

OLog 설정 메뉴얼

손 창욱*

Rare Isotope Science Project
Institute for Basic Science, Daejeon, South Korea
July 7, 2014

Abstract

OLog는 REST 기반 Electronic logbook으로 JAVA 와 Python API 및 Web Client 툴을 제공하고 있다. 본 메뉴얼에서는 OLog Service의 설치 및 Web Client 와 CSS를 이용한 Log작성 방법에 대하여 설명 하였으며, OLog에 대한 자세한 내용 및 자료는 홈페이지를[1] 참고하기 바란다.

1 설치

OLog Service를 사용하기 위해서는 기본적으로 3가지 서버(MySQL, LDAP, Glassfish)가 설치 되어 있어야 한다. 따라서 여기서는 3가지 기본 서버가 설치 되어있다는 가정하에 OLog를 위한 설정 과정을 설명한다. 본 메뉴얼에서 사용된 하드웨어 및 소프트웨어 사양은 아래와 같다.

- MySQL Server 5.5
- LDAP Server (OpenLDAP 2.4.31)
- Glassfish Server 3.1.2.2
- Debian Linux 7 64bit wheezy (kernel 3.2.0)
- OLog Web Service 2.2.6[2]
- Web Client v0.4-beta[2]
- CSS NSLS-II 3.2.16a[3]

모든 서버는 하나의 Server PC에 설치되어 있으며, IP주소는 10.1.5.14로 설정되어 있다.

*scwook@ibs.re.kr

1.1 MySQL 설정

OLog의 DB구조는 Figure 1과 같으며, 각 테이블에는 저장되는 정보는 다음과 같다.

- attributes:
- entries: 로그 생성날짜
- logbooks: 로그북 및 태그 리스트
- logs: 로그 데이터
- logs_attributes:
- logs_logbooks: 로그데이터에 대한 로그북 및 태그 정보
- properties:
- schema_version:
- subscriptions:

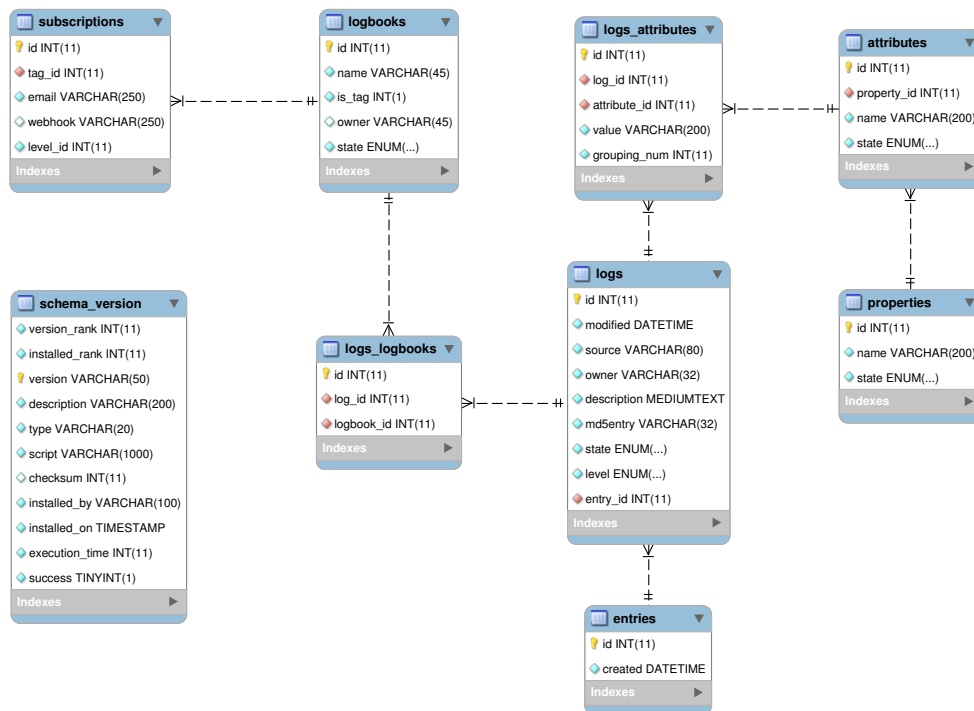


Figure 1 OLOG Database Schema

DB 및 테이블 구조는 MySQL 서버에서 직접 만들 수 있지만 script(olog_schema.sql) 파일을 하나 만들어 실행시키는 것이 편리하다. script 내용은 Appendix A를 참고하길 바라며, 만약 아래 테이블 구조와 다르다면 Web Client를 이용하여 OLog에 접속 후 다시 확인해 보기 바란다.

```
ctrluser@ctrluser:~\$ mysql -u root -p
Enter password:
mysql> source olog_schema.sql;
...
Query OK, 0 rows affected (0.00sec)
mysql> use olog;
mysql> show tables;
+-----+
| Tables_in_olog |
+-----+
| attributes      |
| entries         |
| logbooks        |
| logs            |
| logs_attributes |
| logs_logbooks   |
| properties      |
| schema_version  |
| subscriptions   |
+-----+
9 rows in set (0.01 sec)

mysql>
```

1.2 LDAP 설정

OLog Server는 다음 4개의 사용자 그룹으로 분류되어 있다.

- olog_admins: 관리자 사용자
- olog_logbooks:
- olog_logs: 일반 사용자
- olog_tags:

그룹 및 사용자는 각각 Group과 People을 상위 엔트리로 가지므로 아래와 같이 2개의 엔트리를 가지는 LDIF파일을 생성한 후 LDAP Server에 추가한다.

```
ctrluser@ctrluser:~\$ vi ou.ldif

dn: ou=Group,dc=risp,dc=net
objectClass: organizationalunit
ou: Group
description: groups branch

dn: ou=People,dc=risp,dc=net
objectClass: organizationalunit
ou: People
description: people branch
```

```
ctrluser@ctrluser:~\$ sudo invoke-rc.d slapd stop
ctrluser@ctrluser:~\$ sudo slapadd -c -v -l ou.ldif
added: "ou=People,dc=spinlock,dc=hr" (000000003)
added: "ou=Group,dc=spinlock,dc=hr" (000000004)
\_##### 100.00\% eta   none elapsed           none fast!
Closing DB...

ctrluser@ctrluser:~\$ sudo invoke-rc.d slapd start
```

다음에는 그룹과 사용자를 각각 추가하도록 한다. 사용자는 관리자 권한을 가지는 ctrluser와 일반 사용자 권한을 가지는 scwook을 추가한다.

```
ctrluser@ctrluser:~\$ vi people.ldif

dn: uid=ctrluser,ou=People,dc=risp,dc=net
uid: ctrluser
objectClass: account
objectClass: posixAccount
description: User with admin role
cn: ctrluser
uidNumber: 23001
gidNumber: 23001
homeDirectory: /dev/null

dn: uid=scwook,ou=People,dc=risp,dc=net
uid: scwook
objectClass: account
objectClass: posixAccount
description: User with user role
cn: scwook
uidNumber: 23001
gidNumber: 23001
homeDirectory: /dev/null

ctrluser@ctrluser:~\$ sudo invoke-rc.d slapd stop
ctrluser@ctrluser:~\$ sudo slapadd -c -v -l people.ldif

ctrluser@ctrluser:~\$ vi group.ldif

dn: cn=olog-logbooks,ou=Group,dc=risp,dc=net
cn: olog-logbooks
objectClass: posixGroup
description: olog-logbooks group
gidNumber: 24001

dn: cn=olog-tags,ou=Group,dc=risp,dc=net
cn: olog-tags
objectClass: posixGroup
description: olog-tags group
gidNumber: 24002

dn: cn=olog-logs,ou=Group,dc=risp,dc=net
cn: olog-logs
objectClass: posixGroup
description: olog-logs group
gidNumber: 24003
memberUid: ctrluser
memberUid: scwook

dn: cn=olog-admins,ou=Group,dc=risp,dc=net
cn: olog-admins
```

```
objectClass: posixGroup
description: olog-admins group
gidNumber: 24004
memberUid: ctrluser

ctrluser@ctrluser:~\$ sudo slapadd -c -v -l group.ldif
ctrluser@ctrluser:~\$ sudo invoke-rc.d slapd start
```

추가된 그룹과 사용자는 ldapsearch 명령을 통해 확인 할 수 있다.

```
ctrluser@ctrluser:~\$ ldapsearch -x cn=olog-logs
# extended LDIF
#
# LDAPv3
# base <dc=risp,dc=net> (default) with scope subtree
# filter: cn=olog-logs
# requesting: ALL
#
# olog-logs, Group, risp.net
dn: cn=olog-logs,ou=Group,dc=risp,dc=net
cn: olog-logs
objectClass: posixGroup
description: olog-logs group
gidNumber: 24003
memberUid: ctrluser
memberUid: scwook

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1

ctrluser@ctrluser:~\$
```

패스워드는 ldappasswd 명령을 통해 설정 한다.

```
ctrluser@ctrluser:~\$ ldappasswd -x -D cn=admin,dc=risp,dc=ne\citep{OLOG_DN}t -W -S uid=scwook,ou=
people,dc=risp,dc=net
New password:
Re-enter new password:
Enter LDAP Password:
ctrluser@ctrluser:~\$
```

1.3 Glassfish 설정

설치된 Glassfish 서버에 관리자 계정으로 로그인 한다.

JDBC

Resources - JDBC - Connection Pools에서 새로운 Connection Pool 생성 후 아래와 같이 설정 한다.

- Pool Name: OlogPool
- Resource Type: javax.sql.ConnectionPoolDataSource

- Datasource Classname: com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource
- 다음 속성을 추가한다(Name=Value)
 - Server Name=localhost
 - Database Name=olog
 - User=root(MySQL 서버의 유저 아이디)
 - Password=1234(MySQL 유저 패스워드)

Resources - JDBC - JDBC Resources에서 새로운 JDBC Resources를 생성 후 아래와 같이 설정한다.

- JNDI Name: jdbc/olog
- Pool Name: OlogPool

Realm

Configurations - server-config - Security - Realms에서 새로운 Realms을 생성 후 아래와 같이 설정한다.

- Name: olog
- Class Name: com.sun.enterprise.security.auth.realm.ldap.LDAPRealm
- JASS Context: ldapRealm
- Directory: ldap://10.1.5.14:389
- Base DN: dc=risp,dc=net
- 다음 속성을 추가한다(Name=Value)
 - group-search-filter=memberUid=%s

Resources - JNDI - Custom Resources에서 새로운 Resource를 생성 후 아래와 같이 설정한다.

- JNDI Name: ologGroups
- Resource Type: javax.naming.directory.Directory
- Factory Class: com.sun.jndi.LdapCtxFactory
- 다음 속성을 추가한다(Name=Value)
 - javax.naming.security.principal=cn=admin,dc=risp,dc=net
 - URL=ldap://10.1.5.14/dc=risp,dc=net

Depoly

Applications에서 새로운 Deploy를 생성 후 아래와 같이 설정한다.

- Type: Web Application
- Context Root: Olog

2 Client 도구

OLog는 REST 기반 프로토콜을 이용하여 사용할 수 있는데 개발자를 위해 JAVA 및 Python API를 제공하고 있다.[2] 여기서는 기본적으로 제공하는 Web Client와 EPICS Extension Tool인 CSS를 이용한 로그 작성법에 대하여 설명 하였다.

2.1 Web Client

OLog 서버 주소를 설정하기 위해 Web Client폴더 안에 있는 "configuration.js" 파일을 열어 serviceurl 주소를 다음과 같이 지정해 준다.

```
ctrluser@ctrluser:~\$ vi logbook-0.4-beta\Olog\public_html\static\js\configuration.js
```

```
// For accessing the REST service  
var serviceurl = "https://10.1.5.14:8181/Olog/resources/";
```

주소가 보안 프로토콜인 "https://"로 시작하는 것에 주의한다. 만약 "http://"를 사용하면 서버 연결 오류가 나타난다. 보안 프로토콜의 경우 웹 브라우저 상에서 인증서를 받거나 예외 항목에 추가해야 한다. 간단히 OLog service 주소("https://10.1.5.14:8181/Olog/resources/tags")에 접속하면 보안 경고 창이 나타나는데 여기서 예외 항목에 추가하면 된다. 접속은 Web Browser를 이용하여 Olog 폴더안에 있는 index.html 파일을 실행하면 된다.

```
ctrluser@ctrluser:~\$ firefox logbook-0.4-beta\Olog\index.html
```

로그 작성을 위해 오른쪽 상단에 있는 User버튼을 눌러 로그인을 한다. 로그인이 완료되면 왼쪽 상단에 있는 New Log Entry 버튼이 활성화 되며 이 버튼을 클릭하여 로그를 작성한다.

