

## 尼斯湖水鬼的博客

SpringBoot开发, SpringCloud开发, Java开发



RSS订阅

### SpringBoot开发详解（四）--JDBC操作数据库

2017年04月16日 18:07:00

阅读数: 8131

9

目录

收藏

使用JDBC连接数据库:

通过上几篇幅的介绍，我们已经能够构建一个简单的，没有数据库的API接口项目，可是作为一个接口程序，没有数据库就是在耍流氓。那么本文我们就介绍下最简单的通过JDBC连接数据库的操作。

POM依赖添加

在POM文件中我们需要引入一些依赖来配置数据源，我们在pom.xml中加入以下依赖：

```

1 <!--数据库连接jdbc依赖-->
2 <dependency>
3   <groupId>org.springframework.boot</groupId>
4   <artifactId>spring-boot-starter-jdbc</artifactId>
5 </dependency>
6 <!--mysql连接依赖-->
7 <dependency>
8   <groupId>mysql</groupId>
9   <artifactId>mysql-connector-java</artifactId>
10 </dependency>
11 <!--阿里druid数据库连接依赖-->
12 <dependency>
13   <groupId>com.alibaba</groupId>
14   <artifactId>druid</artifactId>
15   <version>1.0.5</version>
16 </dependency>
```

这里我加入了一个阿里的druid驱动，因为相比较于我们一直使用的jdbc驱动，druid是一个可监控的驱动，可以说是一个为监控而生的数据库驱动。想更详细了解的同学可以去druid的GitHub上查看druid

添加完依赖后，我们需要在配置文件中配置数据源信息，这里，我们配置下在之前建立的application-dev.yml下

```

1 spring:
2   datasource:
3     url: jdbc:mysql://localhost:3306/blog
4     username: root
5     password: 123
6     driver-class-name: com.mysql.jdbc.Driver
7     type: com.alibaba.druid.pool.DruidDataSource
```

完成以上操作的同学，我们的开发已经完成一半了，是的，Spring Boot就是这么简单。不需要写配置类，不需要额外的代码书写。开箱即用在Spring Boot的身上再一次完美体现。

### 使用JdbcTemplate

在书写代码前，有很重要的一件事要做，就是创建数据库以及数据表，这里我在blog库下创建一张tp\_user表。

```

1 CREATE TABLE `tp_user` (
2   `user_id` int(10) NOT NULL AUTO_INCREMENT COMMENT '用户ID',
3   `tel` bigint(19) NOT NULL COMMENT '手机号=用户名',
4   `password` varchar(35) NOT NULL COMMENT '密码',
5   `nickname` varchar(30) DEFAULT NULL COMMENT '昵称',
6   `secret` varchar(35) NOT NULL COMMENT '秘钥',
7   `portrait` varchar(120) DEFAULT NULL COMMENT '头像地址',
8   `i_card` varchar(18) NOT NULL DEFAULT '' COMMENT '身份证',
9   `area` int(10) NOT NULL DEFAULT '0' COMMENT '所在区域',
10  `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '添加时间',
11  `update_time` datetime DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间'
```

### 个人资料



尼斯湖水鬼

[关注](#)

原创  
12

粉丝  
185

喜欢  
23

评论  
43

等级: 博客 4 访问: 13万+

积分: 1090 排名: 4万+



液态硬盘



### 最新文章

[SpringBoot开发详解（十二）--SpringBoot中执行定时任务](#)

[SpringBoot开发详解（十一）--Redis在SpringBoot中的整合](#)

[SpringBoot开发详解（十）--使用JPA访问数据库下篇及使用Page进行数据分页](#)

[SpringBoot开发详解（九）--使用JPA访问数据库上篇](#)

[SpringBoot开发详解（八）--使用Swagger2构建API文档](#)

### 个人分类

spring boot

12篇

### 归档

2017年7月

2篇

2017年6月

1篇

2017年5月

5篇

2017年4月

3篇

2017年2月

1篇

### 热门文章

[SpringBoot开发详解（五）--Controller接收参数以及参数校验](#)  
阅读量: 55581

[SpringBoot开发详解（三）--SpringBoot配置文件YML注意事项](#)  
阅读量: 17424

[SpringBoot开发详解（十二）--SpringBoot中执行定时任务](#)  
阅读量: 12598

```
12 PRIMARY KEY (`user_id`)
13 ) ENGINE=InnoDB AUTO_INCREMENT=238 DEFAULT CHARSET=utf8
```

