

## 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 (七) -- Mybatis整合Spring Boot

2017年05月14日 00:52:21

阅读数: 1762

0

目录

收藏

评论

微信

微博

## Mybatis在SpringBoot中的使用

使用Mybatis作为ORM框架:

如今的接口开发中, ORM框架是我们操作数据库不可或缺的一部分, 而其中Hibernate与Mybatis是最常用的两大框架, 其中Hibernate学习周期较长, 因为它使用HQL语言代替SQL, 以对象为单位操作数据库。而Mybatis使用SQL, 也是最为直观的, 绝大多数的使用Mysql的互联网公司使用Mybatis作为ORM框架, 今天, 我们就来看一下Mybatis在SpringBoot中的整合~~~

引入Mybatis依赖:

```
1 <!--mybatis在spring-boot中的依赖-->
2 <dependency>
3   <groupId>org.mybatis.spring.boot</groupId>
4   <artifactId>mybatis-spring-boot-starter</artifactId>
5   <version>1.2.0</version>
6 </dependency>
```

在引入Mybatis依赖后, 这里有个小贴士, 我们可以不再引入spring-boot-starter-jdbc的依赖, 因为Mybatis的依赖已经存在JDBC依赖了。

添加数据库配置信息:

```
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
```

这里使用的数据库配置信息和我们之前使用JDBC时是一致的, 无需更改。

接下来我们只需在我们的代码中直接使用即可。

使用注解配置:

首先我们先使用注解方式来配置Mybatis, 我们还是使用之前用户的增删改查四种操作, 我们重新建立controller, service以及dao:

controller:

```
1 @RestController
2 @RequestMapping("/mybatis")
3 public class MybatisController {
4
5   @Autowired
6   private MybatisService service;
7
8   /**
9    * 更新用户信息
10   * @param user_id 用户ID
11   * @param nickName 昵称
12   */
13   @PutMapping("/updateUser/{id}")
14   public void updateUser(@PathVariable("id") String user_id, @RequestParam("nickName") String
15                         service.updateUser(user_id,nickName);
16 }
```

### 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+



希腊房价



### 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Swagger 2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

```

18 /**
19  * 获取用户信息
20  * @param id 用户Id
21  * @return
22 */
23 @GetMapping("/getUser/{id}")
24 public UserInfo getUser(@PathVariable("id") Integer id){
25     return service.getUser(id);
26 }
27
28 /**
29  * 删除用户
30  * @param tel
31 */
32 @DeleteMapping("/deleteUserByUserId/{tel}")
33 public void deleteUserByUserId(@PathVariable("tel") String tel){
34     UserInfo user = new UserInfo();
35     user.setTel(tel);
36     service.deleteUserByUserId(user);
37 }
38
39 /**
40  * 使用@RequestBody获取参数，用map类型接收，再取出
41  * @param reqMap
42 */
43 @PostMapping("/createUserByMap")
44 public void createUserByMap(@RequestBody Map<String, Object> reqMap){
45     service.createUser(reqMap);
46 }
47
48
49 }

```

service:

```

1 @Service
2 public class MybatisService {
3
4     @Autowired
5     private MybatisUserAnnotaionMapper annotaionMapper;
6
7
8     public void updateUser(String user_id, String nickName) {
9         annotaionMapper.updateUser(user_id, nickName);
10    }
11
12     public UserInfo getUser(Integer id) {
13         return annotaionMapper.getUser(id);
14    }
15
16     public void deleteUserByUserId(UserInfo userInfo) {
17         annotaionMapper.deleteUserByUserId(userInfo);
18    }
19
20     public void createUser(Map<String, Object> reqMap) {
21         annotaionMapper.createUser(reqMap);
22    }
23 }

```

dao:

```

1 @Mapper
2 public interface MybatisUserAnnotaionMapper {
3
4     @Insert("INSERT INTO blog.tp_user(tel,password,nickname,secret) VALUES (#{tel},md5(#{pwd})
5     void createUser(Map<String, Object> reqMap);
6
7     @Select("select tel,nickname,password FROM blog.tp_user WHERE user_id = #{id}")
8     UserInfo getUser(@Param("id") Integer id);
9
10    @Update("UPDATE blog.tp_user SET nickname = #{nickName} WHERE user_id = #{userId}")
11    void updateUser(@Param("userId") String user_id, @Param("nickName") String nickName);
12
13    @Delete("DELETE FROM blog.tp_user WHERE tel = #{tel}")
14    void deleteUserByUserId(UserInfo userInfo);

```

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）...

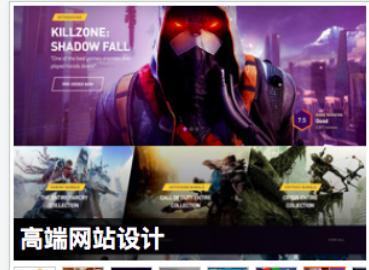
qq\_31211685: 很详细，赞一个

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



高端网站设计

### 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

在这里要说明的是，我们可以使用@param对每一个参数进行传值，也可以直接传入Map类型的参数，只需在#{}内写入MAP的KEY，同样我们也可以使用对象进行传值，#{ }内写入的是对象的属性名。

我们完成上面代码后使用POSTMAN测试一下：

The screenshot shows the POSTMAN interface. In the 'Body' tab, there is a JSON payload:

```

1 {
2   "tel": "13800000000",
3   "pwd": "12345"
4 }

```

In the 'Body' section of the response, the status is -1, msg is '未知错误', and data is null.

Status: 200 OK Time: 15760 ms Size: 176 B

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

我们发现报错了，显示未知错误，我们来看一下控制台的输出日志：

The terminal output shows a stack trace for a NullPointerException:

```

java.lang.NullPointerException: null
at com.zpp.util.HttpAspect.doAfterReturning([HttpAspect.java:60] ~[classes/:na] +4 internal calls)
at org.springframework.aop.support.AbstractAspectJAdvice.invokeAdviceIfApplicable([AbstractAspectJAdvice.java:151] ~[spring-aop-4.3.7.RELEASE.jar:4.3.7.RELEASE])
at org.springframework.aop.support.AbstractAspectJAfterReturningAdvice.afterReturning([AspectJAfterReturningAdvice.java:69] ~[spring-aop-4.3.7.RELEASE.jar:4.3.7.RELEASE])
at org.springframework.aop.framework.adapter.AfterReturningAdviceInterceptor.invoke([AfterReturningAdviceInterceptor.java:53] ~[spring-aop-4.3.7.RELEASE.jar:4.3.7.RELEASE])
at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed([ReflectiveMethodInvocation.java:179] ~[spring-aop-4.3.7.RELEASE.jar:4.3.7.RELEASE])
at org.springframework.aop.interceptor.ExposeInvocationInterceptor.invoke([ExposeInvocationInterceptor.java:92] ~[spring-aop-4.3.7.RELEASE.jar:4.3.7.RELEASE])
at org.springframework.aop.framework.JdkDynamicAopProxy.invoke([JdkDynamicAopProxy.java:204] ~[spring-aop-4.3.7.RELEASE.jar:4.3.7.RELEASE])
at com.czzp.controller.MybatisController$$EnhancerBySpringCGLIB$$ec4eebf3.createUserByMap([generated] ~[classes/:na] +14 internal calls)
at javax.servlet.http.HttpServlet.service([HttpServlet.java:689] ~[tomcat-embed-core-8.5.11.jar:8.5.11] <1 internal call)
at javax.servlet.http.HttpServlet.service([HttpServlet.java:729] ~[tomcat-embed-core-8.5.11.jar:8.5.11] <1 internal call)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter([ApplicationFilterChain.java:228] ~[tomcat-embed-core-8.5.11.jar:8.5.11])

```

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

可以看到，报错的内容是返回为空，其实这时候数据库已经插入了该用户的信息，只是因为我们的返回值是void所以得不到一个正确的回报文。

The screenshot shows the MySQL database table 'user' with the following data:

user_id	tel	password	nickname	secret	portrait	l_card	area	create_time	update_time
2	13311111111	670b14728ad99	13311111111		[Null]	0	2017-04-22 21:0	2017-04-22 21:0	
3	13322222222	96e79218965eb7	13322222222		[Null]	0	2017-04-22 21:0	2017-04-22 21:0	
4	13333333333	e3ce05881a0a1fc	13333333333		[Null]	0	2017-04-22 21:0	2017-04-22 21:0	
5	13344444444	1a100d2c0da019	13344444444		[Null]	0	2017-04-22 21:0	2017-04-22 21:0	
6	13355555555	5b1b68a9abf4d2	13355555555		[Null]	0	2017-04-22 21:0	2017-04-22 21:0	
7	13366666666	21218ca77804d	18888888888		[Null]	0	2017-04-22 21:0	2017-04-22 21:0	
8	18888888888	5269e3a57331c	19999999999		[Null]	0	2017-05-01 21:3	2017-05-01 21:3	
9	19999999999	14ee22eaba2879	19999999999		[Null]	0	2017-05-01 21:3	2017-05-01 21:3	
10	19999999999	827ccb0ee8a70	13800000000		[Null]	0	2017-05-01 21:4	2017-05-01 21:4	
13	13800000000	827ccb0ee8a70	13800000000		[Null]	0	2017-05-13 23:1	2017-05-13 23:1	