2.2 CSS Client

CSS에서 OLog를 사용하기 위해 Edit-Preferences 메뉴에서 아래와 같이 설정한다.

- Olog Service URL: https://10.1.5.14:8181/Olog/resources
- use authentication: 체크 해제
- Prompt User for authentication with each log entry creation: 체크
- Default logbook: 기본으로 사용할 로그북
- Page Size: 50

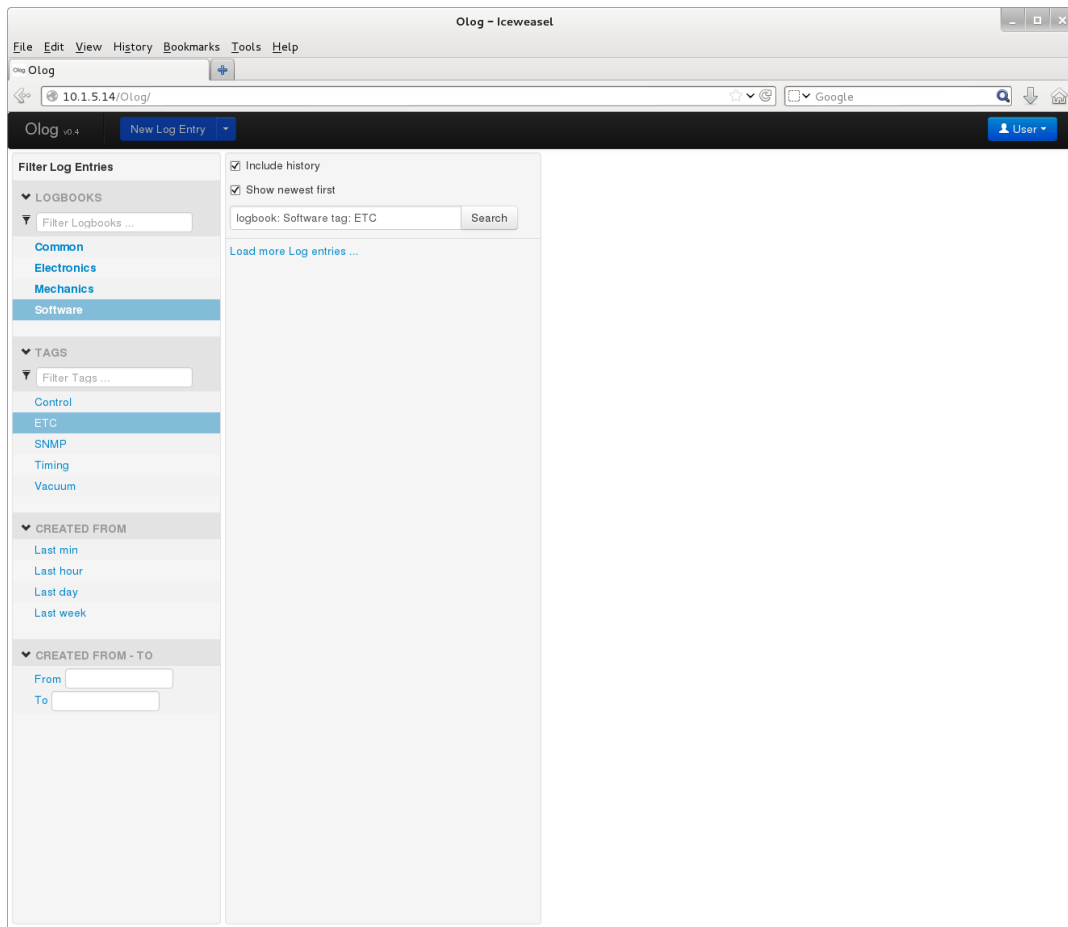


Figure 2 Web Client v0.4 beta

설정을 마친 후 CSS 재시작 한다.

로그 작성을 위해 CSS-Utilities-Create Log Entry 메뉴를 선택 한다. User Name과 Password는 OLog 사용자와 비밀번호를 입력한다. 작성된 로그는 Log Table에서 확인할 수 있는데 Log Table은 Window-Show View-Other메뉴에서 Other-Log Table을 선택 하면 볼 수 있다. Log Table의 Log Query값에 logbook:Common값을 입력하면 작성된 로그를 확인할 수 있으며, 더블 클릭하면 자세한 내용을 볼 수 있다.

