# [使用Slf4j集成Log4j2构建项目日志系统的完美解决方案](http://www.cnblogs.com/hafiz/p/6160298.html)

## 一、背景

　　最近因为公司项目性能需要，我们考虑把以前基于的log4j的日志系统重构成基于Slf4j和log4j2的日志系统，因为，使用slf4j可以很好的保证我们的日志系统具有良好的兼容性，兼容当前常见几种日志系统，而使用log4j2而不是log4j是因为Log4j 1.x 在高并发情况下出现死锁导致cpu使用率异常飙升，而Log4j2.0基于LMAX Disruptor的异步日志在多线程环境下性能会远远优于Log4j 1.x和logback（官方数据是10倍以上）。

　　关于slf4j的原理以及优点，请参见我的另一篇博客：[**slf4j介绍以及实现原理窥探**](http://www.cnblogs.com/hafiz/p/5486858.html)

## 二、搭建步骤

### 1.依赖管理

　　　　1).删除项目中存在的Log4j1.x所必须的log4j和slf4j-log4j12等依赖。

　　　　　　可以到项目的根目录，执行：mvn dependency:tree > tree.log，之后使用 cat tree.log | grep log4j命令进行查找。

1 <exclusions>

2 <exclusion>

3 <groupId>org.slf4j</groupId>

4 <artifactId>slf4j-log4j12</artifactId>

5 </exclusion>

6 <exclusion>

7 <groupId>log4j</groupId>

8 <artifactId>log4j</artifactId>

9 </exclusion>

10 </exclusions>

　　　　2).添加以下slf4j和log4j2的依赖.

1 <!-- slf4j核心包-->

2 <dependency>

3 <groupId>org.slf4j</groupId>

4 <artifactId>slf4j-api</artifactId>

5 <version>1.7.13</version>

6 </dependency>

7 <dependency>

8 <groupId>org.slf4j</groupId>

9 <artifactId>jcl-over-slf4j</artifactId>

10 <version>1.7.13</version>

11 <scope>runtime</scope>

12 </dependency>

13

14 <!--核心log4j2jar包-->

15 <dependency>

16 <groupId>org.apache.logging.log4j</groupId>

17 <artifactId>log4j-api</artifactId>

18 <version>2.4.1</version>

19 </dependency>

20 <dependency>

21 <groupId>org.apache.logging.log4j</groupId>

22 <artifactId>log4j-core</artifactId>

23 <version>2.4.1</version>

24 </dependency>

25 <!--用于与slf4j保持桥接-->

26 <dependency>

27 <groupId>org.apache.logging.log4j</groupId>

28 <artifactId>log4j-slf4j-impl</artifactId>

29 <version>2.4.1</version>

30 </dependency>

31 <!--web工程需要包含log4j-web，非web工程不需要-->

32 <dependency>

33 <groupId>org.apache.logging.log4j</groupId>

34 <artifactId>log4j-web</artifactId>

35 <version>2.4.1</version>

36 <scope>runtime</scope>

37 </dependency>

38

39 <!--需要使用log4j2的AsyncLogger需要包含disruptor-->

40 <dependency>

41 <groupId>com.lmax</groupId>

42 <artifactId>disruptor</artifactId>

43 <version>3.2.0</version>

44 </dependency>

### 2.web.xml中设置log4j2的监听器和过滤器（servlet3.0及以上版本不需要该步操作）

1 <!--对于log4j2，Servlet2.5以前的版本需要-->

2 <listener>

3 <listener-class>org.apache.logging.log4j.web.Log4jServletContextListener</listener-class>