想再我们可以开始代码层面的编写工作了，在这里我先建立了一个UserController类来获取前端传来的参数信息，这部分我们之前已经说过，这里不再赘述。然后将参数传入service层进行业务逻辑处理，之后由dao层负责与数据库交互。是的，这套路就是我们平时在是集开发工作中会用到的，所以我的文章里也会更多的将工作中使用的一些开发习惯以及开发技巧告诉大家。

UserController:

```
1 @RequestMapping("/user")
2 @RestController
3 public class UserController {
4
5     @Autowired
6     private UserService userService;
7
8     /**
9      * 添加用户
10     * @param tel 注册手机号
11     * @param pwd 设置密码
12     */
13     @PostMapping("/createUser")
14     public void createUser(@RequestParam("tel") String tel, @RequestParam("pwd") String pwd){
15         userService.createUser(tel,pwd);
16     }
17
18     /**
19      * 更新用户信息
20      * @param user_id 用户ID
21      * @param nickName 昵称
22      */
23     @PutMapping("/updateUser/{id}")
24     public void updateUser(@PathVariable("id") String user_id, @RequestParam("nickName") String nickName){
25         userService.updateUser(user_id,nickName);
26     }
27
28     /**
29      * 获取用户信息
30      * @param id 用户Id
31      * @return
32      */
33     @GetMapping("/getUser/{id}")
34     public UserInfo getUser(@PathVariable("id") Integer id){
35         return userService.getUser(id);
36     }
37
38     @DeleteMapping("/deleteUserById/{id}")
39     public void deleteUserById(@PathVariable("id") Integer id){
40         userService.deleteUserById(id);
41     }
42 }
```

注意：在这里使用到的@GetMapping, @PostMapping, @PathVariable, @RequestParam等注解，我们在下一篇中再仔细讨论下，这里我们先使用。

@UserService:

```
1 @Service
2 public class UserService {
3
4     @Autowired
5     UserInfoMapper userInfoMapper;
6
7     public void createUser(String tel,String pwd) {
8         userInfoMapper.createUser(tel,pwd);
9     }
10
11     public UserInfo getUser(Integer id) {
12         return userInfoMapper.getUser(id);
13     }
14
15     public void updateUser(String user_id, String nickName) {
16         userInfoMapper.updateUser(user_id,nickName);
17     }
}
```

SpringBoot开发详解（六）-- 异常统一管理  
以及AOP的使用

阅读量: 9583

SpringBoot开发详解（十一）-- Redis在SpringBoot中的整合

阅读量: 8411

### 最新评论

SpringBoot开发详解（十一）...

xia\_zai\_pin\_dao\_: 按照博主的思路写下来后，程序运行是正确的。但是在redis的客户端却是乱码，而且看不到值。但是用代...

SpringBoot开发详解（六）...

qq\_35981283: 很好的文章

SpringBoot开发详解（一）...

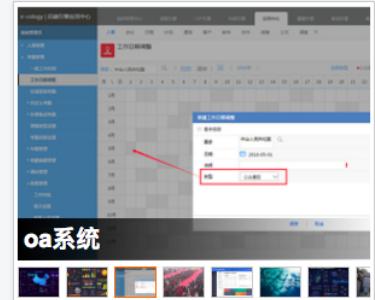
qq\_31211685: 很详细，赞一个

SpringBoot开发详解（六）...

weixin\_42096420: 楼主写得很好，感谢，接下来的我都要看看，太棒了

SpringBoot开发详解（五）...

sss1342746626: 感谢楼主。。终于发现一个接收数据好用的方法了。。。赞



oa系统

### 联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

✉ QQ客服 客服论坛

关闭 报错 广告服务 帮助 反馈

```

18
19     public void deleteUserByUserId(Integer id) {
20         userInfoMapper.deleteUserByUserId(id);
21     }
22 }
```

在service类中我们通常通过@Service注解来标示这是一个service，由spring在启动时自动扫描进容器中。

UserInfoMapper:

```

1 public interface UserInfoMapper {
2
3     void createUser(String tel, String pwd);
4
5     UserInfo getUser(Integer id);
6
7     void updateUser(String user_id, String nickname);
8
9     void deleteUserByUserId(Integer id);
10 }
```

