**Spring boot 整合log4j2**

**一．Maven 依赖**

1.Web starter中排除依赖spring-boot-starter-logging，

<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-web</**artifactId**>  
 <**exclusions**>  
 <**exclusion**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-logging</**artifactId**>  
 </**exclusion**>  
 </**exclusions**>  
</**dependency**>  
<**dependency**>

2.引入log4j2的支持包和log4j包

<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-log4j2</**artifactId**>  
 <**version**>1.3.8.RELEASE</**version**>  
</**dependency**>

<**dependency**>  
 <**groupId**>log4j</**groupId**>  
 <**artifactId**>log4j</**artifactId**>  
 <**version**>${log4j.version}</**version**>  
 </**dependency**>

**二．resources目录下面添加log4j2.xml文件**

文件内容如下 ：

*<?***xml version="1.0" encoding="UTF-8"***?>*<**Configuration status="debug"**>  
 <**Appenders**>  
 <**Console name="SYSOUT" target="SYSTEM\_OUT"**>  
 <**PatternLayout pattern="%d{yyyy-MM-dd}-%t-%x-%-5p-%-10c:%m%n"**/>  
 <**ThresholdFilter level="debug" onMatch="ACCEPT" onMismatch="DENY"**/>  
 </**Console**>  
  
 *<!-- name可以自定义，作用是在Loggers中AppenderRef中使用 -->  
 <!-- fileName定义输出文件名称（当前文件） -->  
 <!-- filePattern定义输出文件名称（文件满足条件后自动截断，生成历史文件） -->* <**RollingFile name="DEBUG\_ROLLING\_FILE"  
 fileName="F:/logs/logs.logs"  
 filePattern="F:/logs/%d{yyyy-MM-dd}-debugs.log"**>  
 <**PatternLayout**>  
 <**Pattern**>%d{yyyy-MM-dd}-%t-%x-%-5p-%-10c:%m%n</**Pattern**>  
 </**PatternLayout**>  
  
 *<!-- 文件截断的条件，具体参考文档 -->*s  
 <**Policies**>  
 <**TimeBasedTriggeringPolicy interval="24"**/>  
 <**SizeBasedTriggeringPolicy size="250 MB"**/>  
 </**Policies**>  
 </**RollingFile**>  
  
 *<!-- 同一来源的Appender可以定义多个 -->* <**RollingFile name="ERROR\_ROLLING\_FILE"  
 fileName="F:/logs/error-logs.logs"  
 filePattern="F:/logs/%d{yyyy-MM-dd}-error.log"**>  
 *<!-- 可以通过该参数来设置获取日志的权限 -->* <**ThresholdFilter level="error" onMatch="ACCEPT" onMismatch="DENY"**/>  
 <**PatternLayout**>  
 <**Pattern**>%d{yyyy-MM-dd}-%t-%x-%-5p-%-10c:%m%n</**Pattern**>  
 </**PatternLayout**>  
 <**Policies**>  
 <**TimeBasedTriggeringPolicy interval="24"**/>  
 <**SizeBasedTriggeringPolicy size="250 MB"**/>  
 </**Policies**>  
 </**RollingFile**>  
 </**Appenders**>  
  
 <**Loggers**>  
 <**Root level="debug"**>  
 <**AppenderRef ref="SYSOUT"**/>  
 <**AppenderRef ref="DEBUG\_ROLLING\_FILE"**/>  
 <**AppenderRef ref="ERROR\_ROLLING\_FILE"**/>  
 </**Root**>  
 </**Loggers**>  
</**Configuration**>

**三．然后在application.properties配置文件中指明日志文件位置**

**logging.config**=**classpath:log4j2.xml**

**四.调用**

然后在类文件中通过

Logger **log** = LogManager.*getLogger*();

来获取日志记录对象。

在需要打印日志的地方调用

**logger**.info(**"我是过滤器。。。。。"**);

**注意：**默认情况下e.printStackTrace()会将异常栈打印在控制台，不会打印到日志文件中去，但是有时我们需要异常信息来排查错误，这时我们可以通过

**logger**.error(**"异常栈"**,e);

来进行打印,如：

} **catch** (Exception e) {  
 e.printStackTrace();  
 **logger**.error(**"异常栈"**,e);**//记录到日志文件中**}