



链滴

# Idea 插件开发手记

作者: [ZephyrJung](#)

原文链接: <https://ld246.com/article/1503549518434>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Idea有很多实用的插件，极大地方便了码农的搬砖工作

但有些功能求人不如求己，毕竟每个人的习惯和遇到的问题是不一样的。

## 问题背景

1. 有的时候（然而我已经忘了什么时候），会想知道某段代码多少行，之前的做法是复制出来放到新的文本编辑器中查看，后来感觉这种事情应当可以写插件来实现，于是诞生了这个项目——计算代码数。

这是个极为简单的功能，但是写插件这种事情我是一头雾水，于是第一步，搜索github，浏览了许多发现了一个插件，作用是给一段文本排序：[LinesSorter](#)，想想恰好能改为我想要的功能，果断FORK来。

2. 在之前的工作中，经常会犯一种低级错误——set错字段，因为两个字段的单词首字母很接近（可头两个字母都是一样的），恰好类型有一样，一个不注意，setOrderId可能就set成了OrnotId（意会），于是产生了第二个想法——标记行内相同的单词。如此只要标记下，就能很直观的看到是否set正了。

轻车熟路，搜索github，发现插件[MultiHighlight](#)，果断FORK。

## 环境搭建

环境搭建很容易，安装intelliJ SDK插件即可，选择SDK时，新建并定位到intelliJ文件夹本身即可。

不同的插件可能基于不同的SDK构建，所以down下来的代码未必能够正常运行，我一般是将代码拷到自己的项目中

除了代码，最重要的是plugin.xml，位于resources中。

项目构建成功后运行，就可以查看效果，在通过调试了解代码的运行逻辑。

## 正文

核心代码非常少，如下：

```
@Override
public void doExecute(Editor editor, @Nullable Caret caret, DataContext dataContext) {
    final Document doc = editor.getDocument();
    int startLine;
    int endLine;
    boolean hasSelection = editor.getSelectionModel().hasSelection();
    if (hasSelection) {
        startLine = doc.getLineNumber(editor.getSelectionModel().getSelectionStart());
        endLine = doc.getLineNumber(editor.getSelectionModel().getSelectionEnd());
        if (doc.getLineStartOffset(endLine) == editor.getSelectionModel().getSelectionEnd()) {
            endLine--;
        }
    } else {
        startLine = 0;
        endLine = doc.getLineCount() - 1;
    }
    if (startLine <= endLine) {
        String message = String.format("%d lines are selected!", endLine - startLine + 1);
```

```

Messages.showInfoMessage(message, "Count Result");
}
}

```

- editor.getDocument() 获取到了当前编辑器的内容
- hasSelection很好理解，就是有没有选中文本
- startLine和endLine获取到了选中文本的行号
- 最后通过showInfoMessage方法，将计算好的行号通过消息框的方式展现

其实我真正写的代码就在最后那个if中，其他的都是Line Sorter的代码 ☹  
oy

## 添加配置

```

<actions>
<group id="editor.lines.group">
<separator/>
<action id="lines.count"
class="org.b3log.zephyr.action.CountLinesAction"
text="Count Lines"
description="Count selected lines or whole file if selection is empty">
</action>
<add-to-group group-id="EditorPopupMenu" anchor="last"/>
</group>
</actions>

```

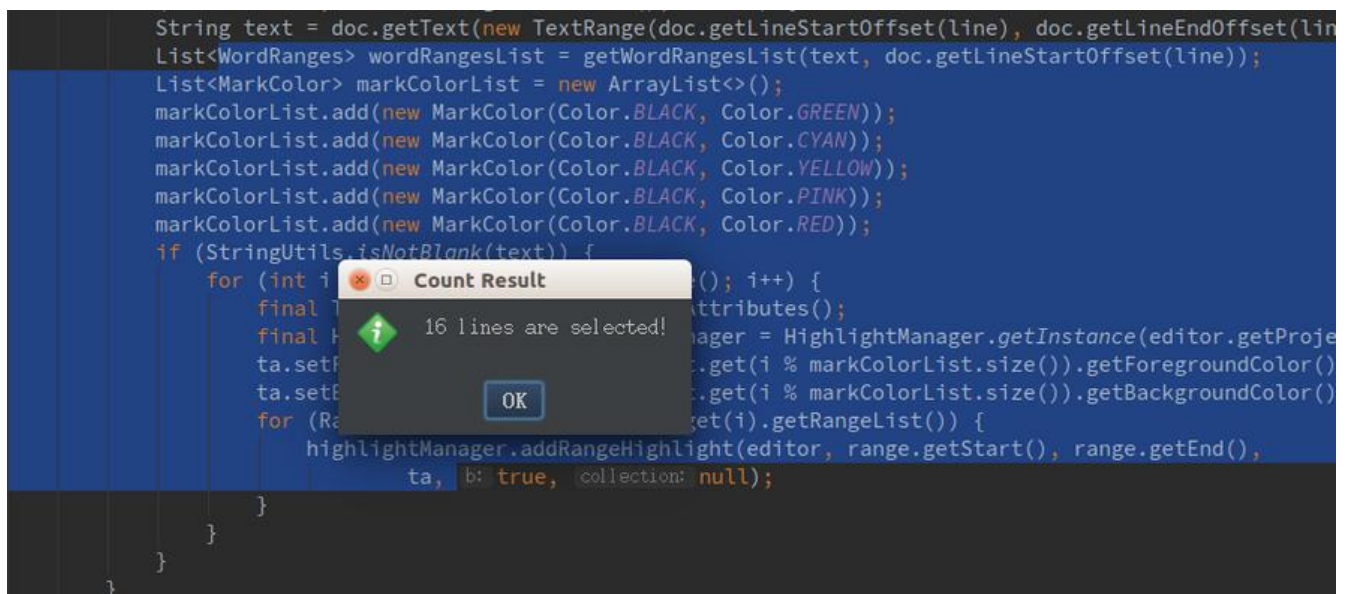
这段配置的重点在于add-to-group，他将这个功能添加到了右键菜单中，这个就是通过文档查到的了 ☹  
oy

## 部署

Build中，会多一个Prepare Plugin Module XXX For Deployment

构建好，通过本地安装就可以供自己使用了

## 效果



```

@Override
public void doExecute(Editor editor, @Nullable Caret caret, DataContext dataContext) {
    try {
        final Document doc = editor.getDocument();
        for (int line = 0; line < doc.getLineCount(); line++) {
            String text = doc.getText(new TextRange(doc.getLineStartOffset(line), doc.getLineEndOffset(line)));
            List<WordRanges> wordRangesList = getWordRangesList(text, doc.getLineStartOffset(line));
            List<MarkColor> markColorList = new ArrayList<>();
            markColorList.add(new MarkColor(Color.BLACK, Color.GREEN));
            markColorList.add(new MarkColor(Color.BLACK, Color.CYAN));
            markColorList.add(new MarkColor(Color.BLACK, Color.YELLOW));
            markColorList.add(new MarkColor(Color.BLACK, Color.PINK));
            markColorList.add(new MarkColor(Color.BLACK, Color.RED));
            if (StringUtils.isNotBlank(text)) {
                for (int i = 0; i < wordRangesList.size(); i++) {
                    final TextAttributes ta = new TextAttributes();
                    final HighlightManager highlightManager = HighlightManager.getInstance(editor.getProject());
                    ta.setForegroundColor(markColorList.get(i % markColorList.size()).getForegroundColor());
                    ta.setBackgroundColor(markColorList.get(i % markColorList.size()).getBackgroundColor());
                    for (Range range : wordRangesList.get(i).getRangeList()) {
                        highlightManager.addRangeHighlight(editor, range.getStart(), range.getEnd(),
                            ta, true, null);
                    }
                }
            }
        }
    }
}

```

供自己使用，所以不讲究那么多了ojoy  
有这个颜色标记，下次应该不会再set错字段了.....

## 后记

官方似乎没有给出明确的API文档（或者我没找到.....），一方面通过一个[Guide](#)，一方面查找相关类的avaDoc（Github上有IntelliJ Community的源码，线上看就好了，要down下来的话，我down了八百M没有完成，放弃了。。。）

稍微复杂一点的功能就是第二个，行内代码着色，感兴趣的可以直接上我的[Github](#)上看

本文纯属灌水