Book> bookList = new ArrayList<Book>();
65 |             bookList = mongoTemplate.find(new Query(criteria), Book.class);
66 |             jsonObject.put("result", 1);
67 |             jsonObject.put("msg", bookList);
68 |         } catch (Exception e) {
69 |             jsonObject.put("result", 0);
70 |             jsonObject.put("msg", "接口异常，请联系管理员！");
71 |         }
72 |
73 |         return jsonObject.toString();
74 |     }
75 |
76 |     public String getId(String id) {
77 |         JSONObject jsonObject = new JSONObject();
78 |         try {
79 |             Criteria criteria = new Criteria();
80 |             criteria.andOperator(Criteria.where("id").is(id));
81 |             List<Book> bookList = new ArrayList<Book>();
82 |             bookList = mongoTemplate.find(new Query(criteria), Book.class);
83 |             jsonObject.put("result", 1);
84 |             jsonObject.put("msg", bookList);
85 |         } catch (Exception e) {
86 |             jsonObject.put("result", 0);
87 |             jsonObject.put("msg", "接口异常，请联系管理员！");
88 |         }
89 |
90 |         return jsonObject.toString();
91 |     }
92 |
93 | }
```

控制层

```
1 | package com.book.controllers;
2 |
3 | import java.util.ArrayList;
4 | import java.util.List;
5 |
6 | import org.springframework.beans.factory.annotation.Autowired;
7 | import org.springframework.data.mongodb.core.MongoTemplate;
8 | import org.springframework.data.mongodb.core.aggregation.Aggregation;
9 | import org.springframework.data.mongodb.core.query.Criteria;
10 | import org.springframework.data.mongodb.core.query.Query;
```

复制

```
11 | import org.springframework.stereotype.Controller;
12 | import org.springframework.web.client.RestTemplate;
13 |
14 | import com.book.book.model.Book;
15 | import com.book.book.service.BookService;
16 | import com.book.demo.service.DemoService;
17 |
18 | import net.paoding.rose.web.annotation.Param;
19 | import net.paoding.rose.web.annotation.Path;
20 | import net.paoding.rose.web.annotation.rest.Get;
21 |
22 | @Controller
23 | @Path("/")
24 | public class BookController {
25 |
26 |     @Autowired
27 |     BookService bookService;
28 |
29 |     @Autowired
30 |     MongoClient mongoClient;
31 |
32 |     @Autowired
33 |     RestTemplate restTemplate;
34 |
35 |     @Autowired
36 |     DemoService demoService;
37 |
38 |     @Get("findAllBooks")
39 |     public String findAll() {
40 |         String bookList = bookService.findAll();
41 |         return "@" + bookList;
42 |     }
43 |
44 |     @Get("getByName")
45 |     public String getByName(@Param("name") String name) {
46 |         String bookList = bookService.getByName(name);
47 |         return "@" + bookList;
48 |     }
49 |
50 |     @Get("getByAuthor")
51 |     public String getByAuthor(@Param("author") String author) {
52 |         String bookList = bookService.getByAuthor(author);
53 |         return "@" + bookList;
54 |     }
55 |
56 |     @Get("getById")
57 |     public String getById(String id) {
58 |         String bookList = bookService.getById(id);
59 |         return "@" + bookList;
60 |     }
61 |
62 | }
```

0

写评论

目录

收藏

微信

微博

QQ

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/hh12211221/article/details/77717364>

文章标签：mongodb document mongodb整合

个人分类：Java


相关热词：mongodb和 mongodb列 mongodb表 mongodb用 mongodb写


上一篇

zookeeper+dubbo和spring的整合

下一篇

linux命令ps aux|grep xxx详解

- 


想对作者说点什么
- 

技匠

2018-02-28 20:32:07

#2楼

而且还差很多jar包啊

[查看回复\(2\)](#)
- 

技匠

2018-02-26 16:51:22

#1楼

兄弟，secondaryPreferredReadPreference这个报错，以及配置文件没有扫描进applicationContext吧？对吗

spring mvc + spring + **mongodb 整合**

2014年10月26日

12.69MB

下载



区块链开发学习线图（高薪）

区块链开发平均薪资是多少？快速学习区块链开发，掌握以太坊开发和智能合约应用。

spring+springmvc+mongodb开发环境搭建 - CSDN博客

6-3

spring、springMVC、mongodb 集成,主要配置spring和springMVC及mongodb的配置即可,具体的配置如下: 1、集成的所需的jar文件:pom.xml <?xml version=...

【Maven+SSM】超详细Spring+SpringMVC+Mongo集成 - CSDN博客

7-29

本文其实和Mybatis无关,沿用ssm这个系列的文章,当然也是基于这个系列的文章代码接入MongoDb数据库,本文准确的说是Spring+SpringMVC+Mongo集成。 ...

SpringMVC整合Mongo**db**开发,高级操作

 461

开发环境: 操作系统：windows xp Mongodb：2.0.6 依赖包：Spring3.2.2 + spring-data-mongodb-1.3.0 + Spring-data-1...

Spring+SpringMVC+Mongo**DB**案例 - CSDN博客

6-10

MongoDB是一个NoSql数据库,MongoDB 是一个基于分布式文件存储的数据库。由 C++...下面是通过一个Spring+SpringMVC+MongoDB的案例来说明mongodb的基本使用。mongodb...

架构之路之spring+springmvc+maven+mongodb的搭建 - CSDN博客

5-27

最近项目开发使用到了spring+springmvc+maven+mongodb,项目中的框架是用springboot进项开发的,对于我们中级开发人员来说,有利有弊,好处呢是spri...

SpringMVC+Mongo**DB**整合及注意事项

 4494

废话不多说 这里先带上项目的各jar包，因为版本之间的差异所爬的坑，已经数不胜数：//重要的jar都圈出来了，其他包可能与本题...

springmvc mongodb maven 整合

 5395

一,项目说明 本项目使用spring springmvc mongodb maven整合的一个demo,欢迎大家预览,不足之处还请谅解. 或者给我回复.谢谢! ...

spring mvc + spring + **mongodb 整合**

7-24

使用spring mvc + spring data mongodb + mongodb +spring 整合,支持html压缩,支持mongodb用户登录,重写了spring的MongoTemplate支持更多方式,重写MongoTemplate支持...

SpringMVC整合Mongo**db**开发,高级操作 - CSDN博客

6-18

开发环境: 操作系统:windows xp Mongodb:2.0.6 依赖包:Spring3.2.2 + spring-data-mongodb-1.3.0 + Spring-data-1.5 + mongodb...

SpringMVC+mongodb应用实例

2016年08月16日

15.48MB

下载



码农怎能不懂英语？！试试这个数学公式
老司机教你一个数学公式秒懂天下英语



springMVC+mongo**DB**基础项目框架

7-25

1、这个项目是 springMVC+mongoDB,用的 mongoTemplate 做和数据库的链接, 日志用的log4j 2、项目中集成了 完整的单元测试,和基本的添加和查询,具体的api可以查看...

SpringMVC+Mongo**DB**整合及注意事项 - CSDN博客

6-21

spring: applicationContext-init.xml 其他配置我就直接省略了,关键是要加上jackson的配置,接着引入mongodb-config.xml。。。 <!-- 启用spring mvc 内...

spring mvc **mongodb**

 215

spring整合mongodb

Spring Data **MongoDB** 一：入门篇（环境搭建、简单的CRUD操作）

 4.8万

Spring Data MongoDB 项目提供与MongoDB文档数据库的集成。Spring Data MongoDB POJO的关键功能区域为中心的模式与Mon...

