



链滴

Idea 文件注释和 java 方法注释

作者: [zk123](#)

原文链接: <https://ld246.com/article/1480507818870>

来源网站: [链滴](#)

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

规范

一个很好规范，应该包括注释这部分。

初始

在Idea中，刚开始创建一个文件的时候，都会由一个这样的注释说明

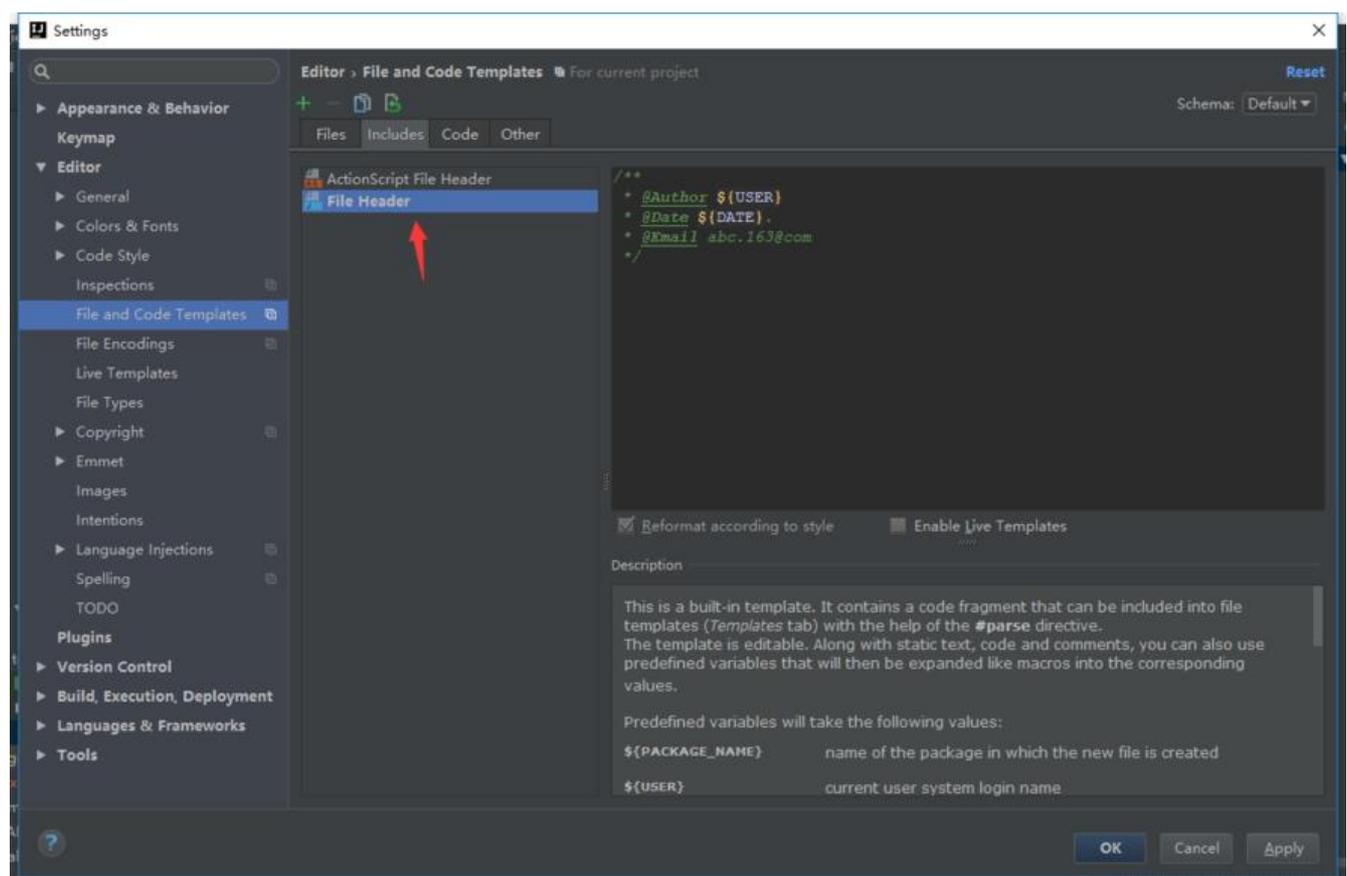
```
/**
 * Created by 作者 on 2016/9/1.
 */
```

