

拦截器使用注解方式校验参数

William_Cheung 阅读数：7896

原创文章，未经博主允许不得转载。 <https://blog.csdn.net/qq496013218/article/details/78677788>

方式去拦截校验参数，本章讲解使用拦截器去校验参数，以及遇到的问题。

，就是案例所用到的去校验参数，类似与servlet开发中里的filter过滤器。用于对拦截前及后处理。

等等。比喻我们在学习jdbc的时候，获取连接，最后关闭连接。其实本质也是AOP的方法（面向切面编程），也就是说符合AOP的横向切入点的功能都可以使用拦截器去实现。

```
HandlerInterceptor {

    reHandle(HttpServletRequest request, HttpServletResponse response, Object handler)
        throws Exception;

    Handle(
        HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView)
        throws Exception;

    rCompletion(
        HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex)
        throws Exception;
```

实现拦截器的预处理，第三个参数为响应的处理器；返回值：true表示继续流程；false表示流程中断（如校验失败，参数缺少），不会继续调用其他的拦截器或处理器，此时我们需要

实现处理拦截器的后处理（返回请求结果之前）。
理完毕回调方法，在返回结果之后进行调用，但仅调用处理器执行链中preHandle返回true的拦截器的afterCompletion。

的时候只需要实现某个方法，或者不需要去实现全部的方法，可以使用拦截器的适配器类（使用了一种适配器模式的设计模式），允许我们只实现需要的方法。

```
lass HandlerInterceptorAdapter implements AsyncHandlerInterceptor {

    mplementation always returns {@code true}.

    olean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)
        hrows Exception {
        eturn true;

    mplementation is empty.
```

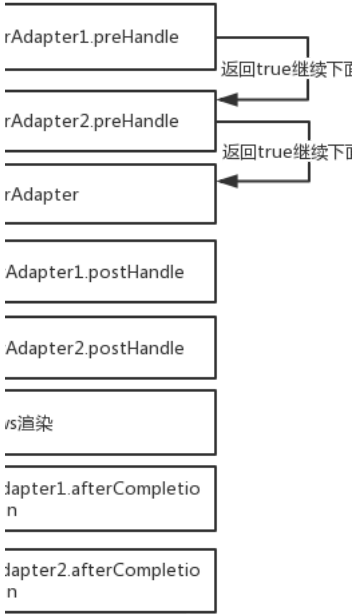
```
id postHandle(17 |
    HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView)18 |
    throws Exception {19 |
        }
```

Implementation is empty.

```
id afterCompletion(
    HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex)
    throws Exception {
```

Implementation is empty. 这个方法实现自AsyncHandlerInterceptor

```
id afterConcurrentHandlingStarted(
    HttpServletRequest request, HttpServletResponse response, Object handler)
    throws Exception {
```



blog.csdn.net/qq496013218

~

💬

📄

🔖

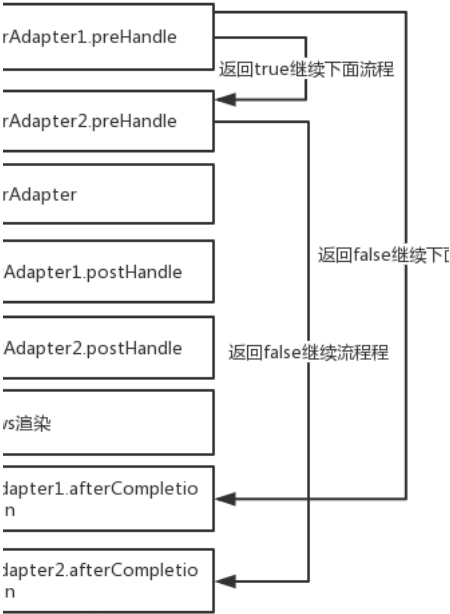
📱

<

>

👍

0



<http://blog.csdn.net/qq496013218>

g配置文件中定义的顺序而定的。
执行所有拦截器的preHandle方法，一直遇到return false为止，比如第二个preHandle方法是return false，则第三个以及以后所有拦截器都不会执行。若都是return true，则按配置的拦截器顺序依次执行。
（若中间抛出异常，则跟return false效果一致，不会继续执行postHandle，只会倒序执行afterCompletion方法。
（页面还未渲染数据）时，按倒序执行postHandle方法。若第三个拦截器的preHandle方法return false，则会执行第二个和第一个的postHandle方法和afterCompletion（postHandle都渲染数据后，执行after进行清理工作）方法。（postHandle和afterCompletion都是倒序执行）

0:

一样，可以忽略：

```
ionPolicy.RUNTIME)  
pe.METHOD)  
    CheckParam {
```

所需要的参数, 多个以小写的逗号隔开

```
ames() default "";
```

类型

```
meter() default Object.class;
```

数;

```
re() default false;
```

Helper {

```
String getBodyString(HttpServletRequest request) throws IOException {
```

~

💬

📄

🔖

📱

<

>

👍

0

```

lder sb = new StringBuilder();
5 |
Reader reader = new BufferedReader(new InputStreamReader(request.getInputStream(), Charset.forName("UTF-8"))) {
6 |
ine = "";
7 | while ((line = reader.readLine()) != null) {
b.append(line);

.toString();

```

```

HandlerInterceptor extends HandlerInterceptorAdapter {

c final Logger LOGGER = Logger.getLogger(OpenHandlerInterceptor.class);

tic final String APPLICATION_CHARSET="application/json; charset=utf-8";

tic final String UTF_8 = "UTF-8";

n preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {

ndler instanceof HandlerMethod){
andlerMethod handlerMethod = (HandlerMethod) handler;
ethod method = handlerMethod.getMethod();
heckParam checkParam = method.getAnnotation(CheckParam.class);
f(checkParam == null || !checkParam.require() || StringUtils.isEmpty(checkParam.fieldNames())) {
    return true;

/防止中文, 符号
tring fieldNames = checkParam.fieldNames().replace(", ", ",");
olean jsonParam = CheckFieldsNotNull(request, fieldNames);
f(!jsonParam){
    ErrorMsg(response);
    return false;

Exception e) {
ntStackTrace();
Msg(response);
n false;

ue;

```

属性是否存在

uest

ldNames

ception

```

boolean CheckFieldsNotNull(HttpServletRequest request, String fieldNames) throws Exception{
fo(String.format("校验字段: {s}", fieldNames));
questParam = HttpHelper.getBodyString(request);
Utils.isEmpty(requestParam)){
n false;

= new Gson();
<Map> arrayList = gson.fromJson(requestParam, ArrayList.class);
ng fieldName : fieldNames.split(",") {
nditions.checkNotNull(arrayList.get(0).get(fieldName), fieldName + "is null");

```

```
ue;54 | }
```

```
ponse  
Exception
```

```
void errorMsg( HttpServletResponse response) throws Exception {  
    PrintWriter out = response.getWriter();  
    out.setCharacterEncoding(UTF_8);  
    out.setContentType(APPLICATION_JSON);  
    out.print(JsonMapper.toJson(Result.fail(ErrorMessageEnum.MISS_REQUEST_PARAM.getValue(), ErrorMessageEnum.MISS_REQUEST_PARAM.getMessage(), esc()))));  
}
```

校验，但是从中遇到一个问题，流只能读一次。所以我们在controller再次获取的是null值，为了解决请求中的流多次复用，所以实现了一个Filter，通过Filter来解决流只能读取一次的问题。

```
class RequestWrapper extends HttpServletRequestWrapper {  
    private byte[] body;  
  
    RequestWrapper(HttpServletRequest request) throws IOException {  
        super(request);  
        body = RequestHelper.getBodyString(request).getBytes(Charset.forName(OpenHandlerInterceptor.UTF_8));  
    }  
  
    @Override  
    public BufferedReader getReader() throws IOException {  
        return new BufferedReader(new InputStreamReader(getInputStream()));  
    }  
  
    @Override  
    public InputStream getInputStream() throws IOException {  
        ByteArrayInputStream bais = new ByteArrayInputStream(body);  
        return new ServletInputStream() {  
            @Override  
            public int read() throws IOException {  
                return bais.read();  
            }  
  
            @Override  
            public boolean isFinished() {  
                return false;  
            }  
  
            @Override  
            public boolean isReady() {  
                return false;  
            }  
  
            @Override  
            public void setReadListener(ReadListener readListener) {}  
        };  
    }  
}
```

