

微软MSDN限时大促

2年期

CSDN 软件商城

Visual Studio Professional

with MSDN

直降¥5000

还送Xbox 限量50套!



C 博客

登录 | 注册



赤山岛

洞庭湖上的一颗小岛，钓鱼，养猫，学程序。（Java，设计，Web）

目录视图

摘要视图

RSS 订阅

个人资料



tanxiang21

+ 加关注 发私信



访问：156564次

积分：2715

等级：BLOG > 5

排名：第8927名

原创：97篇 转载：6篇

译文：14篇 评论：14条

文章搜索

博客专栏



Java初级码农

文章：21篇

阅读：12175

文章分类

Java (67)

Hibernate (9)

Struts (7)

UML (6)

模式 (5)

Spring (3)

IT (2)

Ant (1)

【专家图书】蔡俊鸿翻译的Unity3D书籍上市啦 【知识库】Swift资源大集合 【公告】博客新皮肤上线啦 快来领福利：C币、机械键盘

原 Java注解（自定义方法注解使用及测试）

标签：Java 注解 自定义方法注解

2013-05-11 15:00 9438人阅读 评论(0) 收藏 举报

分类：Java (66)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目标，创建一个方法注解，我们能够初始化时执行该方法，并可以给该方法传入注解的参数值
假设我们设计一个 sayHello(String name) 方法，给该方法做个注解，初始时使用注解传入"小明"

```
[java]
01. package annotation;
02.
03. public class HelloWorldStub {
04.
05.     @HelloWorldAnnotation(name = "小明")
06.     public String sayHello(String name) {
07.         if (name == null) {
08.             name = "";
09.         }
10.         return name + " say hello world!";
11.     }
12. }
```

定义注解

```
[java]
01. package annotation;
02.
03. import java.lang.annotation.Documented;
04. import java.lang.annotation.ElementType;
05. import java.lang.annotation.Retention;
06. import java.lang.annotation.RetentionPolicy;
07. import java.lang.annotation.Target;
08.
09. @Retention(RetentionPolicy.RUNTIME)//注解会在class中存在，运行时可通过反射获取
10. @Target(ElementType.METHOD)//目标是方法
11. @Documented//文档生成时，该注解将被包含在javadoc中，可去掉
12. public @interface HelloWorldAnnotation {
13.
14.     public String name() default "";
```

系统构建 (3)
JS (1)
FreeMarker (2)
Jsp (1)
JVM (3)
IDE (1)
数据库 (14)
项目优化 (12)
Maven (2)
测试 (1)

阅读排行

【问题处理】Plugin exe (9863)
Java注解（自定义方法注 (9428)
基于jQuery省市县联动插 (9265)
UML学习笔记（三）【序 (7549)
Maven（一）Maven及插 (6777)
Tomcat与Jetty插件 配置 (5680)
【数据库摘要】10_Sql_ (3163)
FreeMarker初级攻略 (2964)
Hibernate学习笔记（三） (2615)
【ThreadLocal模式】线 (2314)

```
15.
16. }
```

我们的测试类

```
[java]
01. package annotation;
02.
03. import java.lang.reflect.InvocationTargetException;
04. import org.junit.Test;
05. import static org.junit.Assert.assertEquals;
06.
07. public class TestHelloWorldAnnotation {
08.
09.     @Test
10.     public void testHello() throws IllegalArgumentException,
11.         IllegalAccessException, InvocationTargetException,
12.         SecurityException, NoSuchMethodException, InstantiationException {
13.
14.         //定义操作类
15.         ParseAnnotationStub parse = new ParseAnnotationStub();
16.
17.         //假设我们知道类HelloWorldStub使用了注解，执行HelloWorldStub中带注解的方法
18.         //判断是否使用了注解的name()方法，设置name="小明"，并返回"小明 say hello world!"
19.         String returnValue = parse.parseMethod(HelloWorldStub.class);
20.         assertEquals("小明 say hello world!", returnValue);
21.
22.     }
23. }
```