创建一个抽象接口来定义我们对user的CRUD操作。

UserInfoImpl:

```

1 @Repository
2 public class UserInfoImpl implements UserInfoMapper {
3
4     @Autowired
5     private JdbcTemplate jdbcTemplate;
6
7     @Override
8     public void createUser(String tel, String pwd) {
9         jdbcTemplate.update("INSERT INTO blog.tp_user(tel,password,nickname,secret) VALUES (?)");
10    }
11
12     @Override
13     public UserInfo getUser(Integer id) {
14         List<UserInfo> userList = jdbcTemplate.query("select tel,nickname,password FROM blog.");
15         if(userList != null && userList.size() > 0){
16             UserInfo user = userList.get(0);
17             return user;
18         }else {
19             return null;
20         }
21     }
22
23     @Override
24     public void updateUser(String user_id, String nickname) {
25         jdbcTemplate.update("UPDATE blog.tp_user SET nickname = ? WHERE user_id = ?", nickname);
26     }
27
28     @Override
29     public void deleteUserByUserId(Integer id) {
30         jdbcTemplate.update("DELETE FROM blog.tp_user WHERE user_id = ?", id);
31     }
32 }
```

我们在实现类中直接通过@Autowired注解来加载JdbcTemplate，通过@Repository注解来让spring自动加载，如果你对目前是哪一层不清楚，或者这只是你希望spring去自动加载的，你也可以使用@Componet注解。

做到上面的工作时，其实我们已经完成了一整个数据库读写的操作。但我在这里还是像啰嗦两句，其一，在实际开发中，我们对用户的敏感信息不做明文的保存，一般像密码这类的我们使用MD5加密后存进数据，因为MD5是单向的。如果是身份证一类的信息，我们通常使用对称加密后保存。其二，在使用JdbcTemplate时，有一个方法是queryForObject，要注意的是，该方法只能返回简单类型以及String，无法返回一个实体类。

```

@Override
public UserInfo getUser(Integer id) {
    List<UserInfo> userList = jdbcTemplate.queryForObject("select tel,nickname,password FROM blog.tp_user WHERE user_id = ?", new Object[]{id}, new UserInfo());
    if(userList != null && userList.size() > 0){
        UserInfo user = userList.get(0);
        return user;
    }else {
        return null;
    }
}
```

```

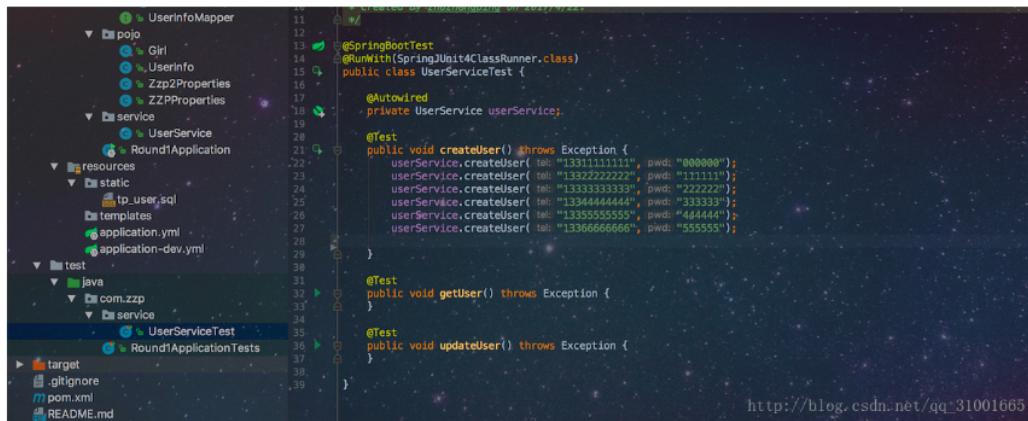
@Override
public void updateUser(String user_id) {
    jdbcTemplate.update("UPDATE blog_tp_user SET ... WHERE user_id = ? ", id);
}

@Override
public void deleteUserById(Integer id) {
    jdbcTemplate.update("DELETE FROM blog_tp_user WHERE user_id = ? ", id);
}

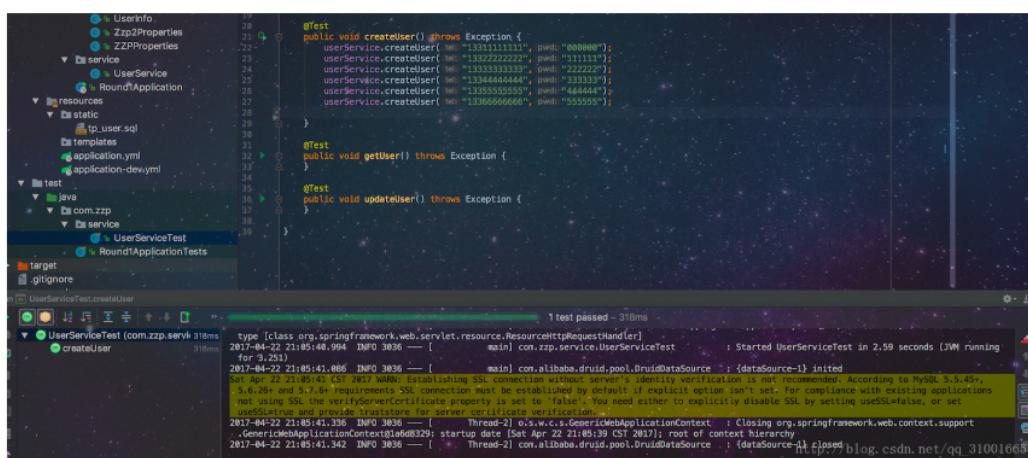
```