pper

一个包装类，大家应该知道装饰器模式。这个类对接下来过滤器中的使用是有非常重要的意义的。
法再次读取，所以通过Filter的重写回写请求对象中；使请求中的流多次复用。
的实现使用了责任链设计模式：

```
ServletRequestReplacedFilter implements Filter {

    destroy() {

    }

    doFilter(ServletRequest request, ServletResponse response,
              FilterChain chain) throws IOException, ServletException {
        request requestWrapper = null;
        if (request instanceof HttpServletRequest) {
            requestWrapper = new RequestReaderHttpServletRequestWrapper((HttpServletRequest) request);
        }

        if (requestWrapper != null) {
            // 将取出来的字符串，再次转换成流，然后把它放入到新request对象中。
            // 1.doFilter方法中传递新的request对象
            doFilter(request, response);
        } else {
            doFilter(requestWrapper, response);
        }
    }

    init(FilterConfig arg0) throws ServletException {
    }
}
```

的HttpServletRequestWrapper类，重写相关ServletRequest方法，实现流的多次读取。
中的流放入ThreadLocal，然后在需要请求参数的地方调用ThreadLocal的get方法即可。

盘手的肺腑之言，写给无数正在亏钱的散户

实现拦截器

阅读数 1059

拦截器就是为了防止用户没有登录而可以访问app、网页... 博文 来自： 燕少江湖

到第八篇：springboot实现自定义拦截器...

阅读数 1223

nbsp;&&&&& &&&... 博文 来自： codecarver

拦截类拦截业务请求，并获取到请求的参数名...

阅读数 2万+

核操作，添加，修改，删除人员，商品时如果需要审... 博文 来自： 刘俊重

拦截器（二）

阅读数 729

拦截器 上篇我们讲到拦截器的使用，在实际的开发工... 博文 来自： 孤狼

5的推广

器实现 阅读数 1万+
接收合作方的通知内容时，需要进行访问控制，即需... 博文 来自： 爱传文档...

实现未登录拦截 阅读数 408
用的是自定义注解，一直没搞懂流程，最近闲来无... 博文 来自： Java_Mrs...

阅读数 649
erceptor;importjava.lang.reflect.Method;importjava.util... 博文

登录 阅读数 2809
登录两种状态的操作，调用后台方法需要进行判定是否... 博文 来自： 不积跬步...

应用 阅读数 687
在代码中的一些元信息，用于一些工具在编译、运行... 博文 来自： xupeng87...

用注解方式校验参数 - CSDN博客

定义注解进行接口鉴权、登录校验..._CSDN博客
式校验参数 12-01 阅读数 7719 前言:上一篇介绍了,使用AOP的方式去拦截校验参...

操盘手的肺腑之言，写给无数正在亏钱的散户
...

权限控制 阅读数 1万+
拦截器实现权限控制，自己花了点时间整理，发到网站... 博文 来自： 越努力，...

+注解实现《登录校验》 - 一切..._CSDN博客
式校验参数- _William_Cheung的博客 12-01 4568 前言:上一篇介绍了,使用AOP的...

器获取url 动态参数 - 培训班的..._CSDN博客
式校验参数 12-01 阅读数 7664 前言:上一篇...博文 来自: _William_Cheung的博客...

拦截器注解 阅读数 1175
者类) 进行权限控制。解决方案在需要权限控制的资... 博文 来自： 艾思祺科...

虑器中使用流读取参数后,controll..._CSDN博客
式校验参数 - _William_Cheung的博客 12-01 4652 前言:上一篇介绍了,使用AOP...

自定义注解做权限管理 - konghen1..._CSDN博客
式校验参数 12-01 4040 前言:上一篇介绍了...来自: _William_Cheung的博客 sprin...

阅读数 511
博文 来自： cst_code...

AOP、拦截器 阅读数 1209
Annotation:注解创建Annotation类1.@Target@Target... 博文 来自： SzeLH的...

虑器中使用流读取参数后,controll..._CSDN博客

拦截器方式实现注解功能拦截器的配..._CSDN博客

~

💬

📖

🔖

📄

<

>

👍

0

式校验参数 12-01 阅读数 7125 前言:上一篇...博文 来自: _William_Cheung的博客...

义注解实现interceptor的指定拦截 阅读数 2969
操作,很多时候我们要采用aop或者interceptor的方式... 博文 来自: Horizon_...

操盘手的肺腑之言,写给无数正在亏钱的散户
i

主解 实现自定义权限验证 - qq_..._CSDN博客
式校验参数 - _William_Cheung的博客 12-01 4302 前言:上一篇介绍了,使用AOP...

导Controller方法名和注解信息(用..._CSDN博客
方式去拦截校验参数,本章讲解使用拦截器去校验参数...博文 来自: _William_Cheu...

功能interceptor,通过注解实现对参数的值... 阅读数 1697
自定义注解,该注解只能作用在方法上@Target(Element... 博文 来自: myth_g的...

拦截器 阅读数 2646
现,在需要的拦截的方法上面加上一个注解@Access... 博文 来自: xwq911的...

控制 阅读数 525
,但是始终都离不开权限控制。一个好的权限控制对... 博文 来自: as846332...

的功能,我们今天就讨论一下利用注解来配置拦截器... 阅读数 549
博文 来自: 杨煜冬煜...

器 加注解精准拦截method 阅读数 5830
,并且是可以自定义哪些具体方法被拦截 1.自定义一... 博文 来自: Flyer的后...

你卸载算我输!爆率9.8,有充值入口我跪键盘!
媒

行拦截器配置 阅读数 9990
羊使用注解拦截action,这在使用中非常不方便,我们来... 博文 来自: wodemay...

实例 阅读数 1万+
.contractReview.action;importjava.util.ArrayList;import... 博文 来自: z6918378...

解的使用 阅读数 934
st/wodemaya8/article/details/45247979springmvc里使... 博文 来自: 肖天翔

拦截器 (转) 阅读数 43
用进行鉴权,如api接口中有个用户唯一标示accessTo... 博文 来自: xp9802的...

页面的访问权限控制 阅读数 15
访问权限控制 在web系统中,经常需要对每个页面... 博文 来自: 海浪儿

战,能让你少些1000行代码! 广告
你的开发效率大幅提升,解放劳动力~

定义注解进行数据拦截实例 阅读数 413
进行数据拦截切面开关配置: spring.aop.auto:truespri... 博文 来自: Blue Sky ...

方式实现脱敏及源码下载地址 阅读数 157
ion="1.0"encoding="UTF-8"?&... 博文 来自: weixin_3...
https://blog.csdn.net/qq496013218/article/details/78677788

~

💬

📖

🔖

📄

<

>

👍

0

Spring+注解实现《登录校验》 阅读数 2万+
应用spring拦截器的第二篇文章，第一篇《解决跨域... 博文 来自： 一切都是...

SpringMVC拦截器 配合使用记录系统操作日志... 阅读数 5304
重,在这里,代码的注释中,我已经详细的介绍了,另外就是... 博文 来自： _予兮

定义注解 阅读数 269
目里加上了用户权限，有些操作需要登录，有些操作... 博文

这些技术必须掌握！ 广告
这些技术等于迈入大厂半条腿...

拦截器方式实现注解功能拦截器的配置 阅读数 2244
e.TYPE,ElementType.METHOD}))@Retention(Retenti... 博文 来自： natural_...

SpringMVC的拦截器 阅读数 2万+
realxt/archive/2009/09/20/spring_mvc_annotation_inte... 博文 来自： Marvin

拦截器详解 阅读数 10万+
SpringMVC，下面讲下他的拦截器。使用SpringMVC... 博文 来自： Java高知

注解做权限判断 阅读数 113
quot;]@Documented@Inherited@Target(ElementType... 博文 来自： jackyrong...

通过注解@RequestBody如何得到http POST请求... 论坛
alue = "repaymentPlanPull", method = RequestMethod.PO...

程序员薪资一路攀高，开发人员如何转向高收入？ 广告
钱工程师挑战高薪！

拦截器 阅读数 123
（请注意后面需要注意的问题）配置文件的配置—spri... 博文 来自： iteye_151...