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

我们按照上一次讲的统一结果管理来重写下我们的controller层：

```

1
2     @Autowired
3     private MybatisService service;
4

```

```

5    @Autowired
6    private ExceptionHandle handle;
7
8    /**
9     * 更新用户信息
10    * @param user_id 用户ID
11    * @param nickName 昵称
12    */
13   @PutMapping("/updateUser/{id}")
14   public Result updateUser(@PathVariable("id") String user_id, @RequestParam("nickName") St
15   Result result = ResultUtil.success();
16   try {
17       service.updateUser(user_id,nickName);
18   }catch (Exception e){
19       result = handle.exceptionGet(e);
20   }
21   return result;
22 //    service.updateUser(user_id,nickName);
23 }
24
25 /**
26  * 获取用户信息
27  * @param id 用户ID
28  * @return
29  */
30 @GetMapping("/getUser/{id}")
31 public Result getUser(@PathVariable("id") Integer id){
32     Result result = ResultUtil.success();
33     try {
34         result.setData(service.getUser(id));
35     }catch (Exception e){
36         result = handle.exceptionGet(e);
37     }
38     return result;
39 //    return service.getUser(id);
40 }
41
42 /**
43  * 删除用户
44  * @param tel
45  */
46 @DeleteMapping("/deleteUserByUserId/{tel}")
47 public Result deleteUserByUserId(@PathVariable("tel") String tel){
48     Result result = ResultUtil.success();
49     try {
50         UserInfo user = new UserInfo();
51         user.setTel(tel);
52         service.deleteUserByUserId(user);
53     }catch (Exception e){
54         result = handle.exceptionGet(e);
55     }
56     return result;
57 //    UserInfo user = new UserInfo();
58 //    user.setTel(tel);
59 //    service.deleteUserByUserId(user);
60 }
61
62 /**
63  * 使用@RequestBody获取参数，用map类型接收，再取出
64  * @param reqMap
65  */
66 @PostMapping("/createUserByMap")
67 public Result createUserByMap(@RequestBody Map<String, Object> reqMap){
68     Result result = ResultUtil.success();
69     try {
70         service.createUser(reqMap);
71     }catch (Exception e){
72         result = handle.exceptionGet(e);
73     }
74     return result;
75 //    service.createUser(reqMap);
76 }

```

我们再执行就可以看到返回结果为success了：

POST <http://localhost:9090/mybatis/createUserByMap>

Authorization Headers (1) Body **Body** Pre-request Script Tests

form-data x-www-form-urlencoded raw binary **JSON (application/json)**

```
1+ {  
2   "tel": "13999999999",  
3   "pwd": "12345"  
4 }
```

Status: 200 OK Time: 339 ms Size: 170 B

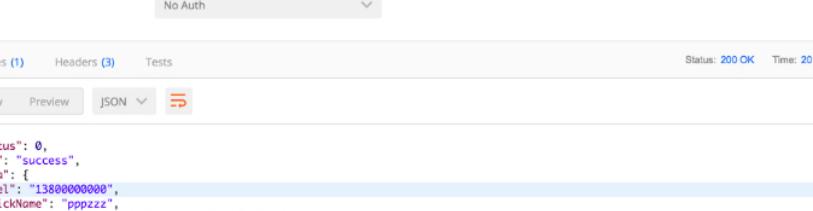
Body Cookies (1) Headers (3) Tests

Pretty Raw Preview JSON

```
1+ {  
2   "status": 0,  
3   "msg": "success",  
4   "data": null  
5 }
```

```
2017-05-13 23:48:57.791 INFO 3218 --- [ main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.Handler]
2017-05-13 23:48:58.165 INFO 3218 --- [ main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
2017-05-13 23:48:58.118 INFO 3218 --- [ main] o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'dataSource' has been auto-detected for JMX exposure
2017-05-13 23:48:58.114 INFO 3218 --- [ main] o.s.j.e.a.AnnotationMBeanExporter : Located MBean 'dataSource': registering with JMX server as MBean [com.alibaba.druid
...poolName,dataSource,type=RuIdbDataSource]
2017-05-13 23:48:58.114 INFO 3218 --- [ main] o.s.j.e.a.AnnotationMBeanExporter : Started beans for JMX exposure on port(s): 8080 (http)
2017-05-13 23:48:58.155 INFO 3218 --- [ main] com.alibaba.druid.support.Plugin : druid init success
2017-05-13 23:48:58.155 INFO 3218 --- [nio-9999-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring FrameworkServlet 'dispatcherServlet'
2017-05-13 23:48:58.155 INFO 3218 --- [nio-9999-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization started
2017-05-13 23:48:58.155 INFO 3218 --- [nio-9999-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization completed in 15 ms
2017-05-13 23:48:58.155 INFO 3218 --- [nio-9999-exec-1] o.s.web.servlet.DispatcherServlet : [/druid/druidMetricsPush/metrics/reloadByHttp]
2017-05-13 23:48:58.155 INFO 3218 --- [nio-9999-exec-1] o.s.web.servlet.DispatcherServlet : method=POST
2017-05-13 23:48:58.155 INFO 3218 --- [nio-9999-exec-1] o.s.web.servlet.DispatcherServlet : id@0:0@0:0@0:1
2017-05-13 23:48:58.155 INFO 3218 --- [nio-9999-exec-1] o.s.web.servlet.DispatcherServlet : class_method=com.zpp.controller.MybatisController:createUserByMap
2017-05-13 23:48:58.155 INFO 3218 --- [nio-9999-exec-1] o.s.web.servlet.DispatcherServlet : @RequestWrapper[createUserByMap,pwd=12345]
2017-05-13 23:48:58.155 INFO 3218 --- [nio-9999-exec-1] o.s.web.servlet.DispatcherServlet : [MybatisController] init
2017-05-13 23:48:58.155 INFO 3218 --- [nio-9999-exec-1] o.s.web.servlet.DispatcherServlet : Set May 13 23:45 2017 MDT! Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.4+ requirements SSL connection must be established if default or explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertificate property is set to 'false'. You can also set it explicitly to ignoreSSL by setting useSSL=false true and provide truststore or verifierStore certificate verification.
2017-05-13 23:48:58.471 INFO 3218 --- [nio-9999-exec-1] com.zpp.util.HttpAspect : responseResult={status=0, msg='success', data=null}; http://blog.csdn.net/qq_31001665
```

其中特别获取数据也是没问题的，封装在我们之前定义的date中了，另外值得说一下的是我把delete方法中的条件改为tel了，大家测试的时候别继续传id了哦～～～



GET <http://localhost:9090/mybatis/getUser/13>

Authorization Headers (1) Body Pre-request Script Tests Params Send Save Cookies Code

Type No Auth

Body Cookies (1) Headers (3) Tests Status: 200 OK Time: 20 ms Size: 253 B

Pretty Raw Preview JSON ↻

```
1+ {  
2   "status": 0,  
3   "msg": "success",  
4+   "data": {  
5     "tel": "13800000000",  
6     "nickName": "pppzzz",  
7     "passWord": "827ccb0eea8a706c4c34a16891f84e7b"  
8   }  
9 }
```

贴士：

使用注解配置时返回的结果如果是一个对象且对象名和数据库字段不统一时我们可以使用@Results和@Result来绑定属性，也可以在SQL中使用 AS 属性名来绑定对象属性。

#### 使用XML配置：

使用XML配置Mybatis是我们从SpringMVC就开始用的一种方式，也是我比较推荐的方式，因为注解配置虽然方便，但不是很利于日后的维护，而且如果其他人修改你的代码，一堆SQL很难发现问题，比如一个半角逗号写成全角逗号.....

如果你和我一样使用IDEA作为开发工具的话，需要把XML放在resources下，因为IDEA编译是在resources下作为静态文件的，如果你写在java目录下，运行时是会报Invalid bound statement (not found)找不到绑定SQL的错误的。当然，我们其实还是有办法来改变IDEA的这个习惯的，我们稍等一会儿来谈这个问题。

我们还是把之前的controller, service, dao都复制一遍（我真的很懒.....），只是指向XML哦地址的dao层而已，再创建一个XML文件放在resources下的mapper目录，其中写我们的SQL：

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-
3 <mapper namespace="com.zzp.dao.impl.MybatisUserXMLMapper">
4
5     <insert id="createUser" parameterType="java.util.Map">
6         INSERT INTO blog_tp_user(tel,password,nickname,secret) VALUES (#{tel},md5(#{pwd}),#{t
7     </insert>
8
9     <update id="updateUser">
10        UPDATE blog_tp_user SET nickname = #{nickname} WHERE user_id = #{userId}
11    </update>
12
13    <select id="getUser" resultType="com.zzp.pojo.UserInfo">
14        SELECT tel,nickname,password FROM blog_tp_user WHERE user_id = #{id}
15    </select>
16
17    <delete id="deleteUserByUserId">
18        DELETE FROM blog_tp_user WHERE tel = #{tel}
19    </delete>
20
21 </mapper>
```

关于namespace以及XML的文件名等细节我就不具体说了，和SpringMVC中是一致的。我们完成后测试一下发现报错了，找不到绑定的SQL，这就该说下我们关于Mybatis的一些配置了，如果你不想在运行期间有那么多乱七八糟的错误的话，最好就是按照我的配置来做以下的更改：

1.在yml中写入xml所在位置，智利我是放在resources目录下的：