问题

我们有时候可能会向，当文件创建的时候，除了应该由的注释，能不能**加点**其它的东西呢，或者改成们想要的注释呢？

解决

1. 打开setting(快捷键ctrl+alt+s),找到 > **File and Code Templates**



2. 一般的文件在File Header中设置,想设置的具体变量在图片右下方Description中有具体描写。

其它

如果想在Java中方法上编写类似的注释。比如Sym里的这部分，应该在**Setting**(快捷键ctrl+alt+s),找 > **Live Templates**中设置

```
/**
 * Bets 1A0001.
 *
 * @param context the specified context
 * @param request the specified request
 * @param response the specified response
 * @throws Exception exception
 */
@RequestMapping(value = "/activity/1A0001/bet", method = RequestMethod.POST)
@Before(adviceClass = {StopwatchStartAdvice.class, LoginCheck.class, CSRFCheck.class, Activity1A0001Validation.class})
@After(adviceClass = StopwatchEndAdvice.class)
public void bet1A0001(final HTTPRequestContext context,
                    final HttpServletRequest request, final HttpServletResponse response) throws Exception {
    context.renderJSON().renderFalseResult();

    final JSONObject requestJSONObject = (JSONObject) request.getAttribute(Keys.REQUEST);

    final int amount = requestJSONObject.optInt(Common.AMOUNT);
    final int smallOrLarge = requestJSONObject.optInt(Common.SMALL_OR_LARGE);

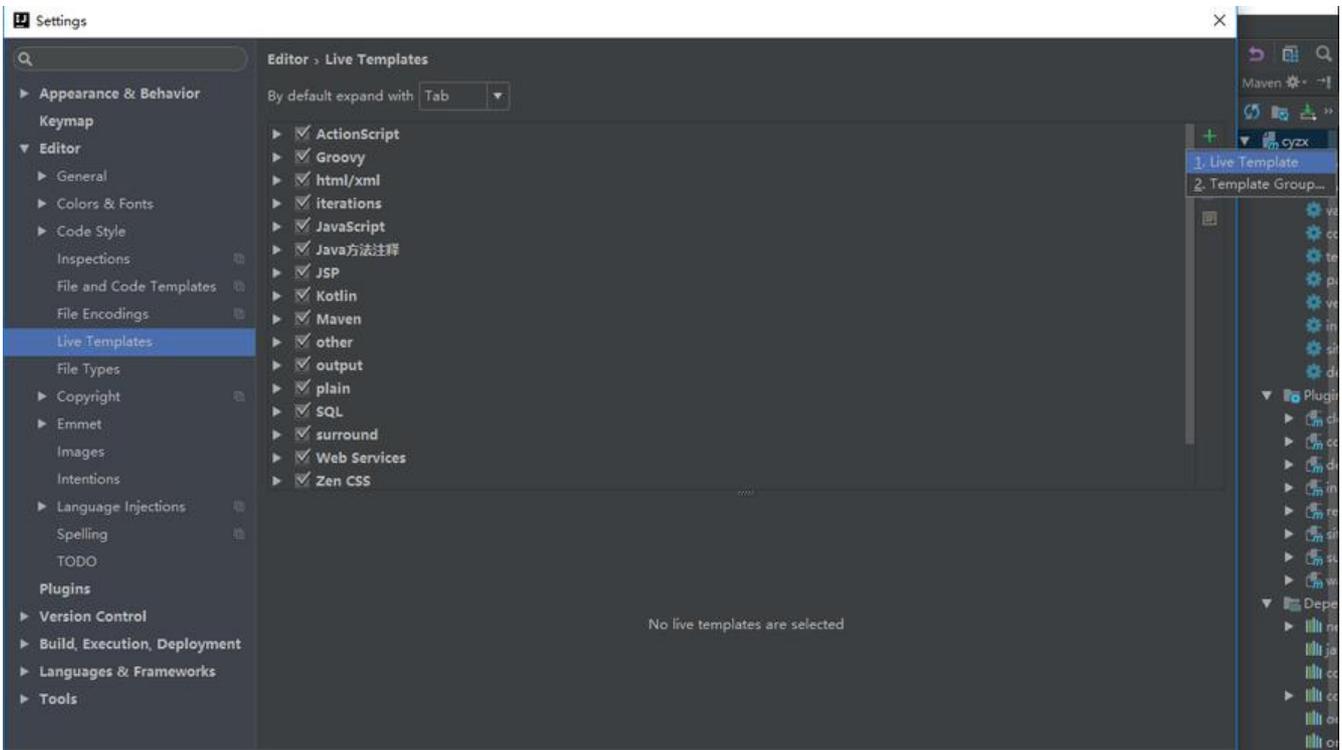
    final JSONObject currentUser = (JSONObject) request.getAttribute(User.USER);
    final String fromId = currentUser.optString(Keys.OBJECT_ID);

    final JSONObject ret = activityMgmtService.bet1A0001(fromId, amount, smallOrLarge);

    if (ret.optBoolean(Keys.STATUS_CODE)) {
        String msg = langPropsService.get("activity1A0001BetedLabel");
        final String small = langPropsService.get("activity1A0001BetSmallLabel");
        final String large = langPropsService.get("activity1A0001BetLargeLabel");
        msg = msg.replace("{smallOrLarge}", smallOrLarge == 0 ? small : large);
        msg = msg.replace("{point}", String.valueOf(amount));
    }
}
```

