**Spring配置log4j**

**一、web.xml配置**

<!-- Log4j配置 -->

<context-param>

<param-name>log4jConfigLocation</param-name>

<param-value>/WEB-INF/classes/log4j.properties</param-value>

</context-param>

<!-- 加载log4j配置文件 -->

<listener>

<listener-class>org.springframework.web.util.Log4jConfigListener</listener-class>

</listener>

**二、log4j配置文件**

#定义根日志级别和输出端(定义了两个输出端,CONSOLE,Orclight )

log4j.rootLogger=DEBUG,CONSOLE,Orclight

#定义第一个输出端，输出到控制台

log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender

log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout

log4j.appender.CONSOLE.layout.ConversionPattern=%d [%t] %-5p %c(%L) - %m%n

#定时第二个输出端，输出到文件,该文件最大空间为1000KB，超过1000KB后自动备份，生成新文件

log4j.appender.Orclight=org.apache.log4j.RollingFileAppender

#log4j.appender.Orclight.file=orclight.log

#log4j.appender.Orclight.file=${catalina.home}/logs/spring.log

log4j.appender.Orclight.file=${catalina.base}/logs/springbase.log

log4j.appender.Orclight.layout=org.apache.log4j.PatternLayout

log4j.appender.Orclight.layout.conversionPattern=%d [%t] %-5p %c(%L) - %m%n

log4j.appender.Orclight.MaxFileSize=1000KB

log4j.appender.Orclight.MaxBackupIndex=10

#定义具体某个包下的日志输出级别

log4j.logger.com.orclight.rbac=INFO,Orclight,CONSOLE

log4j.logger.com.orclight.rbac.services.impl.CacheServiceImpl=INFO,Orclight

log4j.logger.org.springframework=ERROR,Orclight

log4j.logger.org.springframework.jdbc.support=INFO

log4j.logger.org.springframework.scheduling=INFO

#定义sql语句的输出级别

log4j.logger.java.sql.Connection=DEBUG

#将tomcat启动的错误信息输出到指定文件

############################################################

# Handler specific properties.

# Describes specific configuration info for Handlers.

############################################################

handlers = org.apache.juli.FileHandler, java.util.logging.ConsoleHandler

org.apache.juli.FileHandler.level = FINE

org.apache.juli.FileHandler.directory = ${catalina.base}/logs

org.apache.juli.FileHandler.prefix = error-debug.

java.util.logging.ConsoleHandler.level = FINE

java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter