log4j2

1. 介绍

log4j2除了支持properties和xml形式的配置外，还添加了对json和yaml的支持：

> Log4J 2 introduces configuration support through JSON and YAML in addition to properties file and XML.

2. 实战

2.1 SpringBoot

2.1.1 properties文件

> classes路径下添加 `log4j2.properties` 即可自动识别；

内容示例：

```
status = info
name = PropertiesConfig

filters = threshold

filter.threshold.type = ThresholdFilter
filter.threshold.level = debug

appenders = console

appender.console.type = Console
appender.console.name = STDOUT
appender.console.layout.type = PatternLayout
appender.console.layout.pattern = %d{yyyy-MM-dd HH:mm:ss.SSS} %-5p %c{1}:%L - %m%n

rootLogger.level = error
rootLogger.appenderRefs = console
rootLogger.appenderRef.stdout.ref = STDOUT
```

2.1.1.1 maven依赖

```
!-- 忽略自带的日志框架. -->
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter</artifactId>
   <exclusions>
       <exclusion>
           <groupId>org.springframework.boot</groupId>
           <artifactId>spring-boot-starter-logging</artifactId>
      </exclusion>
   </exclusions>
</dependency>
<!-- log4j2 -->
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-log4j2</artifactId>
</dependency>
```

2.1.2 json配置

1. 在classpath路径下添加json文件，如：

```
{
  "configuration": {
    "status": "error",
    "name": "JSONConfigDemo",
    "packages": "com.howtodoinjava",
    "ThresholdFilter": {
      "level": "debug"
    },
    "appenders": {
      "Console": {
        "name": "STDOUT",
        "PatternLayout": {
          "pattern": "%d [%t] %-5p %c - %m%n"
        }
      }
    },
    "loggers": {
      "root": {
        "level": "error",
        "AppenderRef": {
          "ref": "STDOUT"
        }
      }
    }
  }
}
```

2. 在application.properties或application.yml中指定log4j2配置文件位置，如：

```
logging:
  config: classpath:log4j2.json
```

2.1.2.1 maven依赖

同2.1.1.1 maven依赖相同；

其他

参考博客

log4j2介绍及示例 - 推荐