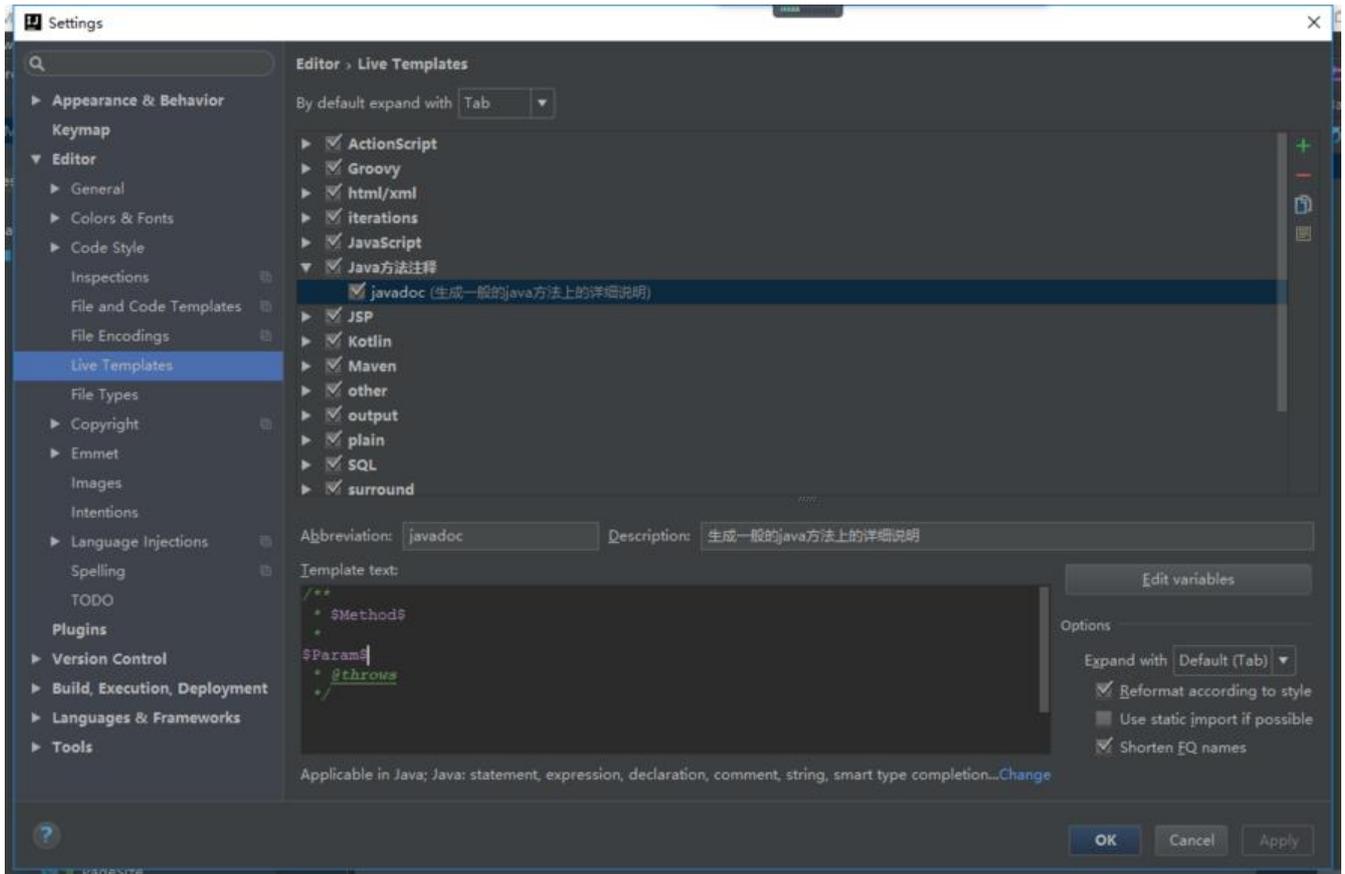
live templates 中可以设置类似java方法上的注释，也可以设置快捷键；（比如：输入sout,按下tab能出现System.out.println()）

