## Log4j2配置

### Configuration

<Configuration status="WARN" monitorInterval="60">

Configuration为根节点，有两个属性status和monitorInterval

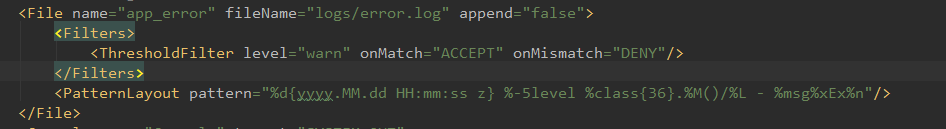
* Status:指定log4j2本身的日志信息打印级别。
* monitorInterval定时重新加载。项目中为60，即60秒重新加载一次。

### Appenders

Appenders标签下可配置多个appender，appender为日志输出的目的地。可以输出到控制台，输出到文件。可配置日志的输出级别，定义策略和日志输出格式。

RollingFile：当满足开发人员设定的策略条件时，可重命名日志文件用于备份，并重新生成一个新的日志文件。其中fileName属性定义生成日志文件的路径和文件名。filePattern定义文件进行备份时的保存路径和文件名。PatternLayout标签为日志输出格式的布局，标签下pattern标签定义日志输出的格式。%d:为日期时间，%t为当前线程，%level打印日志级别，%logger打印定义的logger的name，%msg为具体的日志信息，%n进行换行。(具体更加详细的打印格式，请参考<http://blog.csdn.net/guoquanyou/article/details/5689652>)。policies标签定义策略，如：日志一天进行一次备份，或超过20M进行备份。TimeBasedTriggeringPolicy定义时间策略，此策略的定义与RollingFile的filePattern属性有关，当filePattern属性的单位为天时(%d(MM-dd-yyyy))，TimeBasedTriggeringPolicy中的interval属性为1，那么日志进行备份时以天为单位进行备份，一天备份一次。当filePattern属性的单位为小时时(%d(MM-dd-yyyy HH))，TimeBasedTriggeringPolicy中的interval属性为1，那么日志进行备份时以小时为单位进行备份，一小时备份一次，以此类推。SizeBasedTriggeringPolicy基于大小策略，size属性定义文件的大小，当大于此值时，日志文件进行压缩备份，保存到filePattern指定的路径下。DefaultRolloverStrategy标签指定备份文件的存在数量。

File：日志打印到文件，filename属行定义日志路径和日志文件名，append属行指定是否追加到日志文件，默认为true。当设置为false时，之前的日志会被覆盖。Filters组合过滤器标签，子标签定义具体的过滤器，分别为日志等级过滤器，正则表达式过滤器，时间过滤器。ThresholdFilter日志级别过滤器，level定义日志级别，onMatch当满足定义的日志级别时如何处理，onMismatch当不满足定义的日志级别时如何处理。Filter有三个参数ACCEPT,NEUTRAL,DENY。ACCEPT直接处理，NEUTRAL暂不处理交给下面的filter处理，DENY直接放弃。



当满足日志级别为warn时，进行打印，不满足的直接放弃，即只有满足warn级别的日志时菜进行打印。RegexFilter正则表达式过滤器，条件为满足regex属性中的正则表达式。TimeFilter时间过滤器，满足start - end属性时间范围的日志。

Console的appender不做解释。

### Loggers

Root：root节点指定项目的根日志，如果没有单独指定logger，就默认使用该root日志输出。

AppenderRef:用来指定日志输出到哪个appender。Logger中没有引用则，将此日志输出到root指定的appender中。指定了则在指定的appender和root中的appender中都会输出。Additivity属性进行控制，值为false则只输出到指定的appender中。

Logger：用来单独指定日志的形式，如：指定某个包下的class指定不同的日志级别。Name用来指定改logger使用的类，或者类所在包的全路径。

### Logger level

共有8个级别，按照从低到高为：

All < Trace < Debug < Info < Warn < Error < Fatal < OFF.

All:最低等级的，用于打开所有日志记录.

Trace:是追踪，就是程序推进以下，你就可以写个trace输出，所以trace应该会特别多，不过没关系，我们可以设置最低日志级别不让他输出.

Debug:指出细粒度信息事件对调试应用程序是非常有帮助的.

Info:消息在粗粒度级别上突出强调应用程序的运行过程.

Warn:输出警告及warn以下级别的日志.

Error:输出错误信息日志.

Fatal:输出每个严重的错误事件将会导致应用程序的退出的日志.

OFF:最高等级的，用于关闭所有日志记录.