```json
{
    "editor.suggestSelection": "first",
    "vsintellicode.modify.editor.suggestSelection":
"automaticallyOverrodeDefaultValue",
    "workbench.colorTheme": "Material Theme Darker High Contrast",
    "workbench.iconTheme": "eq-material-theme-icons",
    "editor.formatOnSave": true,
    "javascript.preferences.quoteStyle": "single",
    "typescript.preferences.quoteStyle": "single",
    "editor.tabSize": 2,
    "liveServer.settings.donotVerifyTags": true,
    "liveServer.settings.donotShowInfoMsg": true,
    "security.workspace.trust.untrustedFiles": "open",
    "files.exclude": {
        "**/.git": false
    },
    "git.autofetch": true,
    "terminal.integrated.defaultProfile.windows": "Git Bash",
    "terminal.integrated.automationShell.linux": "",
    "terminal.integrated.automationShell.windows": "",
    "terminal.integrated.profiles.windows": {

        "PowerShell": {
            "source": "PowerShell",
            "icon": "terminal-powershell"
        },
        "Command Prompt": {
            "path": [
                "${env:windir}\\Sysnative\\cmd.exe",
                "${env:windir}\\System32\\cmd.exe"
            ],
            "args": [],
            "icon": "terminal-cmd"
        },
        "Git Bash": {
            "source": "Git Bash"
        }
    }
}
```

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

```sql
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),
```

```sql
    "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 :
#-------------------------------------------------------------------
#   1.   key = value           .
#   2.   key                   ,  value
#   3.   key                   ,    value
#   4.                   '\'                (     '\'                        '\\'
   )
#   5.   Windows              : '\\' or '/'  ('\'            )
#   6.   Unix               :'/
#   7.              #
#   8.   value
trim()                      properties
#-------------------------------------------------------------------

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
```

```xml
logback.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
 [Layout]
 %m :
 %p : trace > debug > info > warn > error       priority
 %r :
 %c :   )             a.b.c                  %c{2}   b.c              .
 %n :                                 . \r\n       \n
 %d :                       (                                    .)
   ) %d{HH:mm:ss}      %d{dd MMMM yyyy HH:mm:ss}
 %C :
   )            org.apache.xyz.SomeClass              %C{2}   xyz.SomeClass

 %M :            method                 .
 %F :                                   .
 %l :            caller
 %L :            caller
 %x :            thread        NDC(nested diagnostic context)            .
 %X :            thread        MDC(mapped diagnostic context)            .
 %% : %                                .
```

```
 %t :
-->

<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

```sh
#!/bin/sh

trim() { echo $1; }

RSS_ROOT="/app/search/RSA"

echo "";

EXE_NAME="                    "
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="          "
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="                    "
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="               "
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="          "
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="            "
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="              "
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="             "
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"
```

```sh
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
```

```sh
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
```

```sh
crontab
# 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
```