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>
```

```
rssStatus.sh
#!/bin/sh

trim() { echo $1; }

RSS_ROOT="/app/search/RSA"

echo "";

EXE_NAME="전자사전 관리데몬"
EXE_FILE="DIC_MGR"
PID_FILE="${RSS_ROOT}/logs/dic_mgr.pid"

if [ -f ${PID_FILE} ]
then
echo -n "";
#echo "${PID_FILE} exist!"
else
echo "${PID_FILE} not exist!"
fi

PID=$(trim `cat $PID_FILE`)
RUN_PID=$(ps ax |grep ${EXE_FILE} |grep -v grep|awk '{print $1}')

if [ ${PID} -eq ${RUN_PID} ]
then
echo "  * ${EXE_NAME}(${EXE_FILE}:${PID})";
else
echo "  * ${EXE_NAME}(${EXE_FILE}:Dead)";
fi

EXE_NAME="전자사전"
EXE_FILE="DicSvr"
PID_FILE="${RSS_ROOT}/logs/dic.pid"

if [ -f ${PID_FILE} ]
then
echo -n "";
#echo "${PID_FILE} exist!"
else
echo "${PID_FILE} not exist!"
```

```
fi

PID=$(trim `cat $PID_FILE`)
RUN_PID=$(ps ax |grep ${EXE_FILE} |grep -v grep|awk '{print $1}')

if [ ${PID} -eq ${RUN_PID} ]
then
echo "  * ${EXE_NAME}(${EXE_FILE}:${PID})";
else
echo "  * ${EXE_NAME}(${EXE_FILE}:Dead)";
fi

EXE_NAME="질의어 처리기 관리데몬"
EXE_FILE="QP_MGR"
PID_FILE="${RSS_ROOT}/logs/qp_mgr.pid"

if [ -f ${PID_FILE} ]
then
echo -n "";
#echo "${PID_FILE} exist!"
else
echo "${PID_FILE} not exist!"
fi

PID=$(trim `cat $PID_FILE`)
RUN_PID=$(ps ax |grep ${EXE_FILE} |grep -v grep|awk '{print $1}')

if [ ${PID} -eq ${RUN_PID} ]
then
echo "  * ${EXE_NAME}(${EXE_FILE}:${PID})";
else
echo "  * ${EXE_NAME}(${EXE_FILE}:Dead)";
fi


EXE_NAME="질의어 처리기"
EXE_FILE="QuerySvr"
PID_FILE="${RSS_ROOT}/logs/1.pid"

if [ -f ${PID_FILE} ]
then
echo -n "";
#echo "${PID_FILE} exist!"
else
echo "${PID_FILE} not exist!"
fi

PID=$(trim `cat $PID_FILE`)
RUN_PID=$(ps ax |grep ${EXE_FILE} |grep -v grep|awk '{print $1}')

if [ ${PID} -eq ${RUN_PID} ]
```

```
then
echo "  * ${EXE_NAME}(${EXE_FILE}:${PID})";
else
echo "  * ${EXE_NAME}(${EXE_FILE}:Dead)";
fi


EXE_NAME="스케줄러"
EXE_FILE="RSS_DAEMON"
PID_FILE="${RSS_ROOT}/logs/rssd.pid"

if [ -f ${PID_FILE} ]
then
echo -n "";
#echo "${PID_FILE} exist!"
else
echo "${PID_FILE} not exist!"
fi

PID=$(trim `cat $PID_FILE`)
RUN_PID=$(ps ax |grep ${EXE_FILE} |grep -v grep|awk '{print $1}')

if [ ${PID} -eq ${RUN_PID} ]
then
echo "  * ${EXE_NAME}(${EXE_FILE}:${PID})";
else
echo "  * ${EXE_NAME}(${EXE_FILE}:Dead)";
fi


EXE_NAME="관리자 서비스"
EXE_FILE="rssAdmMgr"
PID_FILE="${RSS_ROOT}/logs/rssAdmMgr.pid"

if [ -f ${PID_FILE} ]
then
echo -n "";
#echo "${PID_FILE} exist!"
else
echo "${PID_FILE} not exist!"
fi

PID=$(trim `cat $PID_FILE`)
RUN_PID=$(ps ax |grep ${EXE_FILE} |grep -v grep|awk '{print $1}')

if [ ${PID} -eq ${RUN_PID} ]
then
echo "  * ${EXE_NAME}(${EXE_FILE}:${PID})";
else
echo "  * ${EXE_NAME}(${EXE_FILE}:Dead)";
fi
```

