# 主要特性

1、更快的执行速度  
基于先前的Log4j，Logback 重写了内部的实现，在某些特定的场景上，Logback可以比log4j的执行速度快上10倍。

2、更小的内存消耗  
在保证Logback的组件更快执行的同时，所需的内存更加的少；

3、充分的测试  
Logback 历经了几年时间的充分测试，这正是人们选择Logback而不是Log4j的重要原因。

# 构成

## **logback-core**

logback-core是其它模块的基础，其它模块基于它构建

## **logback-classic**

logback-classic类似于Log4J，它被认为是 Log4J的一个改进版，并且它实现了简单日志门面 SLF4J。

## **logback-access**

logback-access主要作为一个与 Servlet容器交互的模块，比如说tomcat或者 jetty，提供一些与 HTTP访问相关的功能。

# 日志等级

日志级别从低到高分为：  
TRACE < DEBUG < INFO < WARN < ERROR < FATAL

# 配置文件的精简结构



这个文件在springboot中默认叫做logback-spring.xml，我们只要新建一个同名文件放在resources下面，配置即可生效。

## contextName

每个logger都关联到logger上下文，默认上下文名称为“default”。但可以使用contextName标签设置成其他名字，用于区分不同应用程序的记录。

## property

用来定义变量值的标签，property标签有两个属性，name和value；其中name的值是变量的名称，value的值时变量定义的值。通过property定义的值会被插入到logger上下文中。定义变量后，可以使“${name}”来使用变量。

## logger

用来设置某一个包或者具体的某一个类的日志打印级别以及指定appender。

## root

根logger，也是一种logger，且只有一个level属性。

## appender

负责日志的组件

# 常用的appender的种类

## ConsoleAppender

把日志添加到控制台

## FileAppender

把日志添加到文件

## RollingFileAppender

滚动记录文件，先将日志记录到指定文件，当符合某个条件时，将日志记录到其他文件,它是FileAppender的子类。

# filter

## 基本概念

filter其实是appender里的子元素，它作为过滤器存在，执行一个过滤器会有返回DENY，NEUTRAL，ACCEPT三个枚举值中的一个。

DENY：日志将立即被抛弃不再经过其他过滤器；

NEUTRAL：有序列表里的下个过滤器过接着处理日志；

ACCEPT：日志会被立即处理，不再经过剩余过滤器。

## 常用的过滤器

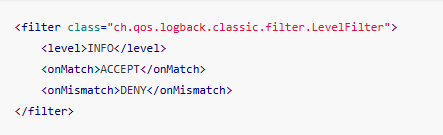
### ****ThresholdFilter****

临界值过滤器，过滤掉低于指定临界值的日志。当日志级别等于或高于临界值时，过滤器返回NEUTRAL；当日志级别低于临界值时，日志会被拒绝。



### ****LevelFilter****

级别过滤器，根据日志级别进行过滤。如果日志级别等于配置级别，过滤器会根据onMath(用于配置符合过滤条件的操作) 和 onMismatch(用于配置不符合过滤条件的操作)接收或拒绝日志。



# 问题

## 1、Exception in thread "main" java.lang.IllegalStateException: Logback configuration error detected: ERROR in ch.qos.logback.core.joran.spi.Interpreter@13:58 - RuntimeException in Action for tag [property] java.lang.StringIndexOutOfBoundsException: String index out of range: 16

### 解决方法

错误原因：

这个错误通常是由于 logback.xml 配置文件中的 <property> 标签中某个字符串的设置不正确导致的。特别是在 <property> 标签中，你可能指定了一个不正确的字符串（如不完整或格式错误的路径，或者变量的替换表达式中错误的索引位置）。

常见原因：

1、变量替换错误：Logback 支持在配置中使用变量（通过 property 标签定义），比如：

<property name="LOG\_HOME" value="/path/to/logs" />

如果你在配置中使用了 ${LOG\_HOME}/logfile.log，但 LOG\_HOME 没有正确被设置，或者路径的字符串格式错误，可能会导致索引越界异常。

2、路径错误：如果你在配置文件中使用了路径相关的配置，且路径字符串没有正确闭合或格式错误，也会导致该问题。例如，可能定义了一个路径并试图在该路径后追加文件名时，字符串索引超出了范围。

3、配置文件格式错误：如果 Logback 配置文件中出现了不匹配的 XML 标签、未闭合的属性或者错误的 XML 语法，也可能导致该错误。