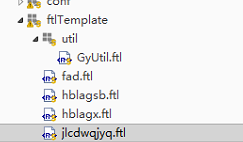
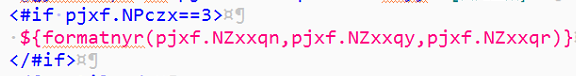
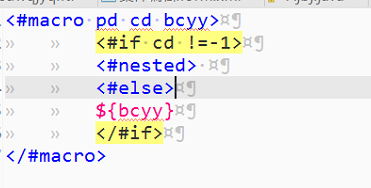
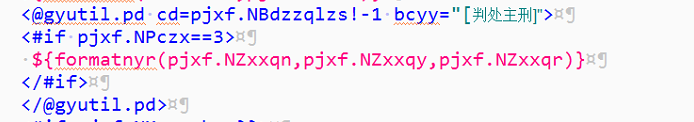
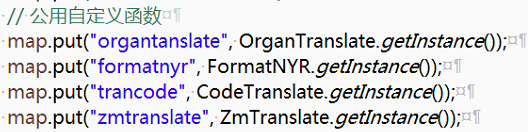
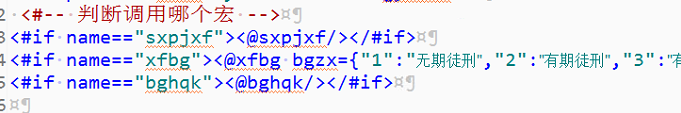
1. 在ftlTemplate的目录下new一个新的file，将后缀改为ftl文件。
2. 在第一行可以引入其他ftl模板文件，在这里我们引入一个工具类模板
3. 新建一个宏区域，目的是为了一个模板里可以写多个模板语句，达到一个模板处处使用的目的，语法为<#macro 宏名称 参数></#macro> 注意一定要有结束符，可以安装Freemarker插件来防止语法出错。
4. 宏区域内就可以用Freemarker的语法来书写模板语法，具体Freemaker的常用标签如下<https://blog.csdn.net/pengpengpeng85/article/details/52070602>
5. 需要特殊注意的是，Freemarker对空相当的敏感，所以所有的地方都得加上空处理，在最简单的标签内饰${Value!defaultValue}这样来处理，比如我们要取出案号，则是${aj.Cah!”[案号]”}这样来书写。在if判断中，比如这样一个简单的if，意思是判处主刑为有期徒刑我们就把有期徒刑年月日打印出来，若这个NPczx为空的话这一行是会报错的，所以这里我们统一处理了一下，我在GyUtil.ftl里面写了一个宏来处理，，宏名称叫pd，有两个参数cd和bcyy，cd是传入的可能为空的值，bcyy是在cd为空的时候的补充语句，我们在刚刚个if前面加上这个宏就可以了，。
6. 在文书模板中我们会常常用到一些公用的方法，比如翻译组织机构，比如翻译单值代码等等，这个时候我们就可以用自定义函数了。现在已经有了单值代码翻译器、组织机构翻译器、罪名翻译器、年月日翻译器。自定义函数的使用方法就是在${}内些方法名然后把参数传入进去，比如，在这里特殊的组织机构翻译器，可以传一个参数也可以传两个参数，传一个参数就会直接把改组织机构代码值翻译成中文，若是传入两个，若两个代码一样则返回“本院”，若不一样则返回第一个代码的代码值。当然了，大家也可以根据自己的实际需求自己写自定义函数。现在讲一讲如何书写自定义函数，首先在这个路径下新建一个java类，实现TemplateMethodModelEx接口，在exec方法内就是自己定义的自定义函数了，arguments就是你传入的参数，如果是只有一个参数.get(0)就可以取出来了，写完后一定要注册才能够使用，注册有两种办法一种是模板内注册一种是后台java内注册，因为公用的自定义函数最好使用单例模式来书写所以更推荐使用java注册，注册方法就是在Map里存一个键值对就可以了，
7. 书写完了宏区域以后要记得在模板开头写一个if语句，根据传入的参数不同来判断调用哪个宏。（这一步很容易忘记，也不太人性化，后期也许能优化？）
8. 接下来就是去调用刚刚写的模板了，首先在解析器目录下新建文书解析器，然后把要使用到的实体都查询到，在使用FreeMarkerWsUtils.generateSentencesByTemplate方法来使用模板返回语句，generateSentencesByTemplate方法有四个参数，分别意思为模板名称，实体Map，是否保持模板里的格式若为否则无论模板里是什么格式多少个空格换行最后返回的都是一句话，模板文件内的宏名称。大家可以在文书解析器里通过模板名称和宏名称来精确的调用宏。