Live Templates



如图，在**Live Templates**绿色加号下有两个选项，其中一个**Templates Group**就是模板组，里面以包含许多**Live Templates**模板。另一个**Live Templates**才是我们要设置的模板。我们可以新建一

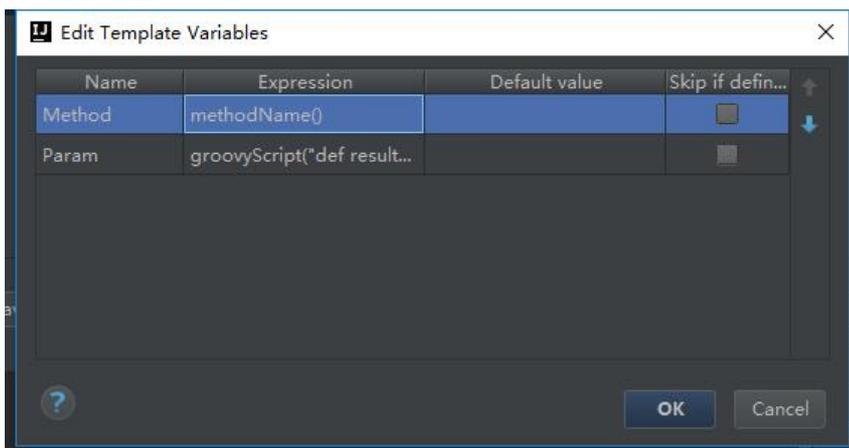
属于自己的模板组，再创建一个新的Live Templates模板。



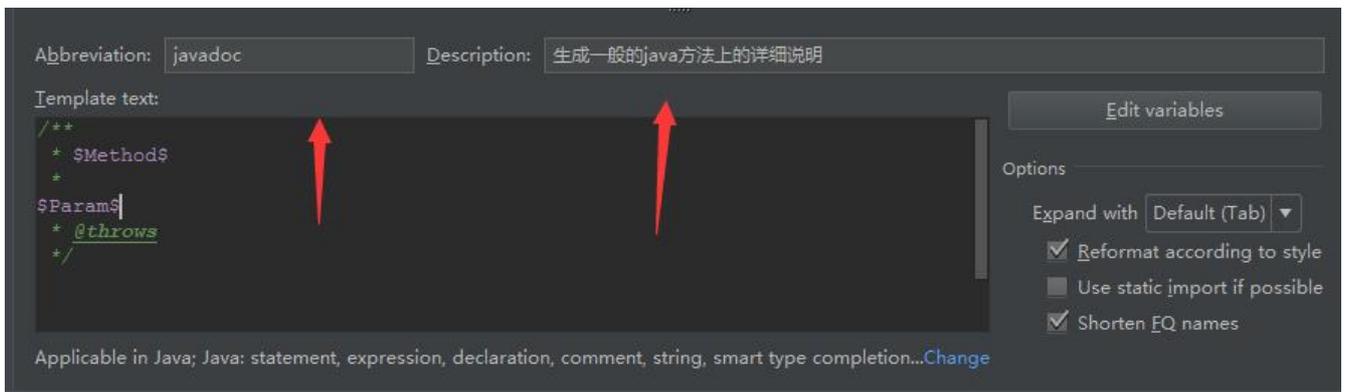
Templates text 中

```
/**  
 * $Method$  
 * $Param$  
 */
```

- 两个\$之间的是预设变量名，点击 **Edit variables**,来设置它们具体代表的含义



- **Abbreviation**是要设置的快捷键，**Description**来描述它们。



- 也可以试着加其它写东西，比如：日期，用户等等。。。

测验

找到一个类，输入预设快捷键，OK。更多详见[这里](#)

好像没有throws的有关设置，我只能自己打上去。

```
/**
 * Apple
 *
 * @param AppleName description()
 * @param AppleTaste description()
 * @throws
 */
public void Apple(String AppleName, String AppleTaste) throws RuntimeException
{
}
```