**java 日志体系概述与应用**

* JAVA日志体系概述
* log4j2的配置与具体使用
* JAVA日志体系概述

**JAVA常用的日志打印组件**

|  |  |
| --- | --- |
| 名称 | 描述 |
| log4j |  |
| logback |  |
| log4j2 |  |
| java.util.looging |  |

**假设Spring 框架要打印日志，应该选择中其中哪一个组件?**

发现哪个都不能选，只能基于应用实际使用的日志组件来。不然就会日志打印会多份。

怎么找到应用实际使用的日志组件 Apache Commons Loging 解决了这个问题

**Apache Commons Loging （JCL）**

Commons Loging 本身只提供日志接口，具体实现在运行时动态寻找对应组件？比如：log4j、jdk14looger 等。但这种动态绑定的方式当系统特别宠大的时候会出现绑定失败的问题。现在比较流行的slf4j 基于静态绑定的方式解决了这个问题？

**slf4j**

sl4j 本身也只提供日志接口，与commons loging 不同的是其采用在classPath 加入以下jar包来表示具体采用哪种实现 ：

* slfj-log4j12.jar (表示指定 log4j)
* slf4j-jdk14.jar(表示指定jdk Looging)
* slf4j-jcl.jar(表示指定jcl)
* log4j-slf4j-impl(表示指定log4j2)
* logback-classic(表示指定logback)

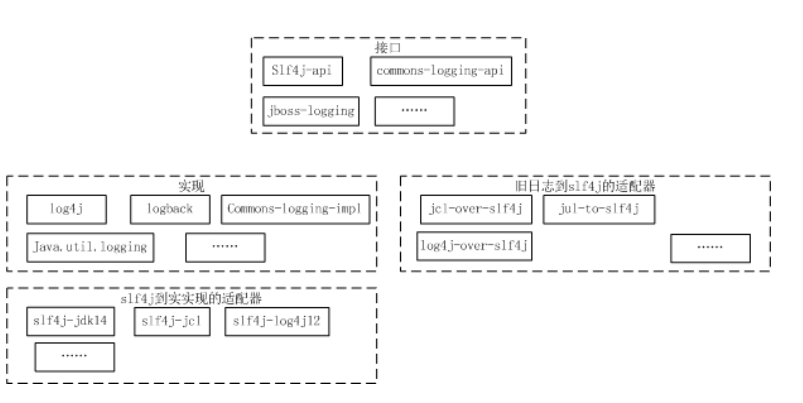
提问：

**假设你们系统当中之前在用 JCL 打印日志，但这时想加入slf4j来打印日志，就会出现两类日志输出如何解决?**

加入如下等JAR包jcl-over-slf4j、log4j-over-sl4j、jul-over-sl4j ，即可在使用原API的情况下，又统一至slf4j 的实现进行输出了。 .

**总结一下各日志组件的对应关系 ：**

JAVA日志体系结构图



**再来假设一个问题：**

如果我们把这两个包 jcl-over-slf4

j-1.5.11.jar 和 slf4j-jcl-1.5.11.jar 都放到 ClassPath 下会有什么情况呢？JCL 代理给 SLF4J，SLF4J 又绑定到 JCL。

**常用日志方案**

## log4j2的配置与具体使用

**log4j2 基于使用示例演示**

pom文件添加相关依懒

编写log4j2.xml 文件

编写Appenders

编写Loggers ，配置一个Root

编写打印示例

**log4j2 三种输出方式及相关配置**

**Console**

name

target

<PatternLayout pattern="%d [%-5level][%t] %m (%C:%F:%L) %n"> 详细介绍

**File**

fileName

append

RollingFile

fileName

filePattern

Policies

**loggers 配置**

Logger

AsyncLogger

root

AsyncRoot

问题：

如何压缩？

如何统一spring 日志的输出?