https://blog.csdn.net/loongshawn/article/details/50951329

**SpringBoot配置log4j输出日志**

2016年03月21日 23:34:23 [loongshawn](https://me.csdn.net/loongshawn" \t "_blank) 阅读数：58240

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/loongshawn/article/details/50951329

**转载请注明来源：[http://blog.csdn.net/loongshawn/article/details/50951329](http://blog.csdn.net/loongshawn/article/details/50951329" \t "_blank)**

* [《SpringBoot添加Email发送功能》](http://blog.csdn.net/loongshawn/article/details/51036490)
* [《SpringBoot配置log4j输出日志》](http://blog.csdn.net/loongshawn/article/details/50951329)
* [《SpringBoot定时任务说明》](http://blog.csdn.net/loongshawn/article/details/50663393)
* [《SpringBoot接口服务处理Whitelabel Error Page》](http://blog.csdn.net/loongshawn/article/details/50915979)
* [《构建基于阿里云OSS文件上传服务》](http://blog.csdn.net/loongshawn/article/details/50710132)

**1、SpringBoot日志输出**

springboot也可以用logback日志输出，但之前用了段时间，感觉还是没有log4j的方便，可以设置每天输出一个日志。

**2、SpringBoot配置log4j**

Springboot配置log4j主要包括两个步骤：第一步，添加log4j.properties配置文件；第二步，初始化配置文件。

**2.1、配置log4j文件**

接下来看看具体的操作方法，添加log4j.properties配置文件，配置文件添加的路径是工程中的src/main/resources下面，如下图：



配置文件内容：

#log4j.rootLogger=CONSOLE,info,error,DEBUG

log4j.rootLogger=info,error,CONSOLE,DEBUG

log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender

log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout

log4j.appender.CONSOLE.layout.ConversionPattern=%d{yyyy-MM-dd-HH-mm} [%t] [%c] [%p] - %m%n

log4j.logger.info=info

log4j.appender.info=org.apache.log4j.DailyRollingFileAppender

log4j.appender.info.layout=org.apache.log4j.PatternLayout

log4j.appender.info.layout.ConversionPattern=%d{yyyy-MM-dd-HH-mm} [%t] [%c] [%p] - %m%n

log4j.appender.info.datePattern='.'yyyy-MM-dd

log4j.appender.info.Threshold = info

log4j.appender.info.append=true

#log4j.appender.info.File=/home/admin/pms-api-services/logs/info/api\_services\_info

log4j.appender.info.File=/Users/dddd/Documents/testspace/pms-api-services/logs/info/api\_services\_info

log4j.logger.error=error

log4j.appender.error=org.apache.log4j.DailyRollingFileAppender

log4j.appender.error.layout=org.apache.log4j.PatternLayout

log4j.appender.error.layout.ConversionPattern=%d{yyyy-MM-dd-HH-mm} [%t] [%c] [%p] - %m%n

log4j.appender.error.datePattern='.'yyyy-MM-dd

log4j.appender.error.Threshold = error

log4j.appender.error.append=true

#log4j.appender.error.File=/home/admin/pms-api-services/logs/error/api\_services\_error

log4j.appender.error.File=/Users/dddd/Documents/testspace/pms-api-services/logs/error/api\_services\_error

log4j.logger.DEBUG=DEBUG

log4j.appender.DEBUG=org.apache.log4j.DailyRollingFileAppender

log4j.appender.DEBUG.layout=org.apache.log4j.PatternLayout

log4j.appender.DEBUG.layout.ConversionPattern=%d{yyyy-MM-dd-HH-mm} [%t] [%c] [%p] - %m%n

log4j.appender.DEBUG.datePattern='.'yyyy-MM-dd

log4j.appender.DEBUG.Threshold = DEBUG

log4j.appender.DEBUG.append=true

#log4j.appender.DEBUG.File=/home/admin/pms-api-services/logs/debug/api\_services\_debug

log4j.appender.DEBUG.File=/Users/dddd/Documents/testspace/pms-api-services/logs/debug/api\_services\_debug

### Debug

log4j.logger.com.ibatis=DEBUG

log4j.logger.com.ibatis.common.jdbc.SimpleDataSource=DEBUG

log4j.logger.com.ibatis.common.jdbc.ScriptRunner=DEBUG

log4j.logger.com.ibatis.sqlmap.engine.impl.SqlMapClientDelegate=DEBUG

log4j.logger.java.sql.Connection=DEBUG

log4j.logger.java.sql.Statement=DEBUG

log4j.logger.java.sql.PreparedStatement=DEBUG

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46

上述log4j.properties文件中，当前配置为线下开发模式，若部署线上环境，需要释放掉注释语句，并注释线下配置语句。

#log4j.appender.info.File=/home/admin/pms-api-services/logs/info/api\_services\_info

log4j.appender.info.File=/Users/dddd/Documents/testspace/pms-api-services/logs/info/api\_services\_info

* 1
* 2
* 3

其中log4j.appender.info.File属性指明了输出日志文件的类型及路径和文件名，类型为info级别，路径为/Users/dddd/Documents/testspace/pms-api-services/logs/info/，输出的文件名为api\_services\_info。

日志配置文件中设置了三个级别的日志输出：info、error、debug，这些级别的日志输出，需要在java类中做相应的控制。

上述日志配置每天会产生一个新的日志文件。

**2.2、初始化log4j配置**

springboot启动是通过main入口函数，所以需要在main入口函数中设置log4j初始化。

public class App {

private static final Logger logger = LoggerFactory.getLogger(App.class);

@Value("${server.port}")

private int port;

@Value("${server.sessionTimeout}")

private int sessionTimeout;

static {

try{

// 初始化log4j

String log4jPath = "";

if(Constant.LOG\_MODE == 1){

// 配置线上地址

log4jPath = App.class.getClassLoader().getResource("").getPath()+"pms-api-services/config/log4j.properties";

logger.info("Log4j线上生产模式初始化。。。");

}else{

// 配置本地地址

log4jPath = App.class.getClassLoader().getResource("").getPath()+"log4j.properties";

logger.info("Log4j线下开发模式初始化。。。");

}

logger.info("初始化Log4j。。。。");

logger.info("path is "+ log4jPath);

PropertyConfigurator.configure(log4jPath);

}catch (Exception e){

logger.error(e.toString());

}

}

public static void main(String[] args) {

final String[] temp = args;

logger.info("oops: main入口函数编码-" +System.getProperty("file.encoding"));

if(Constant.LOG\_MODE == 0){

SpringApplication.run(App.class, args);

logger.info("oops: 线下开发测试");

}else{

logger.info("oops:" + args[0]);

}

if(ArrayUtils.isNotEmpty(args))

{

// 如果你的应用程序，运行后不自动退出，那么处理start时不要执行正常的代码，否则在部署测试 appctl.sh 的时候，会一直等待进程退出

if(args[0].equals("startup"))

{

new Thread(

new Runnable(){

public void run(){

System.out.println("启动Mythread子线程");

logger.info("启动Mythread子线程");

SpringApplication.run(App.class, temp);

}

}).start();

System.out.println("program startup");

logger.info("program startup");

}else if(args[0].equals("stop"))

{

System.out.println("program stop");

logger.info("program stop");

}else if(args[0].equals("restart"))

{

System.out.println("program restart");

logger.info("program restart");

}else if(args[0].equals("status"))

{

System.out.println("program status");

logger.info("program status");

}

}

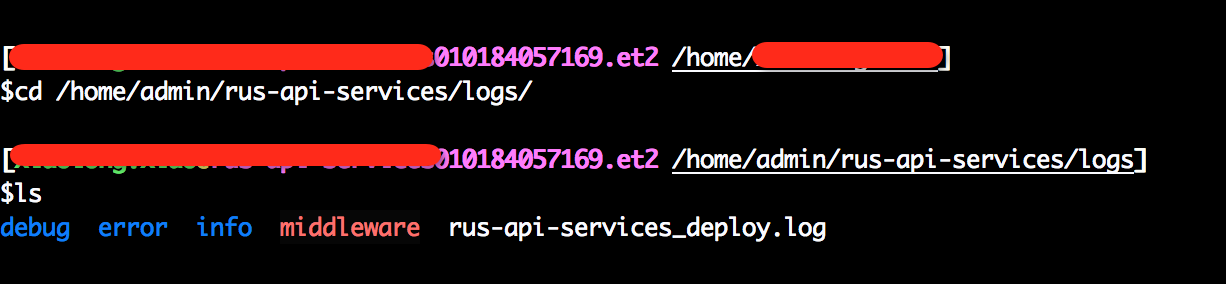
}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69
* 70
* 71
* 72
* 73
* 74
* 75

其中static代码片段就是完成初始化工单，本例中以我之前的工程作为参考，具体的路径还需依据每个人自己工程的实际路径来设置。

**3、log4j日志输出结果**

线上服务器日志目录



info文件夹中的日志



以上有什么错误的地方，欢迎大家指正。

**4、优化配置**

为了使配置更加灵活，对log4j.properties文件进行了优化，如下：

log4j.rootLogger=${log.root}

log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender

log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout

log4j.appender.CONSOLE.layout.ConversionPattern=%d{yyyy-MM-dd-HH-mm} [%t] [%c] [%p] - %m%n

log4j.logger.info=info

log4j.appender.info=org.apache.log4j.DailyRollingFileAppender

log4j.appender.info.layout=org.apache.log4j.PatternLayout

log4j.appender.info.layout.ConversionPattern=%d{yyyy-MM-dd-HH-mm} [%t] [%c] [%p] - %m%n

log4j.appender.info.datePattern='.'yyyy-MM-dd

log4j.appender.info.Threshold = info

log4j.appender.info.append=true

log4j.appender.info.File=${log.base}/pms-api-services/logs/info/api\_services\_info

log4j.logger.error=error

log4j.appender.error=org.apache.log4j.DailyRollingFileAppender

log4j.appender.error.layout=org.apache.log4j.PatternLayout

log4j.appender.error.layout.ConversionPattern=%d{yyyy-MM-dd-HH-mm} [%t] [%c] [%p] - %m%n

log4j.appender.error.datePattern='.'yyyy-MM-dd

log4j.appender.error.Threshold = error

log4j.appender.error.append=true

log4j.appender.error.File=${log.base}/pms-api-services/logs/error/api\_services\_error

log4j.logger.DEBUG=DEBUG

log4j.appender.DEBUG=org.apache.log4j.DailyRollingFileAppender

log4j.appender.DEBUG.layout=org.apache.log4j.PatternLayout

log4j.appender.DEBUG.layout.ConversionPattern=%d{yyyy-MM-dd-HH-mm} [%t] [%c] [%p] - %m%n

log4j.appender.DEBUG.datePattern='.'yyyy-MM-dd

log4j.appender.DEBUG.Threshold = DEBUG

log4j.appender.DEBUG.append=true

log4j.appender.DEBUG.File=${log.base}/pms-api-services/logs/debug/api\_services\_debug

### Debug

log4j.logger.com.ibatis=DEBUG

log4j.logger.com.ibatis.common.jdbc.SimpleDataSource=DEBUG

log4j.logger.com.ibatis.common.jdbc.ScriptRunner=DEBUG

log4j.logger.com.ibatis.sqlmap.engine.impl.SqlMapClientDelegate=DEBUG

log4j.logger.java.sql.Connection=DEBUG

log4j.logger.java.sql.Statement=DEBUG

log4j.logger.java.sql.PreparedStatement=DEBUG

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42

与之相应的main函数新增如下代码，同时去掉static代码块：

System.setProperty("log.root",Constant.LOG\_ROOT\_DAILY);

System.setProperty("log.base",Constant.LOG\_PATH\_DAILY);

* 1
* 2
* 3

**未来的开发工作中建议直接使用logback，目前log4j配置需要为线下、日常、预发、生产4个环境分别指定输出路径，比较折腾，使用logback指定一个相对路径即可，4个环境通用，比较省事。**

 收藏

 分享