链滴

Spring boot 2.0 解决 RequestBody 和 ResponseBody 重复读取问题

作者：adongs

原文链接：https://ld246.com/article/1584946548831

来源网站：链滴

为什么RequestBody只能读取一次?

方法一

1.创建一个自定义HttpServletRequestWrapper

```
import org.springframework.util.StreamUtils;

import javax.servlet.ReadListener;
import javax.servlet.ServletInputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;
import java.io.*;

/**
 * @author yudong
 * @version 1.0
 * @date 2020/3/20 3:04 下午
 * @modified By
 */
public class BodyReaderRequestWrapper extends HttpServletRequestWrapper {
    private final byte [] body;

    /**
     *
     * @param request
     */
    public BodyReaderRequestWrapper(HttpServletRequest request) throws IOException {
        super(request);
        body = StreamUtils.copyToByteArray(request.getInputStream());
```

```
    }

    @Override
    public BufferedReader getReader() {
        return new BufferedReader(new InputStreamReader(this.getInputStream()));
    }

    @Override
    public ServletInputStream getInputStream(){
        final ByteArrayInputStream byteArrayIns = new ByteArrayInputStream(body);
        ServletInputStream servletIns = new ServletInputStream() {
            @Override
            public boolean isFinished() {
                return false;
            }

            @Override
            public boolean isReady() {
                return false;
            }

            @Override
            public void setReadListener(ReadListener readListener) {}

            @Override
            public int read() {
                return byteArrayIns.read();
            }
        };
        return  servletIns;
    }
}
```

2.创建一个filter

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.core.annotation.Order;
import org.springframework.stereotype.Component;

import javax.servlet.*;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpServletRequest;
import java.io.IOException;

/**
 * @author yudong
 * @version 1.0
 * @date 2020/3/20 3:07 下午
 * @modified By
 */
@Component
@Order
@WebFilter(filterName="bodyReader", urlPatterns="/**")
```

```
public class BodyReaderFilter implements Filter  {

    private final static Logger logger = LoggerFactory.getLogger(BodyReaderFilter.class);

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
        logger.info("filter init");
    }

    @Override
    public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, Filter
hain filterChain)
            throws IOException, ServletException {
        HttpServletRequest req = (HttpServletRequest)servletRequest;
        BodyReaderRequestWrapper requestWrapper  = new BodyReaderRequestWrapper(req);
        filterChain.doFilter(requestWrapper, servletResponse);
    }

    @Override
    public void destroy() {

    }

}
```

3.测试

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.util.StreamUtils;
import org.springframework.web.bind.annotation.*;

import javax.servlet.ServletInputStream;
import javax.servlet.http.HttpServletRequest;
import java.nio.charset.Charset;

/**
 * 重复读取测试
 * @author yudong
 * @version 1.0
 * @date 2020/2/11 2:38 下午
 * @modified By
 */
@ResponseBody
@Controller
public class TestController {

    @Autowired
    private Test test;

  @PostMapping("test")
  public String test(HttpServletRequest httpServletRequest, @RequestBody String ss)throws E
ception{
      ServletInputStream inputStream = httpServletRequest.getInputStream();
```

```
        String s = StreamUtils.copyToString(inputStream, Charset.forName("UTF-8"));
        return s;
    }


}
```

方法二

1.创建一个filter

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.core.annotation.Order;
import org.springframework.stereotype.Component;
import org.springframework.web.util.ContentCachingRequestWrapper;

import javax.servlet.*;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpServletRequest;
import java.io.IOException;

/**
 * @author yudong
 * @version 1.0
 * @date 2020/3/20 3:07 下午
 * @modified By
 */
@Component
@Order
@WebFilter(filterName="bodyReader", urlPatterns="/**")
public class BodyReaderFilter implements Filter  {

    private final static Logger logger = LoggerFactory.getLogger(BodyReaderFilter.class);

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
        logger.info("filter init");
    }

    @Override
    public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, Filter
hain filterChain)
            throws IOException, ServletException {
        HttpServletRequest req = (HttpServletRequest)servletRequest;
        ContentCachingRequestWrapper contentCachingRequestWrapper = new ContentCachin
RequestWrapper(req);
        filterChain.doFilter(contentCachingRequestWrapper, servletResponse);
    }

    @Override
    public void destroy() {
```

```
    }

}
```

2.编写测试controller

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.util.ContentCachingRequestWrapper;

import javax.servlet.http.HttpServletRequest;

/**
 * @author yudong
 * @version 1.0
 * @date 2020/2/11 2:38 下午
 * @modified By
 */
@ResponseBody
@Controller
public class TestController {


  @PostMapping("test")
  public String test(HttpServletRequest httpServletRequest, @RequestBody String ss)throws E
ception{
      ContentCachingRequestWrapper contentCachingRequestWrapper = (ContentCachingReq
estWrapper)httpServletRequest;
      byte[] contentAsByteArray = contentCachingRequestWrapper.getContentAsByteArray();
      String body = new String(contentAsByteArray);
      return body ;
   }


}
```

ResponseBody 重复读取

```
    ContentCachingResponseWrapper responseToCache = new ContentCachingResponseWra
per(httpServletResponse);
      //读取值
      String responseBody = new String(responseToCache.getContentAsByteArray());
      //重写写入到responseBody中
      responseToCache.copyBodyToResponse();
```

原文链接：Spring boot 2.0 解决 RequestBody 和 ResponseBody 重复读取问题