The image shows a 'Create Log Entry' dialog box. The title bar reads 'Create Log Entry (on ctrlldb4)'. The form includes the following elements:

- User Name:** A text field containing 'scwook'.
- Password:** A text field with masked characters '....'.
- Date:** A date picker showing 'Jul 2, 2014'.
- Level:** A dropdown menu currently set to 'Info'.
- Log Entry Test:** A large text area for entering the log message.
- Logbooks:** A dropdown menu set to 'Common'.
- Tags:** A dropdown menu set to 'ETC'.
- Hide details:** A button located below the dropdowns.
- Images | Files | Properties | Ticket | Shift:** A set of tabs at the bottom of the main content area.
- Buttons:** 'Cancel' and 'Submit' buttons at the bottom right of the dialog.

Figure 3 Create Log Entry

Bibliography

- [1] OLog Home Page, 2014.
<http://olog.github.io/2.2.7-SNAPSHOT/modules/olog-service/index.html>
 (accessed July 01, 2014).
- [2] OLog Download Page, 2014.
<http://olog.github.io/2.2.7-SNAPSHOT/> (accessed July 01, 2014).
- [3] Contol System Studio Home Page, 2014.
<http://controlsystemstudio.org/> (accessed July 01, 2014).

Appendix A

다음 script는 MySQL 데이터 베이스에 OLog DB schema를 생성하는 script이다. 실제 Table구조(Figure1)는 아래 script와 다른 구조로 생성되는데, Web Client를 통해 접속 하게 되면 최신 DB구조로 변경된다. 예를 들어 levels와 statuses은 삭제되고 entries와 schema_version이 생성된다.

```
1  /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
2  /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
3  /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
4  /*!40101 SET NAMES utf8 */;
5
6  /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
7  /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
8  /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
9
10 -- Create schema ologdb4
11 DROP DATABASE IF EXISTS olog;
12 CREATE DATABASE olog;
13 USE olog;
14
15 -- Definition of table 'levels'
16
17 DROP TABLE IF EXISTS 'levels';
18 CREATE TABLE 'levels' (
19   'id' int(11) NOT NULL AUTO_INCREMENT,
20   'name' varchar(200) NOT NULL,
21   PRIMARY KEY ('id'),
22   UNIQUE KEY 'name' ('name')
23 ) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
24
25 -- Dumping data for table 'levels'
26
27 /*!40000 ALTER TABLE 'levels' DISABLE KEYS */;
28 INSERT INTO 'levels' ('id','name') VALUES
29   (1,'Info'),
30   (4,'Problem'),
31   (3,'Request'),
32   (2,'Suggestion'),
33   (5,'Urgent');
34 /*!40000 ALTER TABLE 'levels' ENABLE KEYS */;
35
36 -- Definition of table 'logbooks'
37
38 DROP TABLE IF EXISTS 'logbooks';
39 CREATE TABLE 'logbooks' (
40   'id' int(11) NOT NULL AUTO_INCREMENT,
41   'name' varchar(45) NOT NULL,
42   'is_tag' int(1) NOT NULL DEFAULT '0',
43   'owner' varchar(45) DEFAULT NULL,
44   'status_id' int(11) NOT NULL DEFAULT '1',
45   PRIMARY KEY ('id'),
46   UNIQUE KEY 'name' ('name'),
47   KEY 'logbooks_status_id_fk' ('status_id'),
48   CONSTRAINT 'logbooks_status_id_fk' FOREIGN KEY ('status_id') REFERENCES 'statuses' ('id') ON
49   DELETE NO ACTION ON UPDATE NO ACTION
50 ) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=latin1;
51
52 -- Dumping data for table 'logbooks'
```

```

53  /*!40000 ALTER TABLE 'logbooks' DISABLE KEYS */;
54  INSERT INTO 'logbooks' ('id','name','is_tag','owner','status_id') VALUES
55    (1,'Operations',0,NULL,1),
56    (2,'Electronics Maintenance',0,NULL,1),
57    (3,'Mechanical Technicians',0,NULL,1),
58    (4,'LOTO',0,NULL,1),
59    (5,'Inverpower Power Supplies',1,NULL,1),
60    (6,'RF Area',1,NULL,1),
61    (7,'Kicker',1,NULL,1),
62    (8,'Bumps',1,NULL,1),
63    (9,'Septums',1,NULL,1),
64    (10,'Large Power Supplies',1,NULL,1),
65    (11,'Timing Systems',1,NULL,1);
66  /*!40000 ALTER TABLE 'logbooks' ENABLE KEYS */;
67
68  -- Definition of table 'logs'
69
70  DROP TABLE IF EXISTS 'logs';
71  CREATE TABLE 'logs' (
72    'id' int(11) NOT NULL AUTO_INCREMENT,
73    'created' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
74    'source' varchar(80) NOT NULL DEFAULT '',
75    'owner' varchar(32) NOT NULL,
76    'level_id' int(11) NOT NULL DEFAULT '1',
77    'status_id' int(11) NOT NULL DEFAULT '1',
78    'description' mediumtext NOT NULL,
79    'md5entry' varchar(32) NOT NULL DEFAULT '',
80    'md5recent' mediumtext NOT NULL,
81    'parent_id' int(11) DEFAULT NULL,
82    PRIMARY KEY ('id'),
83    KEY 'level_id_fk' ('level_id'),
84    KEY 'log_parent_id_fk' ('parent_id'),
85    KEY 'status_id_fk' ('status_id'),
86    CONSTRAINT 'level_id_fk' FOREIGN KEY ('level_id') REFERENCES 'levels' ('id') ON DELETE NO ACTION
      ON UPDATE NO ACTION,
87    CONSTRAINT 'log_parent_id_fk' FOREIGN KEY ('parent_id') REFERENCES 'logs' ('id') ON DELETE NO
      ACTION ON UPDATE NO ACTION,
88    CONSTRAINT 'status_id_fk' FOREIGN KEY ('status_id') REFERENCES 'statuses' ('id') ON DELETE NO
      ACTION ON UPDATE NO ACTION
89  ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
90
91  -- Definition of table 'logs_logbooks'
92
93  DROP TABLE IF EXISTS 'logs_logbooks';
94  CREATE TABLE 'logs_logbooks' (
95    'id' int(11) NOT NULL AUTO_INCREMENT,
96    'log_id' int(11) NOT NULL,
97    'logbook_id' int(11) NOT NULL,
98    'state' enum('open','closed') DEFAULT NULL,
99    'created' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
100    'status_id' int(11) NOT NULL DEFAULT '1',
101    PRIMARY KEY ('id'),
102    KEY 'log_id_fk' ('log_id'),
103    KEY 'logbook_id_fk' ('logbook_id') USING BTREE,
104    KEY 'logs_logbooks_status_id_fk' ('status_id'),
105    CONSTRAINT 'logs_logbooks_status_id_fk' FOREIGN KEY ('status_id') REFERENCES 'statuses' ('id')
      ON DELETE NO ACTION ON UPDATE NO ACTION,
106    CONSTRAINT 'logs_logbooks_logbook_id_fk' FOREIGN KEY ('logbook_id') REFERENCES 'logbooks' ('id')
      ON DELETE NO ACTION ON UPDATE NO ACTION,
107    CONSTRAINT 'logs_logbooks_log_id_fk' FOREIGN KEY ('log_id') REFERENCES 'logs' ('id') ON DELETE
      NO ACTION ON UPDATE NO ACTION
108  ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;

```

```

109
110 -- Definition of table 'properties'
111
112 DROP TABLE IF EXISTS 'properties';
113 CREATE TABLE 'properties' (
114     'id' int(11) NOT NULL AUTO_INCREMENT,
115     'name' varchar(200) NOT NULL,
116     'status_id' tinyint(1) NOT NULL,
117     PRIMARY KEY ('id')
118 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
119
120 -- Dumping data for table 'properties'
121
122 /*!40000 ALTER TABLE 'properties' DISABLE KEYS */;
123 /*!40000 ALTER TABLE 'properties' ENABLE KEYS */;
124
125 -- Definition of table 'attributes'
126
127 DROP TABLE IF EXISTS 'attributes';
128 CREATE TABLE 'attributes' (
129     'id' int(11) NOT NULL AUTO_INCREMENT,
130     'property_id' int(11) NOT NULL,
131     'name' varchar(200) NOT NULL,
132     'status_id' tinyint(1) NOT NULL,
133     PRIMARY KEY ('id'),
134     KEY 'attributes_property_id_fk' ('property_id'),
135     CONSTRAINT 'attributes_property_id_fk' FOREIGN KEY ('property_id') REFERENCES 'properties' ('id'
136     ) ON DELETE NO ACTION ON UPDATE NO ACTION
137 ) ENGINE=InnoDB AUTO_INCREMENT=19 DEFAULT CHARSET=latin1;
138
139 -- Definition of table 'logs_attributes'
140
141 DROP TABLE IF EXISTS 'logs_attributes';
142 CREATE TABLE 'logs_attributes' (
143     'id' int(11) NOT NULL AUTO_INCREMENT,
144     'log_id' int(11) NOT NULL,
145     'attribute_id' int(11) NOT NULL,
146     'value' varchar(200) NOT NULL,
147     'grouping_num' int(11) NOT NULL,
148     PRIMARY KEY ('id'),
149     KEY 'logs_attributes_attribute_id_fk' ('attribute_id'),
150     KEY 'logs_attributes_log_id_fk' ('log_id'),
151     CONSTRAINT 'logs_attributes_attribute_id_fk' FOREIGN KEY ('attribute_id') REFERENCES 'attributes'
152     ('id') ON DELETE NO ACTION ON UPDATE NO ACTION,
153     CONSTRAINT 'logs_attributes_log_id_fk' FOREIGN KEY ('log_id') REFERENCES 'logs' ('id') ON DELETE
154     NO ACTION ON UPDATE NO ACTION
155 ) ENGINE=InnoDB AUTO_INCREMENT=179 DEFAULT CHARSET=latin1;
156
157 -- Definition of table 'statuses'
158
159 DROP TABLE IF EXISTS 'statuses';
160 CREATE TABLE 'statuses' (
161     'id' int(11) NOT NULL AUTO_INCREMENT,
162     'name' varchar(45) NOT NULL,
163     PRIMARY KEY ('id')
164 ) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
165
166 -- Dumping data for table 'statuses'
167
168 /*!40000 ALTER TABLE 'statuses' DISABLE KEYS */;
169 INSERT INTO 'statuses' ('id','name') VALUES
170 (1,'Active'),

```

```

168     (2,'Inactive');
169 /*!40000 ALTER TABLE 'statuses' ENABLE KEYS */;
170
171 -- Definition of table 'subscriptions'
172
173 DROP TABLE IF EXISTS 'subscriptions';
174 CREATE TABLE 'subscriptions' (
175     'id' int(11) NOT NULL AUTO_INCREMENT,
176     'tag_id' int(11) NOT NULL,
177     'email' varchar(250) NOT NULL DEFAULT '',
178     'webhook' varchar(250) DEFAULT NULL,
179     'level_id' int(11) NOT NULL,
180     PRIMARY KEY ('id'),
181     KEY 'subscriptions_tag_id_fk' ('tag_id'),
182     CONSTRAINT 'subscriptions_tag_id_fk' FOREIGN KEY ('tag_id') REFERENCES 'logbooks' ('id') ON
        DELETE NO ACTION ON UPDATE NO ACTION
183 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
184
185 -- Dumping data for table 'subscriptions'
186
187 /*!40000 ALTER TABLE 'subscriptions' DISABLE KEYS */;
188 /*!40000 ALTER TABLE 'subscriptions' ENABLE KEYS */;
189
190 /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
191 /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
192 /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
193 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
194 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
195 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
196 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

```

Listing A.1 olog_schema.sql

Appendix B

LDAP 사용이 어려운 경우 다음과 같이 File Realm을 사용할 수 있다
Configurations - server-config - Security - Realms에서 새로운 Realms을 생성 후 아래와 같이 설정한다.

- Name: olog
- Class Name: com.sun.enterprise.security.auth.realm.file.FileRealm
- JAAS Context: fileRealm
- Key File: \${com.sun.aas.instanceRoot}/config/olog_keyfile

저장 후 방금 생성한 olog를 선택하면 왼쪽 상단에 있는 Manage Users를 클릭하여 사용자를 추가한다. Group List는 관리자 권한의 경우 olog-admins를 일반 사용자의 경우 olog-logs를 입력하면 된다.