执行的注解方法的操作类

```
[java]
01. package annotation;
02.
03. import java.lang.reflect.InvocationTargetException;
04. import java.lang.reflect.Method;
05.
06. public class ParseAnnotationStub {
07.
08.     //包装了下Java基本的方法反射（范围是带了我们特定注解的方法）
09.     //传入我们要执行的类型，所以我们时常发现某些框架要我们定义好类查找的范围，或前后缀什么的
10.     //可以设置返回值为空void或者Object通用，这里我们为了测试采用String返回值
11.     public String parseMethod(Class<?> clazz) throws IllegalArgumentException,
12.         IllegalAccessException, InvocationTargetException,
13.         SecurityException, NoSuchMethodException, InstantiationException {
14.
15.         //获得该对象
16.         Object obj = clazz.getConstructor(new Class[]{}).newInstance(new Object[]{});
17.
18.         //变量该对象的方法
19.         for (Method method : clazz.getDeclaredMethods()) {
20.
21.             //获取方法的注解，这里特定获取方法上@HelloWorld注解
22.             HelloWorldAnnotation say = method.getAnnotation(HelloWorldAnnotation.class);
23.
24.             //如果@HelloWorld注解不空，即方法有@HelloWorld注解
25.             if (say != null) {
26.
27.                 //这里我们先定义了 UseHelloWorld.sayHello(String name)方法
28.                 //这里可以从注解中获取值，或者直接运行，或者缓存该对象方法
29.                 String name = say.name();
30.                 return (String)method.invoke(obj, name);
31.             }
32.
33.         }
34.         return "";
35.     }
36. }
```



顶
0

踩
0

^ 上一篇 JAVA泛型（代码级解释）

▼ 下一篇 【Struts2源码】@Inject查找并缓存注入器

我的同类文章

Java (66)

- | | | | | | |
|------------------------------|------------|--------|-----------------------------|------------|---------|
| • 要点Java20 java.util.Coll... | 2014-05-26 | 阅读 842 | • 要点Java19 java.util.Arrays | 2014-05-21 | 阅读 489 |
| • 要点Java18 java.util.Form.. | 2014-04-18 | 阅读 487 | • 要点Java17 String | 2014-04-18 | 阅读 623 |
| • 【重构】摘要1 | 2014-03-29 | 阅读 355 | • 要点Java16 抽象类 | 2014-03-26 | 阅读 533 |
| • 关于合作编写Java初级教程 | 2014-03-25 | 阅读 571 | • 要点Java15 异常 | 2014-03-25 | 阅读 494 |
| • 要点Java14 捕获异常 | 2014-03-24 | 阅读 674 | • 要点Java13 继承Inheritan.. | 2014-03-20 | |
| • 要点Java12 SDK的目录结.. | 2014-03-14 | 阅读 918 | | | 阅读 1215 |

猜你在找

- | | |
|----------------------|-------------------------------|
| ▪ Java之路 | ▪ java 一自定义注解的使用 |
| ▪ 软件测试基础 | ▪ Java中的Annotation详解和使用 自定义注解 |
| ▪ 零基础学Java系列从入门到精通 | ▪ Java自定义注解Annotation的使用 |
| ▪ 移动手机APP测试从零开始（初级篇） | ▪ java使用自定义注解 |
| ▪ HTML 5移动开发从入门到精通 | ▪ Java自定义注解的使用 |



智能停车场



登陆界面设计



二手福特野马



酒店收银系统



一年拿本科



酒店式公寓



德云社

查看评论

暂无评论

您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点,不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker
OpenStack VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC
WAP jQuery BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora
XML LBS Unity Splashtop UML components Windows Mobile Rails QEMU KDE
Cassandra CloudStack FTC coremail OPhone
Web App SpringSide Maemo Compuware 大数据
Hibernate ThinkPHP HBase Pure Solr Angular
Bootstrap



公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技

江苏乐知网络技术有限公司 提供商务支持

京ICP证09002463号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved