

Log Management

Summary

Log Management provides the function to search and inquire various logs that are generated during system use.

Description

Log inquiry accompanies the function of **registration, inquiry, list, deletion and summary** of system log.

- ① Register Log: register log information. – use AOP function
- ② Inquire Log: view detailed contents of log information.
- ③ Log List: search and view the list of log information.
- ④ Delete Log: delete log information. – use Scheduling function of execution environment
- ⑤ Log Summary: summarize log information and create Summary. - use Scheduling function of execution environment

Package Dependency

Log Management package direct functional dependency to the common package (cmm) of element technology only. However, it forms the deployment file together with calendar package according to dependency between packages so as to be executed without error during component deployment.

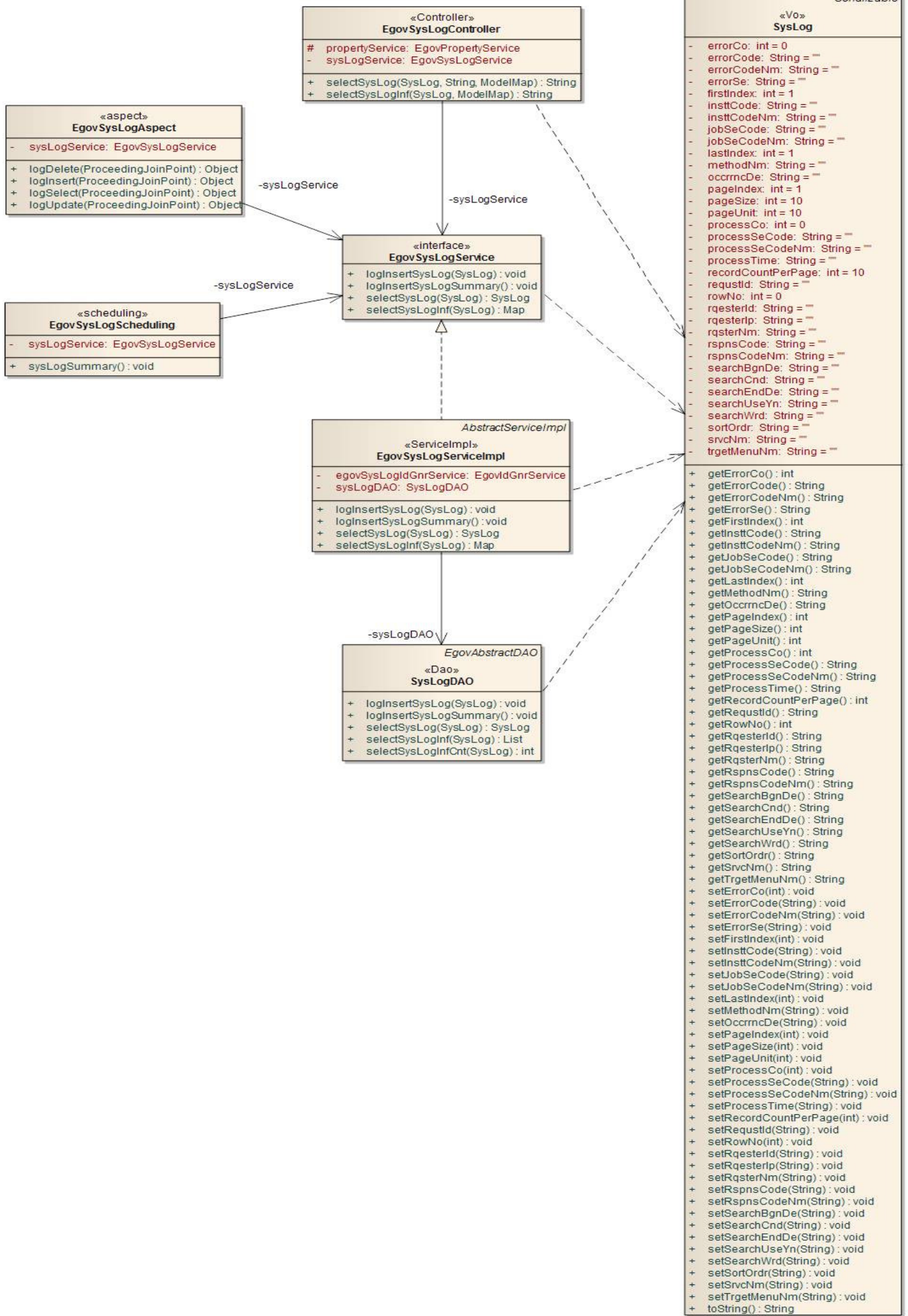
- Package Dependency: [System Management Package Dependency](#)

Related Source

Type	Source	Remarks
Controller	egovframework.com.sym.log.lgm.web.EgovSysLogController.java	Controller class for Log Management
Service	egovframework.com.sym.log.lgm.service.EgovSysLogService.java	Service interface for Log Management
ServiceImpl	egovframework.com.sym.log.lgm.service.impl.EgovSysLogServiceImpl.java	Service implementation class for log management
Model	egovframework.com.sym.log.lgm.service.SysLog.java	Model class for Log Management
DAO	egovframework.com.sym.log.lgm.service.impl.LogManageDAO.java	Data processing class for Log Management
Aspect	egovframework.com.sym.log.lgm.service.EgovLogManageAspect.java	Aspect class for log management
Scheduler	egovframework.com.sym.log.lgm.service.EgovLogManageScheduling.java	Scheduling class for log deletion and summary
JSP	/WEB-INF/jsp/egovframework/com/sym/log/lgm/EgovSysLogList.jsp	Jsp page for log list
JSP	/WEB-INF/jsp/egovframework/com/sym/log/lgm/EgovSysLogInquire.jsp	Jsp page for log inquiry
QUERY XML	resources/egovframework/sqlmap/com/sym/log/lgm/EgovSysLog_SQL_Mysql.xml	Log Management MySQL based QUERY XML
QUERY XML	resources/egovframework/sqlmap/com/sym/log/lgm/EgovSysLog_SQL_Oracle.xml	Log Management Oracle based QUERY XML
QUERY XML	resources/egovframework/sqlmap/com/sym/log/lgm/EgovSysLog_SQL_Tibero.xml	Log Management Tibero based QUERY XML
QUERY	resources/egovframework/sqlmap/com/sym/log/lgm/EgovSysLo	Log Management Altibase

XML	g_SQL_Altibase.xml	based QUERY XML
Idgen XML	resources/egovframework/spring/context-idgen.xml	Log Management Id generation Idgen XML

Class Diagram



ID Generation

ID Generation Related DDL and DML

- To utilize ID Generation Service, **SYSLOG_ID** is added to the Sequence storage table, COMTECOPSEQ.

```
table_name varchar(16) NOT NULL,  
next_id DECIMAL(30) NOT NULL,  
PRIMARY KEY (table_name));
```

```
INSERT INTO COMTECOPSEQ VALUES('SYSLOG_ID','0');
```

ID Generation Configuration (context-idgen.xml)

```
<bean name="egovSysLogIdGnrService"  
  class="egovframework.rte.fdl.idgnr.impl.EgovTableIdGnrService"  
  destroy-method="destroy">  
  <property name="dataSource" ref="dataSource" />  
  <property name="strategy" ref="sysLogStrategy" />  
  <property name="blockSize" value="1"/>  
  <property name="table" value="COMTECOPSEQ"/>  
  <property name="tableName" value="SYSLOG_ID"/>  
</bean>  
  
<bean name="sysLogStrategy"  
  class="egovframework.rte.fdl.idgnr.impl.strategy.EgovIdGnrStrategyImpl">  
  <property name="prefix" value="SYSLOG_" />  
  <property name="cipers" value="13" />  
  <property name="fillChar" value="0" />  
</bean>
```

Table

Table Name	Table Name(English)	Remarks
System Log	COMTNSYSLOG	Manage system log information.
System Log Summary	COMTSSYSLOGSUMMARY	Manage system log summary information.

AOP

context-syslogaop.xml

```
<bean id="syslog" class="egovframework.com.sym.log.lgm.service.EgovSysLogAspect" />  
  
<aop:config>  
  <aop:aspect id="sysLogAspect" ref="syslog">  
    <!-- insert 로 시작되는 service Method -->  
    <aop:around pointcut="execution(public *  
egovframework.com..impl.*Impl.insert*(..))" method="logInsert" />  
    <!-- update 로 시작되는 service Method -->  
    <aop:around pointcut="execution(public *  
egovframework.com..impl.*Impl.update*(..))" method="logUpdate" />  
    <!-- delete 로 시작되는 service Method -->  
    <aop:around pointcut="execution(public *  
egovframework.com..impl.*Impl.delete*(..))" method="logDelete" />  
    <!-- select 로 시작되는 service Method -->
```

```

        <aop:around pointcut="execution(public *
egovframework.com..impl.*Impl.select*(..))" method="logSelect" />
    </aop:aspect>
</aop:config>

```

Set [AOP](#) for implementing the function to register system log.

- Create EgovLogManageAspect class to implement the function to register system log

```
package egovframework.com.sym.log.lgm.service;
```

```
import javax.annotation.Resource;
```

```
import org.aspectj.lang.ProceedingJoinPoint;
import org.springframework.util.StopWatch;
```

```
import egovframework.com.cmm.LoginVO;
import egovframework.com.cmm.util.EgovUserDetailsHelper;
```

```
public class EgovSysLogAspect {
```

```
    @Resource(name="EgovSysLogService")
    private EgovSysLogService sysLogService;
```

```
    /**
```

```
     * Create the system log information.
     * Method that starts with insert of sevice class
     *
```

```
     * @param ProceedingJoinPoint
```

```
     * @return Object
```

```
     * @throws Exception
```

```
    */
```

```
    public Object logInsert(ProceedingJoinPoint joinPoint) throws Throwable {
```

```
        StopWatch stopWatch = new StopWatch();
```

```
        try {
```

```
            stopWatch.start();
```

```
            Object retValue = joinPoint.proceed();
```

```
            return retValue;
```

```
        } catch (Throwable e) {
```

```
            throw e;
```

```
        } finally {
```

```
            stopWatch.stop();
```

```
            SysLog sysLog = new SysLog();
```

```
            String className = joinPoint.getTarget().getClass().getName();
```

```
            String methodName = joinPoint.getSignature().getName();
```

```
            String processSeCode = "C";
```

```
            String processTime = Long.toString(stopWatch.getTotalTimeMillis());
```

```
            String uniqId = "";
```

```
            String ip = "";
```

```
            /* Authenticated */
```

```
            Boolean isAuthenticated = EgovUserDetailsHelper.isAuthenticated();
```

```
            if(isAuthenticated.booleanValue()) {
```

```
                LoginVO user =
```

```
(LoginVO)EgovUserDetailsHelper.getAuthenticatedUser();
```

```
                uniqId = user.getUniqId();
```

```
                ip = user.getIp();
```

```
            }
```

```

        sysLog.setSrvcNm(className);
        sysLog.setMethodNm(methodName);
        sysLog.setProcessSeCode(processSeCode);
        sysLog.setProcessTime(processTime);
        sysLog.setRqesterId(uniqId);
        sysLog.setRqesterIp(ip);

        sysLogService.logInsertSysLog(sysLog);
    }
}

/**
 * Create system log information.
 * Method that starts with update of sevice Class
 *
 * @param ProceedingJoinPoint
 * @return Object
 * @throws Exception
 */
public Object logUpdate(ProceedingJoinPoint joinPoint) throws Throwable {
    Stopwatch stopWatch = new Stopwatch();

    try {
        stopWatch.start();

        Object retValue = joinPoint.proceed();
        return retValue;
    } catch (Throwable e) {
        throw e;
    } finally {
        stopWatch.stop();

        SysLog sysLog = new SysLog();
        String className = joinPoint.getTarget().getClass().getName();
        String methodName = joinPoint.getSignature().getName();
        String processSeCode = "U";
        String processTime = Long.toString(stopWatch.getTotalTimeMillis());
        String uniqId = "";
        String ip = "";

        /* Authenticated */
        Boolean isAuthenticated = EgovUserDetailsHelper.isAuthenticated();
        if(isAuthenticated.booleanValue()) {
            LoginVO user =
(LoginVO)EgovUserDetailsHelper.getAuthenticatedUser();
            uniqId = user.getUniqId();
            ip = user.getIp();
        }

        sysLog.setSrvcNm(className);
        sysLog.setMethodNm(methodName);
        sysLog.setProcessSeCode(processSeCode);
        sysLog.setProcessTime(processTime);
        sysLog.setRqesterId(uniqId);
        sysLog.setRqesterIp(ip);

        sysLogService.logInsertSysLog(sysLog);
    }
}

```

```

}

/**
 * Create the system log information.
 * Method that starts with delete of sevice Class
 *
 * @param ProceedingJoinPoint
 * @return Object
 * @throws Exception
 */
public Object logDelete(ProceedingJoinPoint joinPoint) throws Throwable {

    Stopwatch stopWatch = new Stopwatch();

    try {
        stopWatch.start();

        Object retValue = joinPoint.proceed();
        return retValue;
    } catch (Throwable e) {
        throw e;
    } finally {
        stopWatch.stop();

        SysLog sysLog = new SysLog();
        String className = joinPoint.getTarget().getClass().getName();
        String methodName = joinPoint.getSignature().getName();
        String processSeCode = "D";
        String processTime = Long.toString(stopWatch.getTotalTimeMillis());
        String uniqId = "";
        String ip = "";

        /* Authenticated */
        Boolean isAuthenticated = EgovUserDetailsHelper.isAuthenticated();
        if(isAuthenticated.booleanValue()) {
            LoginVO user =
(LoginVO)EgovUserDetailsHelper.getAuthenticatedUser();
            uniqId = user.getUniqId();
            ip = user.getIp();
        }

        sysLog.setSrvNm(className);
        sysLog.setMethodNm(methodName);
        sysLog.setProcessSeCode(processSeCode);
        sysLog.setProcessTime(processTime);
        sysLog.setRqesterId(uniqId);
        sysLog.setRqesterIp(ip);

        sysLogService.logInsertSysLog(sysLog);
    }
}

/**
 * Create the system log information.
 * Method that starts with select of sevice Class
 *
 * @param ProceedingJoinPoint
 * @return Object
 * @throws Exception
 */

```

```

public Object logSelect(ProceedingJoinPoint joinPoint) throws Throwable {

    Stopwatch stopWatch = new Stopwatch();

    try {
        stopWatch.start();

        Object retValue = joinPoint.proceed();
        return retValue;
    } catch (Throwable e) {
        throw e;
    } finally {
        stopWatch.stop();

        SysLog sysLog = new SysLog();
        String className = joinPoint.getTarget().getClass().getName();
        String methodName = joinPoint.getSignature().getName();
        String processSeCode = "R";
        String processTime = Long.toString(stopWatch.getTotalTimeMillis());
        String uniqId = "";
        String ip = "";

        /* Authenticated */
        Boolean isAuthenticated = EgovUserDetailsHelper.isAuthenticated();
        if(isAuthenticated.booleanValue()) {
            LoginVO user =
(LoginVO)EgovUserDetailsHelper.getAuthenticatedUser();
            uniqId = user.getUniqId();
            ip = user.getIp();
        }

        sysLog.setSrcvNm(className);
        sysLog.setMethodNm(methodName);
        sysLog.setProcessSeCode(processSeCode);
        sysLog.setProcessTime(processTime);
        sysLog.setRqesterId(uniqId);
        sysLog.setRqesterIp(ip);

        sysLogService.logInsertSysLog(sysLog);

    }

}
}

```

Scheduling

context-scheduling-sym-log-lgm.xml

(src/main/resources/egovframework/spring/com/context-scheduling-sym-log-lgm.xml)

```

<!-- System log summary -->
<bean id="sysLogging"
    class="org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBean">
    <property name="targetObject" ref="egovLogManageScheduling" />
    <property name="targetMethod" value="sysLogSummary" />
    <property name="concurrent" value="false" />
</bean>

<!-- System log summary trigger-->
<bean id="sysLogTrigger"

```



```

class="org.springframework.scheduling.quartz.SimpleTriggerBean">
<property name="jobDetail" ref="sysLogging" />
<!-- Execute in 1 minute after start.(milisecond) -->
<property name="startDelay" value="60000" />
<!--Execute in every 1 hour. (milisecond) -->
<property name="repeatInterval" value="3600000" />
</bean>

<!--Register Scheduler-->
<bean id="logSummaryScheduler"
class="org.springframework.scheduling.quartz.SchedulerFactoryBean">
<property name="triggers">
<list>
<ref bean="sysLogTrigger" />
<ref bean="webLogTrigger" />
<ref bean="trsmrcvLogTrigger" />
<ref bean="userLogTrigger" />
</list>
</property>
</bean>

```

Set [Scheduling](#) to implement function of summary and deletion of system log

- Create EgovLogManageScheduling class to implement the function of summary and deletion of system log

```

@Service("egovSysLogScheduling")
public class EgovSysLogScheduling {

    @Resource(name="EgovSysLogService")
    private EgovSysLogService sysLogService;

    /**
     * Summarize the system log information.
     * Summarize and enter the log of previous day and delete the log for 1 week ago.
     *
     * @param
     * @return
     * @throws Exception
     */
    public void sysLogSummary() throws Exception {
        sysLogService.logInsertSysLogSummary();
    }
}

```

Related Function

Log Management is composed of **log list inquiry** and **log detail inquiry function**.

Log List Inquiry

Business Rule

System log list is inquired by 10 items per page. The paging consists of 10 pages. The condition of search is implemented by the date of generation and processing type. To perform detail inquiry function of system log, click the **Detail Inquiry** button.

Related Codes

N/A

Screen and execution manual

Action	URL	Controller method	QueryID
List inquiry	/sym/log/lgm/SelectSysLogList.do	selectSysLogInf	"SysLogDAO.selectSysLogInf",
			"SysLogDAO.selectSysLogInfCnt"

시스템 로그조회
발생일자 ~
처리구분
조회

번호	요청ID	발생일자	메소드명	처리구분	처리시간	요청자	상세보기
1	SYSLOG_0000000044170	20090428	selectZipList	조회	16	업무담당자	
2	SYSLOG_0000000044171	20090428	selectZipListTotCnt	조회	16	업무담당자	
3	SYSLOG_0000000044172	20090428	selectZipList	조회	47	업무담당자	
4	SYSLOG_0000000044173	20090428	selectZipListTotCnt	조회	16	업무담당자	
5	SYSLOG_0000000044174	20090428	selectZipDetail	조회	31	업무담당자	
6	SYSLOG_0000000044175	20090428	selectZipDetail	조회	94	업무담당자	
7	SYSLOG_0000000044176	20090428	updateZip	수정	47	업무담당자	
8	SYSLOG_0000000044177	20090428	selectZipList	조회	94	업무담당자	
9	SYSLOG_0000000044178	20090428	selectZipListTotCnt	조회	63	업무담당자	
10	SYSLOG_0000000044179	20090428	selectZipDetail	조회	15	업무담당자	

Inquiry: to inquire, select the search conditions on the top, enter the search conditions and click the Inquiry button.

Click List: move to **Log Management Detail Inquiry** screen.

System Log Detail Inquiry

Business Rule

System log detail inquiry is composed of popup window. Click **Close** button to close the window.

Related Codes

N/A

Screen and execution manual

Action	URL	Controller method	QueryID
Detail Inquiry	/sym/log/lgm/InquireSysLog.do	selectSysLog	"SysLogDAO.selectSysLog"

시스템 로그 정보

요청ID	SYSLOG_0000000044170
발생일자	20090428
서비스명	egovframework.com.sym.ccm.zip.service.impl.EgovCcmZipManageServiceImpl
메소드명	selectZipList
처리구분	조회
처리시간	16
요청자	업무담당자
요청자IP	

닫기

Reference

- See Implementation Environment: [AOP](#)
- See Implementation Environment: [Scheduling](#)
- See Implementation Environment: [ID Generation](#)