## 一个springboot项目,提供给了前端播放录音接口,这个接口是作为代理请求第三方的播放录音接口,通过openfeign请求的，不小心关闭了流，通过如下方式解决了，请你帮我在详细分析一下.

1.Slf4jLogger 开启debug级别的日志

@Override  
protected Response logAndRebufferResponse(String configKey,  
 Level logLevel,  
 Response response,  
 long elapsedTime)  
 throws IOException {  
 if (logger.isDebugEnabled()) {//关键代码  
 return super.logAndRebufferResponse(configKey, logLevel, response, elapsedTime);  
 }  
 return response;  
}

2.feign.Logger#logAndRebufferResponse 方法 会读取流变成ByteArrayBody,从而可重复读取

// HTTP 204 No Content "...response MUST NOT include a message-body"  
// HTTP 205 Reset Content "...response MUST NOT include an entity"  
if (logLevel.ordinal() >= Level.FULL.ordinal()) {  
 log(configKey, ""); // CRLF  
}  
byte[] bodyData = Util.toByteArray(response.body().asInputStream());  
bodyLength = bodyData.length;  
if (logLevel.ordinal() >= Level.FULL.ordinal() && bodyLength > 0) {  
 log(configKey, "%s", decodeOrDefault(bodyData, UTF\_8, "Binary data"));  
}  
log(configKey, "<--- END HTTP (%s-byte body)", bodyLength);  
return response.toBuilder().body(bodyData).build();

1. ByteArrayBody 是可以反复读取的，并且没有关闭

private static final class ByteArrayBody implements Response.Body {  
  
 private final byte[] data;  
  
 public ByteArrayBody(byte[] data) {  
 this.data = data;  
 }  
  
 private static Body orNull(byte[] data) {  
 if (data == null) {  
 return null;  
 }  
 return new ByteArrayBody(data);  
 }  
  
 private static Body orNull(String text, Charset charset) {  
 if (text == null) {  
 return null;  
 }  
 checkNotNull(charset, "charset");  
 return new ByteArrayBody(text.getBytes(charset));  
 }  
  
 @Override  
 public Integer length() {  
 return data.length;  
 }  
  
 @Override  
 public boolean isRepeatable() {  
 return true;//关键代码  
 }  
  
 @Override  
 public InputStream asInputStream() throws IOException {  
 return new ByteArrayInputStream(data);//关键代码  
 }  
  
 @SuppressWarnings("deprecation")  
 @Override  
 public Reader asReader() throws IOException {  
 return new InputStreamReader(asInputStream(), UTF\_8);  
 }  
  
 @Override  
 public Reader asReader(Charset charset) throws IOException {  
 checkNotNull(charset, "charset should not be null");  
 return new InputStreamReader(asInputStream(), charset);  
 }  
  
 @Override  
 public void close() throws IOException {} //关键代码  
  
}

4.DefaultFeignLoggerFactory FeignClientFactoryBean

package org.springframework.cloud.openfeign;  
  
import feign.Logger;  
import feign.slf4j.Slf4jLogger;  
  
  
public class DefaultFeignLoggerFactory implements FeignLoggerFactory {  
  
 private final Logger logger;  
  
 public DefaultFeignLoggerFactory(Logger logger) {  
 this.logger = logger;  
 }  
  
 @Override  
 public Logger create(Class<?> type) {  
 return this.logger != null ? this.logger : new Slf4jLogger(type);  
 }  
  
}