http://blog.csdn.net/qq\_31001665

好了，接下来我们开始测试吧，我们首先通过单元测试来添加一些用户信息。我们只需要在service中对想测试的方法右键，选择`goto->test`，IDEA会自动帮我们在`test`目录下建立测试单元。



我们增加一点代码，然后点击左侧的运行按钮，会看到在输出台中我们的代码执行成功了，绿色代表成功，红色则是失败。



接着我们继续使用POSTMAN进行测试，在使用POSTMAN时要注意的是我们调用添加接口时，要在body中选择form-data一项来添加参数，如果使用`@PutMapping`时，我们要使用`x-www-form-urlencoded`一项，这是由于http的机制不同造成。

POST localhost:9090/user/createUser

Authorization Headers Body Params Send Save Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Bulk Edit
tel	13311111111	
pwd	123456	

Body Cookies (1) Headers (3) Tests Status: 200 OK Time: 14 ms Size: 222 B

Pretty Raw Preview JSON

http://blog.csdn.net/qq\_31001665

PUT localhost:9090/user/updateUser/1

Authorization Headers (1) Body Params Send Save Cookies Code

form-data

Key	Value
tel	13311111111

Body Cookies (1) Headers (3) Tests Status: 200 OK Time: 14 ms Size: 222 B

Pretty Raw Preview JSON

http://blog.csdn.net/qq\_31001665

form-data x-www-form-urlencoded raw binary

Key	Value
nickName	亚洲第一帅
New key	value

Body Cookies (1) Headers (3) Tests Status: 200 OK Time: 14 ms Size: 222 B

Pretty Raw Preview JSON

http://blog.csdn.net/qq\_31001665/article/details/70196825

以上就是我们使用JDBC连接数据库的完整案例了，我们可以看到，开发十分的简单，都不用创建Bean就可以连接数据库了，不过一般开发中我们不使用JDBC直接连接（写完你才说！你TM在逗我！），而是更多的使用ORM框架来做数据库交互。在以上的案例中其实任然有许多值得优化的地方，比如如果我输入参数很多，是不是在controller层中要写很多参数来一个一个接收呢？眼尖的小伙伴应该已经发现我在返回用户信息时使用了User实体类。如果查询用户故意用小于0的数字查询呢？又或者有人恶意注册怎么限制呢？如果报错了，怎么不在界面展示一堆报错信息呢？有没有更好体验的报错提示呢？这些，我们下次一起来聊聊，聪明的你可以先想一下，或许你的想法比我想到的更好更完善~~~

以上所有的代码我已经上传到GitHub

## round1-springboot

如果心急的小伙伴也可以去clone我已经完成的项目，这个项目中把一些常用功能都写了，并且都写注释啦！！！

## MySpringBoot

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/qq\\_31001665/article/details/70196825](https://blog.csdn.net/qq_31001665/article/details/70196825)

文章标签： spring

个人分类： spring boot



那都不是事！学会AI这些技能，让你叱咤人工智能领域！

时代趋势，顺流而上！掌握人工智能，避免就业危机！成为稀缺人才我们有方法...

