**SpringBoot默认日志Logback：**

**默认**情况下SpringBoot会用Logback来记录日志，并用INFO级别输出到控制台。

C:\Users\caddy\AppData\Local\Temp\1524710291(1).png

日志信息如图分解如下：

时间日期：精确到毫秒

日志级别：ERROR,WARN,INFO,DEBUG or TRACE

进程ID

分隔符：---标识实际日志的开始

线程名：方括号括起来的（**可能会截断控制台输出**）

Logger名：通常使用源代码的类名

日志内容：

**Maven依赖**

SpringBoot默认的日志框架logback

Maven依赖是spring-boot-starter-logging

spring-boot-starter包含spring-boot-starter-logging，该部分就是SpringBoot默认的日志框架logback，所以不需要引入maven依赖

**日至级别：控制台输出**

TRACE<DEBUG<INFO<WARN<ERROR<FATAL

设为WARN则低于WARN的信息不会输出

SpringBoot默认配置ERROR，WARN，INFO级别的日志输出到控制台

**日志文件：**

默认情况SpringBoot只将日志输出到控制台，不会写入日志文件

Application.yml配置文件只能配置保存路径，日志格式，

复杂的区分INFO和ERROR日志，每天产生一个日志文件等场景满足不了，只能自定义配置.

首先application.yml配置日志，此时控制台输出 没有多彩字体

#-------------logging start----------#

**logging:**

**path**: c:/firstlog #日志路径

**file**: c:/firstlog/firstlog.log #日志文件

**level**: DEBUG #日志级别 注意要大写此时设置为

**pattern**: #日志输出的格式

**level**: DEBUG

**console**: '%d{yyyy-MM-dd HH:mm:ss.SSS} -%5p ${PID:-} [%15.15t] %-40.40logger{39} : %m%n'

**file**: '%d{yyyy-MM-dd HH:mm:ss.SSS} -%5p ${PID:-} [%15.15t] %-40.40logger{39} : %m%n'

#-------------logging end----------#

测试：

logger.info("测试INFO日志信息");//控制台，日志文件正常记录

logger.debug("测试DEBUG是否会写到file");//控制台不输出，日志文件不记录

logger.trace("测试trace是否会写到file");//级别低，控制台不输出，日志文件不记录

logger.warn("测试WARN是否会写到file");//控制台，日志文件正常记录

logger.error("测试ERROR是否会写到file");//控制台，日志文件正常记录

搜集控制台输出DEBUG的结果如下：

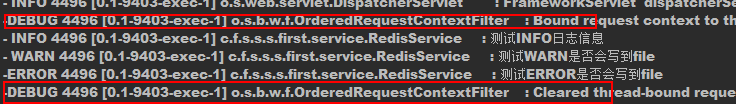
Application.yml配置

**debug**: true

#默认false 核心Logger（包含嵌入式容器，hibernate，spring）会输出更多内容，但是应用日志并不会输出为DEBUG级别

测试结果控制台输出，日志文件也会记录，并不是想要的输出结果。而且项目启动时会输出很多错误信息，但是不影响程序运行（暂时没发现影响）

结果为：



**日志配置文件logback-spring.xml**

Resources目录下

logback-spring.xml会覆盖掉application.yml内相同的配置

搜集到如下结果：

根据不同的日志系统，按如下规则组织配置文件名，放到src/main/resources下就能被正确加载。SpringBoot官方推荐优先使用带-spring的文件名，springboot可以为带-spring的文件添加特有的配置项。

Logback:

**logback-spring.xml**

**logback-spring.groovy**

logback.xml

logback.groovy

Log4j:

**log4j-spring.properties**

**log4j-spring.xml**

log4j.properties

log4j.xml

Log4j2:

**log4j2-spring.xml**

log4j2.xml

JDK(Java Util Logging):

logging.properties

**logback-spring.xml详解：**

具体在项目里配置文件查看src/mian/resources/logback-spring.xml

实现多环境，日志文件分割，将不同级别日志输出到不同文件，指定具体包或者类为指定级别日志，多配置几个appender

配置文件有详细注释

如果需要使用自定义日志文件，需要在application.yml里指定文件位置,指定方式是

**logging:**

**config: classpath:logging-my.xml**

如果想针对不同运行时Profile使用不同的日志配置，这个功能会很有用，不过直接在**logback-spring.xml**中使用springProfile配置，不需要logging.config指定不同环境使用不同配置文件。

**待完善mybatis sql输出**