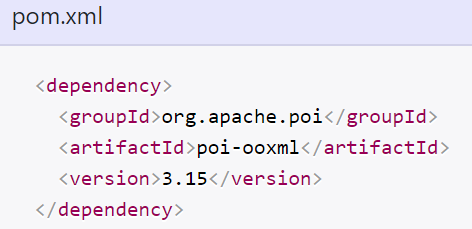
Maven에서 POI 엑셀 다운로드 하는법

<https://www.mkyong.com/java/apache-poi-reading-and-writing-excel-file-in-java/>

<https://yonjun.tistory.com/3>

1. Pom.xml에 dependency 추가



1. Controller.java에 추가

package com.mkyong;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

public class ApachePOIExcelWrite {

private static final String FILE\_NAME = "/tmp/MyFirstExcel.xlsx";

public static void main(String[] args) {

XSSFWorkbook workbook = new XSSFWorkbook();

XSSFSheet sheet = workbook.createSheet("Datatypes in Java");

Object[][] datatypes = {

{"Datatype", "Type", "Size(in bytes)"},

{"int", "Primitive", 2},

{"float", "Primitive", 4},

{"double", "Primitive", 8},

{"char", "Primitive", 1},

{"String", "Non-Primitive", "No fixed size"}

};

int rowNum = 0;

System.out.println("Creating excel");

for (Object[] datatype : datatypes) {

Row row = sheet.createRow(rowNum++);

int colNum = 0;

for (Object field : datatype) {

Cell cell = row.createCell(colNum++);

if (field instanceof String) {

cell.setCellValue((String) field);

} else if (field instanceof Integer) {

cell.setCellValue((Integer) field);

}

}

}

try {

FileOutputStream outputStream = new FileOutputStream(FILE\_NAME);

workbook.write(outputStream);

workbook.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

System.out.println("Done");

}

}

JSP Form

<div id=*"download"* style="display: *none*; overflow: *visible*">

<div>

<spring:message code=*"TITLE.COM.0001"* />

</div>

<div class=*"ranges-group"*>

<form name=*"formSearchRulesetImport"* id=*"formSearchRulesetImport"*>

<table class=*"table-group"*>

<tr>

<th scope=*"row"*>

<spring:message code=*"FLD.LSTA.0008"* />

</th>

<td>

<div class=*"ranges-group"*>

<div class=*"range-8"*>

<div class=*"form-select-outer"*>

<select name=*"download\_stats\_func"* class=*"form-select"* data-scroll=*"false"*>

<option value=*"csv"*><spring:message code=*"SEL.COM.0060"* /></option>

<option value=*"excel"*><spring:message code=*"SEL.COM.0061"* /></option>

</select>

</div>

</div>

</div>

</td>

</tr>

</table>

<div class=*"table-bottom align-center"*>

<button type=*"button"* class=*"btn-basic btn-download"*>

<spring:message code=*"BTN.COM.0015"* />

</button>

<button type=*"button"* class=*"btn-basic btn-cancel"*>

<spring:message code=*"BTN.COM.0014"* />

</button>

</div>

</form>

</div>

</div>

Js 파일

mCfg = {

DOM:{

btnDownloadPlus : "#assetbtn .btn-download-plus",

//dlg : "#windowProgress",

selDownload : "#download",

btnDownload : "btn-download",

btnCancel : "btn-cancel",

downloadStatsFunc : "[name=download\_stats\_func]",

msg : ".progress-msg",

rateTxt : ".progress-rate-txt",

rateBox : ".progress-rate-box",

hiddenFrame : "hiddenIFrame"

},

urlList : gCONTEXT\_PATH + 'system/asset\_list.json',

urlForm : gCONTEXT\_PATH + 'system/asset\_form.html',

urlDownload : gCONTEXT\_PATH + 'system/asset\_list\_export.do',

urlExcelDownload : gCONTEXT\_PATH + 'system/asset\_list\_excel\_export.do',

urlImport : gCONTEXT\_PATH + 'system/asset\_import\_form.html',

urlEqpJson : gCONTEXT\_PATH + "system/eqp\_codes.json",

formId : '#searchAssetList',

gridId : '#gridAssetList'