```
EXE_NAME="부가기능 서비스"
EXE_FILE="org.h2.tools.Server"
PID_FILE="${RSS_ROOT}/logs/db.pid"

RUN_PID=$(ps ax |grep ${EXE_FILE} |grep -v grep|awk '{print $1}')
#echo ${RUN_PID}

if [ "" != "${RUN_PID}" ]
then
echo "  * ${EXE_NAME}(${EXE_FILE}:${RUN_PID})";
else
echo "  * ${EXE_NAME}(${EXE_FILE}:Dead)";
fi

EXE_NAME="검색어 자동완성"
EXE_FILE="AutoCompleteServer.jar"
PID_FILE="${RSS_ROOT}/logs/AutoComplete.pid"

RUN_PID=$(ps ax |grep ${EXE_FILE} |grep -v grep|awk '{print $1}')
#echo ${RUN_PID}

if [ "" != "${RUN_PID}" ]
then
echo "  * ${EXE_NAME}(${EXE_FILE}:${RUN_PID})";
else
echo "  * ${EXE_NAME}(${EXE_FILE}:Dead)";
fi



echo -e "\n\n";
```

```
backup_daily.sh
#!/bin/sh
datetime=$(date +%Y%m%d)
datetimeAgo=`date +%Y%m%d --date '30 days ago'`

echo "cd /app/search/RSA/bin"
cd /app/search/RSA/bin

echo "./RssConfBackup /app/search/RSA"
./RssConfBackup /app/search/RSA 1>/dev/null

echo "cd /app/search/RSA/backup/$datetime"
cd /app/search/RSA/backup/$datetime

#echo "mysqldump -u search -psearch12#$ AD > AD_$datetime.sql"
#mysqldump -u search -psearch12#$ AD > AD_$datetime.sql

echo "cd .."
cd ..
```

```
echo "tar cvf ./hrd176_$datetime.tar.gz ./$datetime"
tar cfz ./hrd176_$datetime.tar.gz ./$datetime

#echo "cp -f ./hrd176_$datetime.tar.gz /data/backup"
#cp -f ./hrd176_$datetime.tar.gz /data/backup/

#echo "rm -rf ./hrd176_$datetime"
#rm -rf ./$datetime

echo "rm -rf ./hrd176_$datetimeAgo.tar.gz"
rm -rf ./hrd176_$datetimeAgo.tar.gz

#echo "rm -rf /data/backup/hrd176_$datetimeAgo.tar.gz"
#rm -rf /data/backup/hrd176_$datetimeAgo.tar.gz

#echo "cp -f  /app/search/RSA/bin/html/js/* /data/bin/html/js"
#cp -f  /app/search/RSA/bin/html/js/* /data/bin/html/js
echo "cp -Rf /app/search/RSA/conf/* /data/conf"
#cp -Rf /app/search/RSA/conf/* /data/conf
#echo "cp -f /app/search/RSA/logs/front/search_log/logAnal.out
/data/logs/front/search_log"
#cp -f /app/search/RSA/logs/front/search_log/logAnal.out
/data/logs/front/search_log
#echo "cp -f /app/search/RSA/logs/front/statistics_log/favor*
/data/logs/front/statistics_log"
#cp -f /app/search/RSA/logs/front/statistics_log/favor*
/data/logs/front/statistics_log
```

```
backup_lmonthly.sh
#!/bin/sh
datetime=$(date +%Y%m%d)
datemonthAgo=`date +%Y%m --date '150 days ago'`
datemonth=$(date +%Y%m)

echo "cd /app/search/RSA/backup"
cd /app/search/RSA/backup

echo "cp hrd176_$datetime.tar.gz hrd176_$datemonth.tar.gz"
cp hrd176_$datetime.tar.gz hrd176_$datemonth.tar.gz 1>/dev/null 2>/dev/null

echo "rm -rf hrd176_$datemonthAgo.tar.gz"
rm -rf hrd176_$datetimeAgo.tar.gz
```

```
crontab 내용
# Recommend Dictionary update
0 5 * * * /app/search/RSA/bin/acsctl.sh restart
# Real-time favorite keyword
0 * * * * /app/search/RSA/bin/realRank.sh
#
# backup script
```

```
#30 6 * * * /app/search/RSA/bin/backup_ldaily.sh
0 9 * * 7  /app/search/RSA/bin/apispider/apiSpider.sh
0 7 1 * *  /app/search/RSA/bin/backup_lmonthly.sh
```

From:
http://125.132.25.164/dokuwiki/ - 문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:user:dhan:temp&rev=1630470170

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