<http://blog.csdn.net/lifuxiangcaohui/article/details/7278595>

版权声明：本文为博主原创文章，未经博主允许不得转载。

log4j 大家都知道，就不在多说了，重点说说slf4j;

简单日记门面(simple logging Facade for java)SLF4J是为各种loging  
APIs提供一个简单统一的接口，从而使得最终用户能够在部署的时候配置自己希  
望的loging APIs实现

准确的说，slf4j并不是一种具体的日志系统，而是一个用户日志系统的  
facade，允许用户在部署最终应用时方便的变更其日志系统。  
在系统开发中，统一按照slf4j的API进行开发，在部署时，选择不同的日  
志系统包，即可自动转换到不同的日志系统上。比如：选择JDK自带的日志系  
统，则只需要将slf4j-api-1.5.10.jar和slf4j-jdk14-1.5.10.jar放置到  
classpath中即可，如果中途无法忍受JDK自带的日志系统了，想换成log4j的日  
志系统，仅需要用slf4j-log4j12-1.5.10.jar替换slf4j-jdk14-1.5.10.jar即可  
（当然也需要log4j的jar及配置文件）

SLF4J获得logger对象：  
private static final Logger logger = LoggerFactory.getLogger(Test.class);  
输出日志信息：  
logger.debug(“debug”);

LOG4J获得logger对象：

public class A {  
private static Logger logger = Logger.getLogger(A.class);  
}

|  |
| --- |
| 下面对slf4j和log4j做一下总结：  （1）大部分人在程序里面会去写logger.error(exception),其实这个时候log4j回去把这个exception tostring。真正的写法应该是logger(message.exception);而slf4j就不会使得程序员犯这个错误。   (2)log4j间接的在鼓励程序员使用string相加的写法，而slf4j就不会有这个问题  你可以使用logger.error("{} is+serviceid",serviceid);   （3）使用slf4j可以方便的使用其提供的各种集体的实现的jar。（类似commons-logger）   （4）从commons--logger和log4j merge非常方便，slf4j也提供了一个swing的tools来帮助大家完成这个merge。 |