12522

查看更多>>

想对作者说点什么？ [我来说一句](#)

小炫风 2018-04-08 22:49:08 #5楼

idea 骚气

我敢说畅游 2018-03-23 23:49:38 #4楼

Description: The Bean Validation API is on the classpath but no implementation could be found Action: Add an implementation, such as Hibernate Validator, to the classpath 怎么解决这个问题啊？

wandong01 2018-01-24 19:49:16 #3楼

查看回复(2)

你的idea主题是什么？

查看 7 条热评

## Spring Boot JDBC 连接数据库

文本将对在Spring Boot构建的Web应用中，基于MySQL数据库的几种数据库连接方式进行介绍。包括JDBC、JPA、MyBatis、多数据源和事务。JDBC1、属性配置文件（applica...

catoop 2016-01-12 23:31:11 阅读数: 90625

## spring boot学习系列：spring boot与jdbcTemplate的整合案例

简单入门了spring boot后，接下来写写跟数据库打交道的案例。博文采用spring的jdbcTemplate工具类与数据库打交道。...

u013871100 2017-03-31 15:41:05 阅读数: 8429

本篇内容来自《深入实战springboot》一书 对于传统关系型数据库, Spring Boot可以使用JPA资源库来实现对数据库的操作,即将Java的普通对象通过对象关系映射持久化到数据...

2018-5-17

## springboot学习笔记--Spring Boot使用数据库(MySQL) - CSDN博客

本篇内容来自《深入实战springboot》一书 对于传统关系型数据库, Spring Boot可以使用JPA资源库来实现对数据库的操作,即将Java的普通对象通过对象关系映射持久化到数据...

2018-5-17

## spring boot + JdbcTemplate 使用druid数据源

添加依赖 org.springframework.boot spring-boot-starter-jdbc ...

u014695188 2016-08-21 00:45:19 阅读数: 8568

## SpringBoot进阶之访问数据库(含源码) - CSDN博客

本文介绍在Spring Boot基础上配置数据源和通过JdbcTemplate编写数据访问的示例。简单介绍一下 @Controller:修饰class,用来创建处理http请求的对象 @RestController:Spring...

2018-5-18

## SpringBoot的数据库操作 - CSDN博客

<https://blog.csdn.net/fzz19960915/article/details/72804098> 文章标签: springboot 数据库 javaweb postman 个人分类: SpringBoot ...

2018-5-8

## Maven搭建Spring Boot JDBC

编写pom.xml添加依赖

goodstuddayuppyeah 2017-05-17 20:30:33 阅读数: 1157

## 第一个 Spring Boot 程序：使用 spring jdbc 访问关系型数据库

文档地址: <http://blog.csdn.net/lzlovez/article/details/53761091> pom.xml

Lzlovez 2016-12-26 14:56:51 阅读数: 732

## Springboot对数据库简单的增删(-)查 - CSDN博客

SpringBoot中连接MySQL数据库,并使用JPA进行数据库的相关操作 今天给大家介绍一下如何SpringBoot中连接Mysql数据库,并使用JPA进行数据库的相关操作。步骤一:在pom.xml...

2018-5-17

## spring boot数据库操作(以mysql为例) - CSDN博客

根据慕课网视频教程简单整理了数据库操作流程... 添加数据库支持 在maven中添加组件(pom.xml) <dependency> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-starter-data-jpa</artifactId> </dependency>

2018-4-12

## Spring Boot 菜鸟教程 4 JDBC

JDBC Java Data Base Connectivity,是一种用于执行SQL语句的Java API, 可以为多种关系数据库提供统一访问, 它由一组用Java语言编写的类和接口组成。不管是Hiber...

JE\_GE 2016-11-25 22:41:47 阅读数: 2268

## SpringBoot 数据库的基本操作 - CSDN博客

SpringBoot 数据库的基本操作 本博文将记录下SpringBoot 关于数据库的基本操作,比较简单哈,但也有一些知识点需要我们注意。假设数据库中有一个Student表,本文实现如下...

2018-5-3

## Spring boot操作mysql数据库 - CSDN博客

前面一篇已经搭建好了spring boot的开发环境并且完成了第一个Hello world程序,本文紧接着上篇的内容,开始数据库相关的知识点。工程相关配置 pom.xml 在dependenci...