4 </listener>

5 <filter>

6 <filter-name>log4jServletFilter</filter-name>

7 <filter-class>org.apache.logging.log4j.web.Log4jServletFilter</filter-class>

8 </filter>

9 <filter-mapping>

10 <filter-name>log4jServletFilter</filter-name>

11 <url-pattern>/\*</url-pattern>

12 <dispatcher>REQUEST</dispatcher>

13 <dispatcher>FORWARD</dispatcher>

14 <dispatcher>INCLUDE</dispatcher>

15 <dispatcher>ERROR</dispatcher>

16 </filter-mapping>

　　注意：log4j2不再支持properties文件了，只支持xml，json或是yaml，不指定位置的情况下默认在src/main/resources下查找。

　　　　　如果需要自定义位置，需要在上面的web.xml中添加以下代码

1 <context-param>

2 <param-name>log4jConfiguration</param-name>

3 <param-value>/WEB-INF/classes/log4j2.xml</param-value>

4 </context-param>

### 3.log4j2.xml

1 <?xml version="1.0" encoding="UTF-8"?>

2

3 <Configuration status="off" monitorInterval="1800">

4

5 <properties>

6 <property name="LOG\_HOME">/opt/logs/hafiz/log4j2Demo/logs</property>

7 <property name="ERROR\_LOG\_FILE\_NAME">error</property>

8 </properties>

9

10

11 <Appenders>

12 <Console name="Console" target="SYSTEM\_OUT">

13 <PatternLayout pattern="%d %-5p (%F:%L) - %m%n" />

14 </Console>

15

16 <RollingRandomAccessFile name="ErrorLog"

17 fileName="${LOG\_HOME}/${ERROR\_LOG\_FILE\_NAME}.log"

18 filePattern="${LOG\_HOME}/${ERROR\_LOG\_FILE\_NAME}.log.%d{yyyy-MM-dd}.gz">

19 <PatternLayout

20 pattern="%d %-5p (%F:%L) - %m%n"/>

21 <Policies>

22 <TimeBasedTriggeringPolicy/>

23 <SizeBasedTriggeringPolicy size="100 MB"/>

24 </Policies>

25 <DefaultRolloverStrategy max="20"/>

26 </RollingRandomAccessFile>

27

28 </Appenders>

29

30 <Loggers>

31 <!-- 3rdparty Loggers -->

32 <logger name="org.springframework.core" level="info">

33 </logger>

34 <logger name="org.springframework.beans" level="info">

35 </logger>

36 <logger name="org.springframework.context" level="info">

37 </logger>

38 <logger name="org.springframework.web" level="info">

39 </logger>

40

41 <logger name="com.hafiz.www.controller" level="error" includeLocation="true" additivity="false">

42 <appender-ref ref="ErrorLog"/>

43 <appender-ref ref="Console"/>

44 </logger>

45

46 <root level="info" includeLocation="true">

47 <appender-ref ref="Console"/>

48 </root>

49 </Loggers>

50 </Configuration>

### 4.测试类UserController.java

1 package com.hafiz.www.controller;

2

3 import com.hafiz.www.po.UserEntity;

4 import com.hafiz.www.service.UserService;

5 import org.slf4j.Logger;

6 import org.slf4j.LoggerFactory;

7 import org.springframework.beans.factory.annotation.Autowired;

8 import org.springframework.stereotype.Controller;

9 import org.springframework.web.bind.annotation.RequestMapping;

10 import org.springframework.web.bind.annotation.RequestMethod;

11 import org.springframework.web.bind.annotation.ResponseBody;

12

13 import java.util.ArrayList;

14 import java.util.List;

15

16 /\*\*

17 \* Desc:用户信息控制器

18 \* Created by hafiz.zhang on 2016/8/27.

19 \*/

20 @Controller

21 @RequestMapping("/user")

22 public class UserController {

23

24 private static final Logger logger = LoggerFactory.getLogger(UserController.class);

25

26 @Autowired

27 private UserService userService;

28

29 @RequestMapping(value = "/all", method = RequestMethod.GET)

30 @ResponseBody

31 public List<UserEntity> getAllUsers(){

32 logger.info("[GET] /user/all getAllUsers");

33 List<UserEntity> list = userService.getAllUsers();

34 logger.debug("This is log of level of debug");

35 logger.trace("log4j2 Demo");

36 logger.error("哎呀，出错啦~");

37 return list;

38 }

39 }

注意：在JVM启动参数中增加 -DLog4jContextSelector=org.apache.logging.log4j.core.async.AsyncLoggerContextSelector 开启异步日志.

**小问题：**

**因为我之前是log4j 1.x，并且我在web.xml里面指定了log4j.properties文件的位置，第一次运行项目，忘了修改其值为log4j2.xml，出现了"Context [] startup failed due to previous errors"的错误，导致项目启动不起来，web.xml中修改为如下方式以后就正常了。**

1 <context-param>

2 <param-name>log4jConfiguration</param-name>

3 <param-value>/WEB-INF/classes/log4j2.xml</param-value>

4 </context-param>