http://temp.repia.com/dhan/openapiclient-0.0.1-SNAPSHOT.jar.gz

```
TB_ORGAN
CREATE CACHED TABLE "PUBLIC"."TB_ORGAN"(
    "OFCFS_CODE" VARCHAR(16),
    "OFCPS_NM" VARCHAR(100),
    "DEPT_CODE" VARCHAR(16),
    "DEPT_NM" VARCHAR(100),
    "USER_NM" VARCHAR(100),
    "OFFM_TELNO" VARCHAR(32),
    "EMAIL" VARCHAR(100),
    "JOB_DC" CLOB,
    "ORDR" NUMBER
);
```

```
java -Dlogback.configurationFile=resources/logback.xml  -cp .:./resources/ -jar openapiclient-0.0.1-SNAPSHOT.jar HRD ORGAN refresh
```

```
#================================================================================
#   api.properties : 시스템
#================================================================================
#   1.  key = value 구조입니다.
#   2.  key값은 공백문자를 포함불가, value값은 공백문자를 가능
#   3.  key값으로 한글을 사용불가,    value값은 한글사용이 가능
#  4. 줄을 바꿀 필요가 있으면 '\'를 라인의 끝에 추가(만약 '\'문자를 사용해야 하는 경우는 '||'를 사용)
#   5.  Windows에서의 디렉토리 표시 : '\\' or '/'  ('\' 사용하면 안됨)
#   6.  Unix에서의 디렉토리 표시 : '/'
#  7. 주석문 처리는  #사용
#   8.  value값 뒤에 스페이스가 존재하는 경우 서블릿에서 참조할때는 에러발생할 수 있으므로 trim()하거나 마지막 공백없이 properties 값을 설정할것
#================================================================================

database.type=h2
database.driver=org.h2.Driver
database.jdbcurl=jdbc:h2:tcp://127.0.0.1:9092/~/rsa41
database.user=sa
database.password=fpvldk!@#

ORGAN.table=TB_ORGAN
```

```
logback.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
 [Layout]
 %m : 로그내용이 출력
 %p : trace > debug > info > warn > error 등의 priority 출력
 %r : 어플리케이션이 시작되어 로깅이벤트가 발생하는 시점까지의 경과시간을 밀리세컨드로 출력
 %c : 예) 카테고리가 a.b.c 처럼 되어있다면 %c{2}는 b.c가 출력됩니다.
 %n :  플랫폼 종속적인 개행문자가 출력된다. \r\n 또는 \n 일것이다
 %d : 로깅이벤트가 일어나 날짜 출력 ( 프로그램의 실행속도를 느리게 한다.)
```

```
 예) %d{HH:mm:ss} 또는 %d{dd MMMM yyyy HH:mm:ss}
 %C : 호출자의 클래스명 출력
 예) 클래스구조가 org.apache.xyz.SomeClass 처럼 되어있다면 %C{2}는 xyz.SomeClass 가
출력됩니다
 %M : 로깅이 발생한 method 이름을 나타냅니다.
 %F : 로깅이 발생한 프로그램 파일명을 나타냅니다.
 %l : 로깅이 발생한 caller의 정보를 나타냅니다
 %L : 로깅이 발생한 caller의 라인수를 나타냅니다
 %x : 로깅이 발생한 thread와 관련된 NDC(nested diagnostic context)를 출력합니다.
 %X : 로깅이 발생한 thread와 관련된 MDC(mapped diagnostic context)를 출력합니다.
 %%: % 표시를 출력하기 위해 사용한다.
 %t : 로그이벤트가 발생된 쓰레드의 이름을 출력합니다
-->

<configuration scan="true" scanPeriod="30 seconds">
    <appender name="ROLLING"
class="ch.qos.logback.core.rolling.RollingFileAppender">
        <rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
            <!-- rollover daily -->
            <fileNamePattern>logs/openapiclient-%d{yyyy-MM-
dd}.%i.log.zip</fileNamePattern>
            <timeBasedFileNamingAndTriggeringPolicy
class="ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP">
                <!-- or whenever the file size reaches 100MB -->
                <maxFileSize>100MB</maxFileSize>
            </timeBasedFileNamingAndTriggeringPolicy>
        </rollingPolicy>
        <encoder>
            <pattern>%d{HH:mm:ss.SSS} %-5level %logger{36} -
%msg%n</pattern>
        </encoder>
    </appender>
    <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
        <layout class="ch.qos.logback.classic.PatternLayout">
            <Pattern>%d{HH:mm:ss.SSS} [%-5level] - %msg%n</Pattern>
        </layout>
    </appender>

    <!-- Loggers -->
    <logger name="org.apache.catalina" level="ERROR">
    </logger>

    <logger name="org.apache.commons" level="ERROR">
    </logger>

    <logger name="org.springframework" level="DEBUG" >
    </logger>

    <logger name="java.sql" level="DEBUG">
    </logger>
```

```
    <logger name="org.mybatis.spring" level="DEBUG">
    </logger>

    <root level="ERROR">
        <appender-ref ref="ROLLING"/>
        <appender-ref ref="STDOUT" />
    </root>
</configuration>
```

From:
http://rwiki.repia.com/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://rwiki.repia.com/doku.php?id=wiki:user:dhan:temp&rev=1630414443

Last update: **2022/03/10 19:52**