```
1 mybatis:
2   mapper-locations: classpath:mapper/*.xml
```

2.在启动类上加入需要扫描的dao层

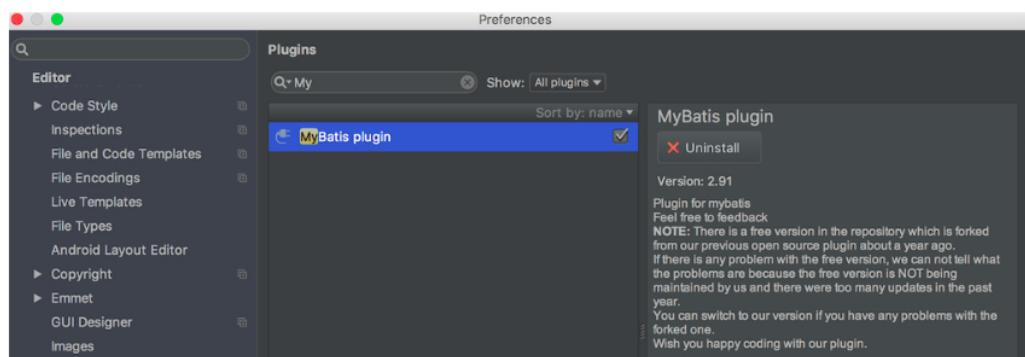
```
1 @MapperScan(basePackages = "com.zzp.dao")
```

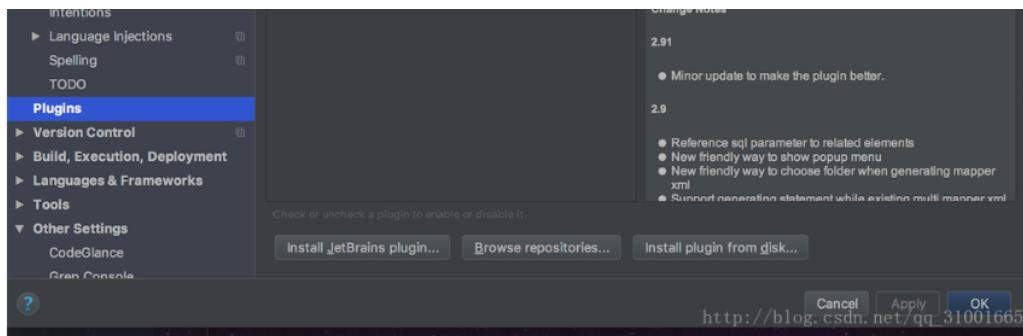
3.给你的dao层加入@Repository注解，让Spring能够扫描到

完成以上三点后我们再一次测试，就可以得到正常结果啦~~~

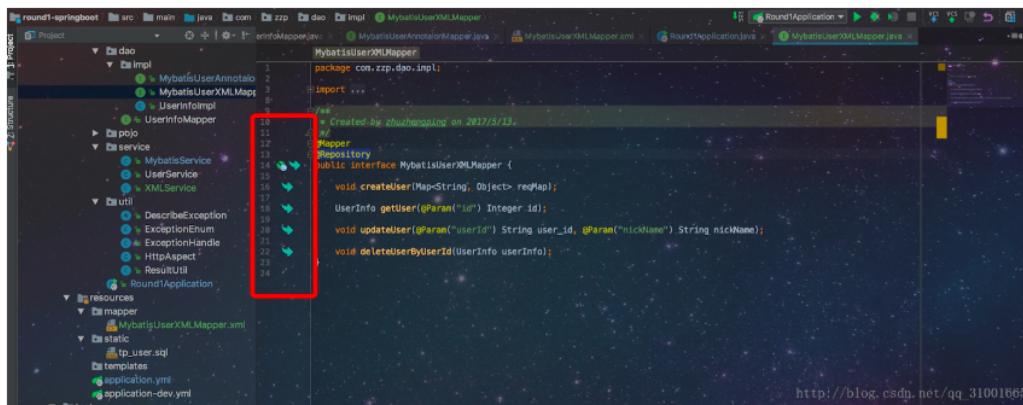
彩蛋：

在XML中，我们无法直接从dao层进去XML中SQL语句中，但是如果你使用IDEA的话，就可以通过插件实现这一功能，在IDEA插件中搜索Mybatis就有啦：





安装后就可以直接进入SQL啦，并且还会提示你该方法写没写SQL。



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/71914945](https://blog.csdn.net/qq_31001665/article/details/71914945)

文章标签：[mybatis](#) [spring](#) [spring-boot](#)

个人分类：[spring boot](#)



大数据：2018最有“钱”力股，你知道如何下手吗？

升职难、涨薪难还是担心被淘汰，老程序员手把手教你学大数据，争做稀缺人才，增强核心竞争力.....

[查看更多>>](#)

想对作者说点什么？ [我来说一句](#)

ian0410 2018-02-20 17:04:01 #1楼

[查看回复\(1\)](#)

有个小问题 在第一次请求的那个controller 传参的时候是不是不可以用@RequestBody? 会报  
org.springframework.web.HttpMediaTypeNotSupportedException: Content type 'application/x-www-form-urlencoded;charset=UTF-8' not supported 用 @RequestParam 就报空指针，就正常了

[上一页](#) 1 [下一页](#)

如何快速搭建Springboot项目1.访问<http://start.spring.io>2.选择Maven Project，然后选择Java，然后选择Springboot版本，最后点击Generate...

 liu455042806 2018-04-09 14:39:48 阅读数: 264

## Spring boot Mybatis 整合（完整版）

对Mybatis注解方式有兴趣的，可以查看我的另一篇：[Spring boot Mybatis 整合（注解版）](#) 注解方式更加简洁简单，极大程度的提高了开发速度。如果大家需要使用更高的版本比如s...

 Winter\_chen001 2017-08-16 15:59:47 阅读数: 161759

## 最全面的springBoot集成mybatis+mysql项目搭建经验分享

大家都知道，现在Spring框架几乎无处不用，目前最新版本据说是Spring5，现在常用的基本还是Spring4.X，很多公司甚至用的还是Spring3.x，而Spring领域其中最好用的莫过于Spr...

 guobinhui 2018-02-08 15:56:23 阅读数: 752

## springboot入门（三）-- springboot集成mybatis及mybatis generator工具使用

mybatis是一个半自动化的orm框架，所谓半自动化就是mybatis只支持数据库查出的数据映射到pojo类上，而实体到数据库的映射需要自己编写sql语句实现，相较于hibernate这种完全自动...

 travellersY 2017-11-28 13:58:39 阅读数: 6118

## maven多模块，springboot整合mybatis

第一次写博客，嘴下留情。项目架构： 模块依赖关系： web 层依赖service+util， service层依赖dao+utils， dao层依赖entity...

 hardy\_programmer 2018-01-30 18:31:40 阅读数: 1398

## Springboot 整合 Mybatis 的完整 Web 案例

原文首发于bysocket新年第一篇 Springboot 技术文诞生。泥瓦匠准备写写 Springboot 相关最佳实践。一方面总结下一些 Springboot 相关，一方面和大家交流交流 Spr...

 hliq5399 2017-08-31 16:42:16 阅读数: 1567

## springboot集成mybatis实例

介绍下springboot集成mybatis的使用 (1)在pom.xml添加springboot依赖; (2)在application.properties文件中配置mysql连接配置文件 (3)在...

 abccchengxuyuan 2017-10-02 23:53:31 阅读数: 618

## spring boot(六)：如何优雅的使用mybatis

这两天启动了一个新项目因为项目组成员一直都使用的是mybatis，虽然个人比较喜欢jpa这种极简的模式，但是为了项目保持统一性技术选型还是定了 mybatis。到网上找了一下关于spring boot...