拦截器无法使用 问答

定义注解 阅读数 8817
目里加上了用户权限，有些操作需要登录，有些操作... 博文 来自： zhuyangx...

请求参数 阅读数 1561
ilue=""test";method=RequestMethod.POST... 博文 来自： litte_frog...

注解的结合使用 阅读数 2370
如spring, hibernate, springMvc, SpringBoot等。使... 博文 来自： fengchao...

阅读数 1万+
P，控制器Controller 负责处理由DispatcherServlet 分... 博文 来自： 一味前行

自定义注解做权限管理 阅读数 3188
说明了Annotation所修饰的对象范围：Annotation可... 博文 来自： konghen1...

~

💬

📖

🔖

📱

<

>

👍

0

参数注入 阅读数 5207
特别便利的一个小功能,那就是在Controller层映射的... 博文 来自: Captain_...

ip拦截器 阅读数 222
rtjava.util.ArrayList;importjavax.servlet.http.HttpServle... 博文 来自: 刘成江的...

器 阅读数 49
解; (2) 基于xml (schema) 。公司要求使用xml方... 博文 来自: hw12877...

实现以及拦截器无效的解决办法 阅读数 1288
要我们自己去实现拦截器, 比如判断用户是否登录, ... 博文 来自: just coding

ceptor(登录拦截器)无法注入bean 阅读数 1万+
拦截器)无法注入bean的问题.源码结构图: LoginInter... 博文 来自: mjifto的博客

拦截器浅谈 阅读数 2980
: (1)修饰属性、方法、构造函数 @Autowire:自动... 博文 来自: 孟凡霄

拦截器 阅读数 1080
篇文章, 在很多时候, 我们往往通过注解定义自己的... 博文 来自: qq_3601...

第三个参数handler 阅读数 1668
, 但是之前一直没有关注过第三个参数Objecthandler... 博文 来自: Donggua...

器中的参数传到Contraoller中让它自动封装参数值
indle方法 中可以将请求参数重新定义吗 去请求接口, 让C... 论坛

阅读数 7733
结合推荐对象间关联关系的社会化推荐算法》的笔记 (... 博文 来自: 雨石

阅读数 4万+
vv.video.qq.com/geturl?vid=v00149uf4ir&otype=json ... 博文 来自: 专注于互...

服务器80端口 阅读数 2万+
(本地服务计算机) ubuntu 14.04 LTS(阿里云服务器) ... 博文 来自: Anteoy的...

IL.Image模块(ios android icon图标自动... 阅读数 10万+
tp://www.pythonware.com/products/pil/ 2.解压后, 进... 博文 来自: 专注于co...

发实践 阅读数 30万+
存 媒体 其他组织换句话讲就是不让个人开发者注册。 ... 博文 来自: 小雨同学...

阅读数 4389
是通过下载exe格式的安装包来安装使用, 今天偶然下... 博文 来自: fzhmoive...

作原理是什么? 阅读数 5万+
:接的缓冲池技术, 缓冲池中的连接可以被任何需要他... 博文 来自: Napoleon...

阅读数 6295
(1).进入Python 下载官网: https://www.python.org/g... 博文 来自: qq_3627...

linux安装nginx 阅读数 7万+
ie x) 是一个高性能的HTTP和反向代理服务器，也是一... 博文 来自: maoyuan...

简单的安装方法) 阅读数 26万+
个月了，作为实习狗的我，被叫去研究Docker了，汗... 博文 来自: 我走小路...

tiny_cnn代码详解（1）——开篇 阅读数 2万+
脸性别识别教程》系列博客的编写之后，我开始将工... 博文 来自: 陈俊岭的...

o.xsd报错 阅读数 3万+
Multiple annotations found at this line: - cvc-complex-ty... 博文 来自: 黑无常

端 阅读数 4508
如果该目录下没有该文件，你可以手动创建一个。thea... 博文 来自: 教育数据...

识别-中间极光线,并解决图像拉伸问题 阅读数 4684
景念大家，最近公司一段时间忙吧，无暇东顾。扫噶... 博文 来自: chonbj的...

算法解析 阅读数 63万+
(每条边只走一次，两两可达) 孤立的一个点也是一... 博文 来自: 九野的博客

同时调用多个倒计时(最新的) 阅读数 51万+
多个倒计时(最新的) 最近需要网页添加多个倒计时. 查... 博文 来自: Websites

法注入的问题（与文件包位置有关） 阅读数 23万+
en构建，大致结构如下：核心Spring框架一个module... 博文 来自: 开发随笔

动态库（一） 阅读数 1万+
明一下如何在VS环境下生成和使用C++的静态库与动... 博文 来自: luyan的博客

多开联机补丁 信息发布与收集点 阅读数 5万+
博文 来自: Smile_qiq...

崩溃的解决办法 阅读数 4011
l, 一般都是用来做选项卡的切换，这里记录一下使用fr... 博文 来自: 想你依然...

阅读数 2310
llImage *src=cvLoadImage("D:/a_project/image/1.jpg",... 博文 来自: weiwei20...

Objective-C培训 交互设计视频教程 颜色模型

live底部 react extjs glyph 图标 java拦截器学习 java 拦截器学习



_William_Cheung

关注

原创	粉丝	喜欢	评论
132	76	54	64

等级: 博客 6

积分: 5726

勋章:

访问: 56万+

排名: 7856



最新文章

[IntelliJ IDEA中多项目进行 grouping-modules](#)

[TeamViewer 免费许可到期的解决办法](#)

[Spring Boot + Spring Security OAuth2示例](#)

[在Java项目中如何反编译class文件及批量反编译?](#)

[Redis在Windows系统下安装及启动事项](#)

博主专栏

[IntelliJ IDEA](#)
文章数: 12 篇 访问量: 9万+

[Java](#)
文章数: 10 篇 访问量: 2万+

[Spring洽谈](#)
文章数: 9 篇 访问量: 9597

个人分类

[Nginx](#)

2篇

[Linux](#)

11篇

[Java](#)

68篇

[Java面试题](#)

1篇

[MySQL](#)

11篇

展开

归档

[2019年4月](#)

1篇

[2019年3月](#)

1篇

[2018年10月](#)

2篇

[2018年9月](#)

1篇

[2018年5月](#)

1篇

展开

热门文章

[IntelliJ Idea 导入多个maven项目展示在右侧栏Maven Projects](#)

阅读数 30246

[Tomcat 8 启动startup.bat闪退的解决方法](#)

~

0

阅读数 26999

使用logback输出日志，自定义输出格式

阅读数 23461

java.lang.ArithmeticException: Rounding necessary

阅读数 22825

使用IntelliJ IDEA中的Spring Initializr来快速构建Spring Boot/Cloud工程

阅读数 22450

最新评论

Tomcat 8 启动startu...

weixin_41288824: [reply]qq496013218[/reply] 正解

云服务器安装MySQL详细指南

weixin_44149454: 乱七八糟的

正确认识Docker、Kubern...

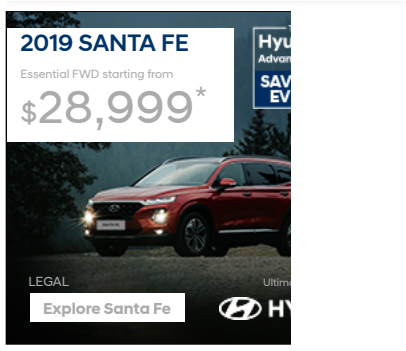
lawwimpy: 这篇文章抄来抄去,有这功夫不能写点自己的东西??????

ubuntu 安装 gedit

g_tlover: 一次安装就可以吗

IntelliJ IDEA 热部署...

qq_24612753: [reply]qq496013218[/reply]配置tomcat只能是部分能起作用，绝大部分不行





程序人生



CSDN资讯

 QQ客服

 kefu@csdn.net

 客服论坛

 400-660-0108

工作时间 8:30-22:00

关于我们 招聘 广告服务 网站地图

 百度提供站内搜索 京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司


公司


网络110报警服务 经营性网站备案信息


北京互联网违法和不良信息举报中心


中国互联网举报中心 家长监护 版权申诉

~









<

>