

接口中过滤emoji表情工具

过滤emoji表情，可以用来判断参数中是否包含emoji表情，可以用来过滤掉参数中的emoji表情

```
package com.aspire.andfans.domain.util;

import org.apache.commons.lang.StringUtils;

public class EmojiFilter {

    public static boolean containsEmoji(String source) {
        int len = source.length();
        boolean isEmoji = false;
        for (int i = 0; i < len; i++) {
            char hs = source.charAt(i);
            if (0xd800 <= hs && hs <= 0xdbff) {
                if (source.length() > 1) {
                    char ls = source.charAt(i + 1);
                    int uc = ((hs - 0xd800) * 0x400) + (ls - 0xdc00) + 0x10000;
                    if (0x1d000 <= uc && uc <= 0x1f7ff) {
                        return true;
                    }
                }
            } else {
                // non surrogate
                if (0x2100 <= hs && hs <= 0x27ff && hs != 0x263b) {
                    return true;
                } else if (0x2B05 <= hs && hs <= 0x2b07) {
                    return true;
                } else if (0x2934 <= hs && hs <= 0x2935) {
                    return true;
                } else if (0x3297 <= hs && hs <= 0x3299) {
                    return true;
                } else if (hs == 0xa9 || hs == 0xae || hs == 0x303d
                    || hs == 0x3030 || hs == 0x2b55 || hs == 0x2b1c
                    || hs == 0x2b1b || hs == 0x2b50 || hs == 0x231a) {
                    return true;
                }
            }
            if (!isEmoji && source.length() > 1 && i < source.length() - 1) {
                char ls = source.charAt(i + 1);
                if (ls == 0x20e3) {
                    return true;
                }
            }
        }
    }

    return isEmoji;
}
```

```
}
```

```
private static boolean isEmojiCharacter(char codePoint) {  
    return (codePoint == 0x0) || (codePoint == 0x9) || (codePoint == 0xA)  
        || (codePoint == 0xD)  
        || ((codePoint >= 0x20) && (codePoint <= 0xD7FF))  
        || ((codePoint >= 0xE000) && (codePoint <= 0xFFFF))  
        || ((codePoint >= 0x10000) && (codePoint <= 0x10FFFF));  
}
```

```
/**
```

```
 * 过滤emoji 或者 其他非文字类型的字符
```

```
 *
```

```
 * @param source
```

```
 * @return
```

```
 */
```

```
public static String filterEmoji(String source) {  
    if (StringUtils.isBlank(source)) {  
        return source;  
    }  
    StringBuilder buf = null;  
    int len = source.length();  
    for (int i = 0; i < len; i++) {  
        char codePoint = source.charAt(i);  
        if (isEmojiCharacter(codePoint)) {  
            if (buf == null) {  
                buf = new StringBuilder(source.length());  
            }  
            buf.append(codePoint);  
        }  
    }  
    if (buf == null) {  
        return source;  
    } else {  
        if (buf.length() == len) {  
            buf = null;  
            return source;  
        } else {  
            return buf.toString();  
        }  
    }  
}
```

```
}
```