 gebitan505 2017-02-08 17:07:01 阅读数: 98839

## spring-boot 集成mybatis配置使用

基本照搬了mybatis-spring-boot-samples中的代码，先上手。添加maven依赖 在pom文件中增加如下内容 org.mybatis.spring.boot ...

 zl18310999566 2017-01-05 14:35:39 阅读数: 17643

## spring-boot+mybatis开发实战：如何在spring-boot中使用mybatis持久层框架

前言： 本项目基于maven构建，使用mybatis-spring-boot作为spring-boot项目的持久层框架 spring-boot中使用mybatis持久层框架与原spring项目使用...

 eguid\_1 2016-09-20 23:07:12 阅读数: 7738

## Spring Boot 集成MyBatis

Spring Boot 集成MyBatis在配置MyBatis前，我们先配置一个druid数据源。Spring Boot 集成druid有很多个配置选项，使用Spring Boot的Con...

isea533 2015-12-27 15:29:01 阅读数: 309540

## Spring Boot之springboot+mysql+mybatis搭建

创建数据库及数据表: CREATE DATABASE `springbootdb` DEFAULT CHARACTER SET utf8;

u012343297 2017-12-18 15:39:17 阅读数: 2040

## Spring boot Mybatis

最近刚接触Spring boot, 正是因为他的及简配置方便开发, 促使我下定决心要用它把之前写的项目重构, 那么问题来了, spring boot怎么整合mybatis呢, 下面几个配置类来搞定。1、文件结...

xiaoyu411502 2015-09-01 22:01:44 阅读数: 115194

## 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



## Spring Boot MyBatis 连接数据库

最近比较忙, 没来得及抽时间把MyBatis的集成发出来, 其实mybatis官网在2015年11月底就已经发布了对SpringBoot集成的Release版本, Github上有代码: <https://github.com>...

catoop 2016-01-21 10:50:21 阅读数: 67192

## Spring Boot中使用MyBatis注解配置详解

之前在Spring Boot中整合MyBatis时, 采用了注解的配置方式, 相信很多人还是比较喜欢这种优雅的方式的, 也收到不少读者朋友的反馈和问题, 主要集中于针对各种场景下注解如何使用, 下面就对几种常见...

qq\_26562641 2016-11-25 11:13:13 阅读数: 2860

## java 搭建基于springboot的ssm(spring + springmvc + mybatis)的maven项目

今天我们来搭建一个简单的基于springboot的ssm(spring + springmvc + mybatis)的maven项目, 使用的数据库为mysql。好了废话不多说。开始搞起。最终项目目录...

libo yang71 2017-06-19 11:42:19 阅读数: 20187

## springboot+spring+springmvc+mybatis详解

再三注意: 在阅读本文之前务必配置Gradle环境。gradle倒入jar包教程 在开始之前。我认为有必要提醒。你需要先点击FILE->NEW->OTHERS->Gradle project...

qq\_36068028 2017-08-02 10:01:08 阅读数: 2131

## SpringBoot学习笔记(4) Spring Boot 集成 Mybatis

Mybatis 官方提供了 [mybatis-spring-boot-starter](https://github.com/mybatis/spring-boot-starter) <http://www...>

a67474506 2016-07-24 14:14:51 阅读数: 51122

## 高端网站设计

高端网站美工设计 网站定制开发

百度广告



## SpringBoot (五) : SpringBoot整合MyBatis

怎么说了, 写博客虽然是一件很费时间的事情, 而且还是个菜鸟, 但是如果写的东西能够帮助到别人, 还是值得开心的。回顾: 上篇写了JdbcTemplate, 但是想到使用Mybatis, JPA的人估计不...

saytime 2017-07-08 00:51:17 阅读数: 32967

## spring boot配置mybatis和事务管理

Spring Boot配置Mybatis一、spring boot与mybatis的配置1.首先, spring boot 配置mybatis需要的全部依赖如下: org.springframework...

wohaqiyi 2017-06-07 08:37:14 阅读数: 47285

## spring boot 整合mybatis 通过官方mybatis-spring-boot-starter