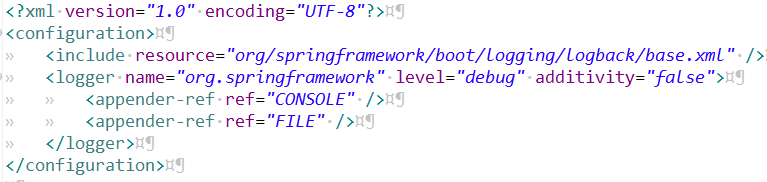
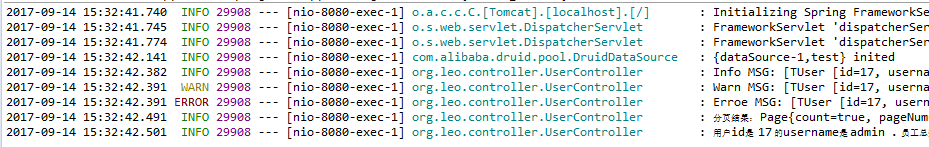
# Spring Boot自定义Logback

在《Java面试题 Part9 Spring事务的处理过程》中，为了看一下Spring AOP对事务的处理，我添加了log的配置：

在resources目录下增加logback-spring.xml，内容如下：



执行一次请求，列出如下日志：

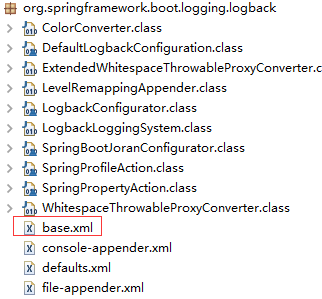


但是在实际工作中，这些配置满足不了我们的个性化需求，比如：精简一下log内容（把线程ID去掉），输出SQL，指定log文件等。

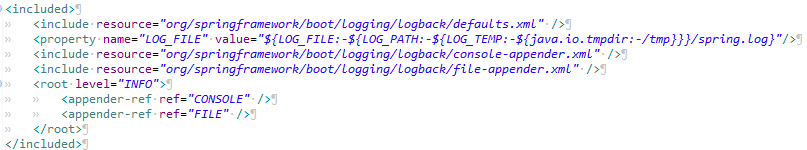
后面将对logback-spring.xml进行改造。

## base.xml

之前的logback-spring.xml文件，首先就引入了base.xml，这个文件就在spring-boot这个jar内的



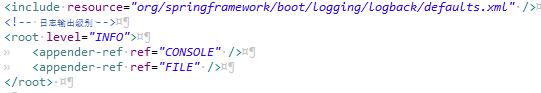
打开可以看到



我们的改造就基于对这些配置与文件的学习。

## 默认配置

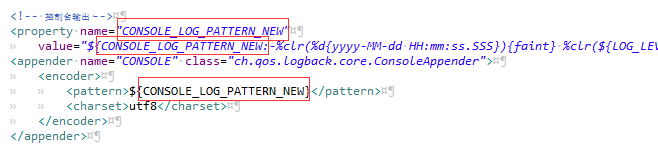
在logback-spring.xml中，我们去掉对base.xml的引用，改为defaults.xml



注意，实际工作中root这个节点一定要放在文件的最后面，后面的配置内容要在include和root两个节点之间。

## 控制台

在defaults.xml中，有个“CONSOLE\_LOG\_PATTERN”的属性，这个就是控制台输出的格式，因为我不想要显示PID，所以仿照defaults.xml与coosole-appender.xml新建了如下属性：



注意三个红框都要修改。这么做，控制台输出的log格式就可以自定义，并且还是彩色的。

## 日志文件

默认的日志文件输出配置在file-appender.xml，仿照其做如下配置：

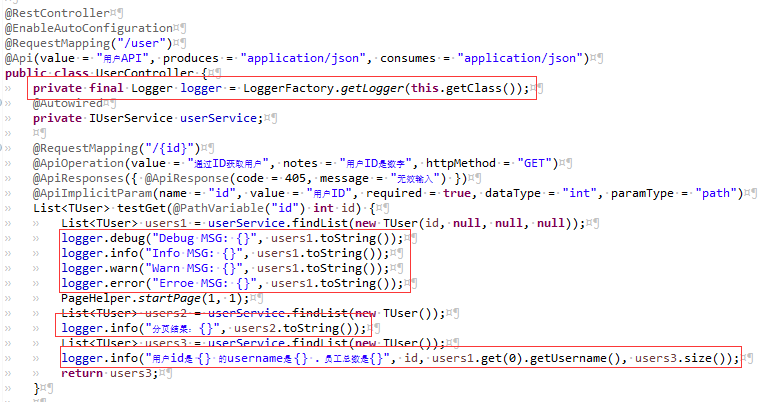


## 配置要显示日志的类

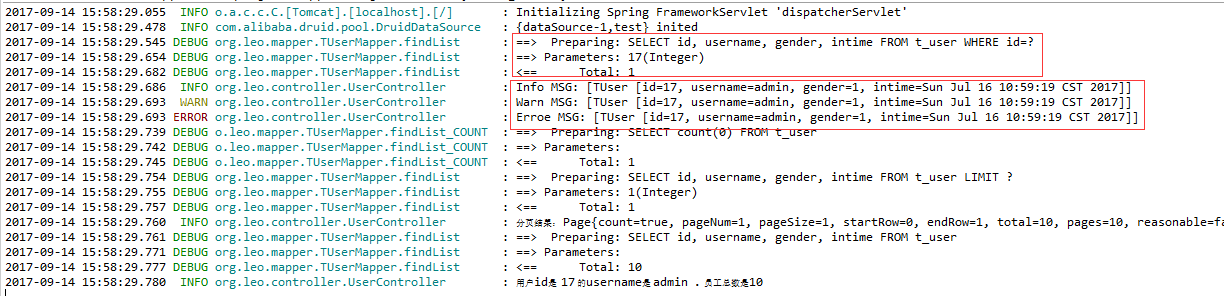


这里我少写了Service层的日志，实际中应该加上。

代码：



## 结果



对比第一个日志，PID就没了，但是彩色还保留着。

第一个红框就是SQL语句。

第二个红框里，在代码中我们写了Debug MSG，但是在配置里我们配置的是info，所以只输出了info、warn、error信息。