2018-5-20

## SpringBoot - JDBC 连接数据库

按照下列顺序即可配置好 SpringBoot 采用 JDBC 方式链接数据库 1、配置 Maven 依赖 (pom.xml) 所需的依赖 &lt;dependencies&a...

qq\_34158598 2018-03-14 12:54:21 阅读数: 41

## SpringBoot非官方教程 | 第三篇：SpringBoot用JdbcTemplate访问Mysql

本文介绍springboot通过jdbc访问关系型mysql,通过spring的JdbcTemplate去访问。

forezp 2017-04-22 22:31:56 阅读数: 44491

## Spring boot + jdbc学习笔记

pom.xml:

sunhuwh 2016-08-05 11:18:20 阅读数: 4060

## springboot学习笔记(二):基于MySql数据库的JDBC操作

1 , 连接数据库SpringBoot的Web应用中, 基于MySql数据库的JDBC操作 JDBC 连接数据库主要配置 1 , 属性配置文件 (application.properties) spring...

zhangheng1633 2016-09-01 19:00:26 阅读数: 8309

## Spring Boot采用JDBC方式连接数据库

spring boot采用jdbc方式连接数据库

concisefreedom 2017-04-23 22:47:44 阅读数: 4974

## 后台springBoot+jdbc例子

2017年04月19日 71.61MB 下载

## spring boot(10)-tomcat jdbc连接池

默认连接池 tomcat jdbc是从tomcat7开始推出的一个连接池, 相比老的dbcp连接池要优秀很多。spring boot将tomcat jdbc作为默认的连接池, 只要在pom.xml中引入...

wangb\_java 2017-07-09 11:18:26 阅读数: 6496

## 高效、准确、低成本的文字识别api

证件、文件一键扫描, 准确率高达99%。经过海量用户和多种复杂场景考验



## springboot与jdbc

本文将介绍如何将spring boot 与 JdbcTemplate一起工作。Spring对数据库的操作在jdbc上面做了深层次的封装, 使用spring的注入功能, 可以把DataSource...

qq\_16887777 2016-07-12 08:26:42 阅读数: 286

## springboot手写JDBC面对2000并发量毫无压力

声明: 本博客所有文章都是原创, 不会有任何抄袭现象。请转载的朋友, 标注出处, 谢谢。最近比较流行springboot微服务, 那么持久层到底怎么样才算好? 面对高并发的压力, 如何抵制? 本文将以JDBC来...

u011550710 2017-07-26 10:38:29 阅读数: 4391

## Spring Boot Mysql 数据库操作

本文默认你的开发环境, 数据库已经安装好 想使用使用数据库, 我们需要现在pom文件中添加相应的依赖 pom.xml ...

u010399316 2016-12-01 13:57:40 阅读数: 2586

## SpringBoot 数据库操作(下)

新建DTO接口 新建DTO接口并继承JpaRepository。指定操作的类与ID的类型 public interface GrilDto extends JpaRepository { ...

 FTDD\_HW 2017-08-09 14:20:43 阅读数: 169

## Spring Boot实战之Rest接口开发及数据库基本操作

Spring Boot实战之Rest接口开发及数据库基本操作 1、修改pom.xml，添加依赖库,本文使用的是mysql

org.springframework...

 sun\_t89 2016-07-14 21:07:30 阅读数: 24753

## 广州购房如何入户广州

广州入户

百度广告



## Spring Boot 微框架学习(利用Spring Boot编写一个访问数据库的helloworld)

Spring Boot 项目旨在简化创建产品级的 Spring 应用和服务。你可通过它来选择不同的 Spring 平台。可创建独立的 Java 应用和 Web 应用，对于创建的Web应用甚至不需要启动...

 canot 2016-05-19 00:19:39 阅读数: 4867

## SpringBoot中连接MYSQL数据库，并使用JPA进行数据库的相关操作

今天给大家介绍一下如何SpringBoot中连接Mysql数据库，并使用JPA进行数据库的相关操作。步骤一：在pom.xml文件中添加MYSQL和JPA的相关Jar包依赖，具体添加位置在depende...

 linzhiqiang0316 2016-09-21 16:52:50 阅读数: 21236

## SpringBoot操作MySQL数据库