关于 Java 中 Match 类的 appendReplacement（）方法的一个坑

作者：Levent
原文链接：https://ld246.com/article/1512312977043
来源网站：链滴

关于Java中Match类的appendReplacement（）方法一个坑

问题描述

在向替换结果中追加信息的时候，调用match.appendReplacement(buffer,str)时出现了上述异常。

```
class XX{
    public String renderString(String source, Map<String, Object> context, Map<String, Obj
ct> data) throws OgnlException {
        Pattern pattern = Pattern.compile(DELIM);
        Matcher matcher = pattern.matcher(source);
        StringBuffer buffer = new StringBuffer();
        while (matcher.find()) {
            String e = matcher.group(1);
            if (e == null) throw new NullPointerException("expression can not be null");
            Object value = Ognl.getValue(e, context, data);
            String str = null == value ? "null" : value.toString();
            matcher.appendReplacement(buffer, str); //在此处报错
        }
        matcher.appendTail(buffer);
        return buffer.toString();
    }
}
```

问题原因

通过阅读Matcher类的源码并查阅文档得知，在 matcher.appendReplacement(buffer, str)方法中st中若含有'\'、'$'字符时，将存在特殊含义，
'$n'代表匹配的第n组结果，'\'将对其之后的字符进行转义。

文档如下

实现非终端添加和替换步骤。
此方法执行以下操作：

它从添加位置开始在输入序列读取字符，并将其添加到给定字符串缓冲区。在读取以前匹配之前的后字符（即位于索引 start() - 1 处的字符）之后，它就会停止。

它将给定替换字符串添加到字符串缓冲区。

它将此匹配器的添加位置设置为最后匹配位置的索引加 1，即 end()。

替换字符串可能包含到以前匹配期间所捕获的子序列的引用：$g 每次出现时，都将被 group(g) 计算结果替换。$ 之后的第一个数始终被视为组引用的一部分。如果后续的数可以形成合法组引用，将被合并到 g 中。只有数字 '0' 到 '9' 被视为组引用的可能组件。例如，如果第二个组匹配字符串 "fo "，则传递替换字符串 "$2bar" 将导致 "foobar" 被添加到字符串缓冲区。可能将美元符号 ($) 作为替字符串中的字面值（通过前面使用一个反斜线 (\$)）包括进来。

注意，在替换字符串中使用反斜线 (\) 和美元符号 ($) 可能导致与作为字面值替换字符串时所产生结果不同。美元符号可视为到如上所述已捕获子序列的引用，反斜线可用于转义替换字符串中的字面字符。

此方法设计用于循环以及 appendTail 和 find 方法中。例如，以下代码将 one dog two dogs in t e yard 写入标准输出流中：

```
Pattern p = Pattern.compile("cat");
Matcher m = p.matcher("one cat two cats in the yard");
StringBuffer sb = new StringBuffer();
while (m.find()) {
    m.appendReplacement(sb, "dog");
}
m.appendTail(sb);
System.out.println(sb.toString());
```

参数：
sb - 目标字符串缓冲区。
replacement - 替换字符串。
返回：
匹配器。
抛出：
IllegalStateException - 如果没有尝试任何匹配，或者以前的匹配操作失败。
IndexOutOfBoundsException - 如果替换字符串引用模式中不存在的捕获组。

源码如下

```
public final class Matcher implements MatchResult {
   //省略
   public Matcher appendReplacement(StringBuffer sb, String replacement) {
      //省略
      while (cursor < replacement.length()) {
         char nextChar = replacement.charAt(cursor);
         if (nextChar == '\\') {
            cursor++;
            if (cursor == replacement.length())
               throw new IllegalArgumentException(
                   "character to be escaped is missing");
            nextChar = replacement.charAt(cursor);
            result.append(nextChar);
            cursor++;
         } else if (nextChar == '$') {
            //省略
         } else {
            result.append(nextChar);
            cursor++;
         }
      }
      // Append the intervening text
      sb.append(text, lastAppendPosition, first);
      // Append the match substitution
      sb.append(result);
```

```
            lastAppendPosition = last;
            return this;
        }
    }
```

原文链接：关于 Java 中 Match 类的 appendReplacement（）方法的一个坑