Spring+SpringMVC+Mongo**DB**+Maven整合

7-24

通过IDEA搭建maven工程整合Spring+SpringMVC+MongoDB,通过bootstrap编写的jsp页面,主要功能有用户注册,用户登录,添加用户,删除用户,更新用户,多条件搜索,停用用户,启用...

idea搭建springdata+mongodb+maven+springmvc

 793

idea搭建springdata+mongodb+maven+springmvc项目环境：IntelliJ IDEA2017+maven3.5.0+MongoDB 3.2+JDK1.7+spring4...

springmvc + **mongodb 整合**

 448

前几天准备做一个项目，由于数据横向可拓展性，最终选择了用mongodb 数据库 本来以为应该是很好配的，结果弄了好几天才将...

SpringMVC和MongoDB整合Demo+文档

2016年01月13日

22.63MB

下载



SpringMVC项目(MongoDB3.0)

2017年02月16日

46.59MB

下载



SpringMvc整合mongodb maven项目

SpringMvc整合mongodb maven项目 先新建一个maven project pom.xml配置如下：3.2.3.RELEASE 1.7.2 1.2.16 4.11 ...

1003

对于程序员来说，英语到底多重要

不背单词和语法，一个公式学好英语



Spring+SpringMVC+MongoDB案例

MongoDB是一个NoSql数据库，MongoDB 是一个基于分布式文件存储的数据库。由 C++ 语言编写。旨在为 WEB 应用提供可扩展...

2748

SpringMVC+MongoDB+Maven整合

SpringMVC+MongoDB+Maven整合

679

Spring MVC中使用Mongodb总结

近期项目做了次架构调整，原来是使用MySQL+GeoHash来存储LBS数据（地理位置信息），现在使用NOSQL数据库Mongodb来存...

1.3万

SpringMVC整合MongoDB开发 架构搭建

SpringMVC整合MongoDB开发 架构搭建 [日期：2015-02-28] 来源：Linux社区 作者：dennisit [字体：大 中 小] ...

262

SpringMVC整合MongoDB开发 高级操作

SpringMVC整合MongoDB开发 高级操作 [日期：2015-02-28] 来源：Linux社区 作者：dennisit [字体：大 中 小] ...

1598

mongodb整合入spring

测试spring整合mongodb，并批量插入数据demo

646

spring集成mongodb

public class Country implements Serializable { //实体类参考本人博客地址：https://blog.csdn.net/yl_hahha/arti...

33

个人资料



wisdom_bo

关注

原创	粉丝	喜欢	评论
69	9	24	27

等级：	博客 4	访问：	11万+
积分：	1680	排名：	3万+
勋章：			

最新文章

- ajaxfileupload.js解决浏览器兼容问题
- java itext根据模板生成pdf
- windows环境下启动mongodb服务
- mongodb——数据的导入

mongodb——复制数据库和表

个人分类

linux	4篇
机器学习	7篇
Java	23篇
数据库	11篇
服务器	4篇

展开

归档

2018年5月	2篇
2017年12月	5篇
2017年9月	3篇
2017年8月	10篇
2017年7月	5篇

展开

热门文章

- centos7下安装Anaconda
阅读量：22081
- anaconda 使用import sklearn.model_select
ion 出错ImportError: No module named mo
阅读量：16693
- bootstrap-table.js多种方法填充单元格数据
阅读量：7989
- 配置maven环境变量出错
阅读量：7126
- nodejs package.json详解
阅读量：5345

最新评论

- windows下maven私服ne...
n950814abc：感谢博主的分享
- java JSON转String
qq_28197211：标题写反了
- anaconda 使用import...
lu_fun：LOCKERROR: It looks like conda is alrea
dy doing s...
- anaconda 使用import...
lu_fun：[reply]weixin_37029453[/reply] 用pip insta
ll --upg...
- 配置maven环境变量出错
qq_34809531：原本设置的环境变量是系统的Pat
h，将MAVEN_HOME改为用户的，然后添加到用
户的PATH中就...