},

m$ = {

//dlg : $(mCfg.DOM.dlg),

selDownload : $(mCfg.DOM.selDownload),

btnDownload : $(mCfg.DOM.btnDownload),

downloadStatsFunc : $(mCfg.DOM.downloadStatsFunc),

form : $(mCfg.formId),

grid : $(mCfg.gridId)

}

//다이얼로그 창 처럼 여는 법

m$.selDownload.jqxWindow({

width: 400, height:140, autoOpen: **false**,

resizable: **false**, isModal: **true**, modalOpacity: 0.5,

cancelButton : m$.selDownload.find('.' + mCfg.DOM.btnCancel),

initContent : **function**(){

slui.attach.setTransformSelect(mCfg.DOM.selDownload);

}

});

//위의 다이얼로그 창을 열기 위해서 binding

$(mCfg.DOM.btnDownloadPlus).on("click", showExportDlg);

//실제 다이얼 로그 창을 여는 함수

showExportDlg = **function**() {

m$.selDownload.jqxWindow("open");

m$.selDownload.off().on("click", "." + mCfg.DOM.btnDownload, downloadExcel);

},

downloadExcel = **function**(){

**if**(!\_SL.validate(m$.form))

**return**;

// Export 실행

**if**(m$.downloadStatsFunc.val() == 'csv'){

console.log("csv")

m$.form.attr({

action : mCfg.urlDownload

}).submit();

//refMng.submitToFrame(mCfg.URL.logExport, param);

}

**else**{

console.log("Excel")

m$.form.attr({

action : mCfg.urlExcelDownload

}).submit();

}

closeDlg();

};

// 창을 열고 닫는 함수

closeDlg = **function**() {

//m$.dlg.jqxWindow("close");

m$.selDownload.jqxWindow("close");

},

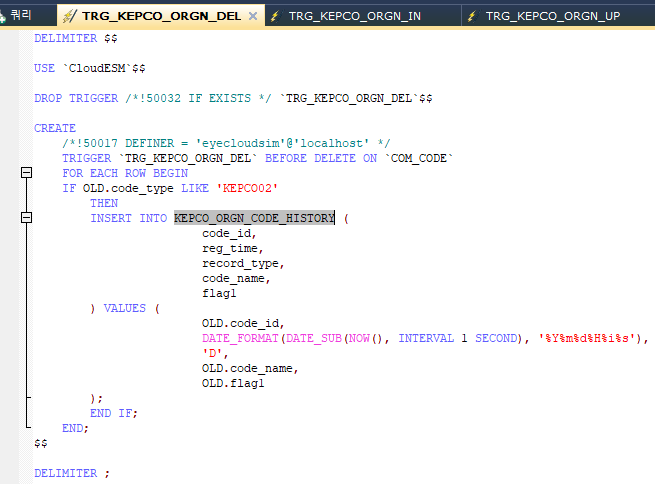
PIMS 버전

- 진행 : mysql의 트리거를 이용한 데이터 활용

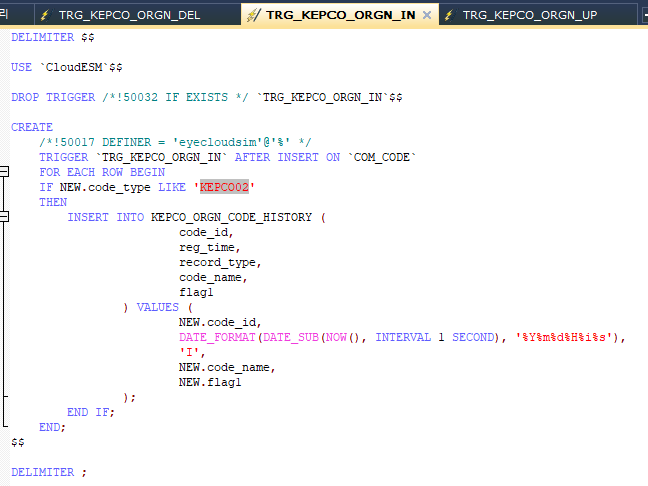
위치: 연동관리 > 외부시스템관리 > 조직도관리 > 변경이력관리 tap

Mysql 트리거 설정.

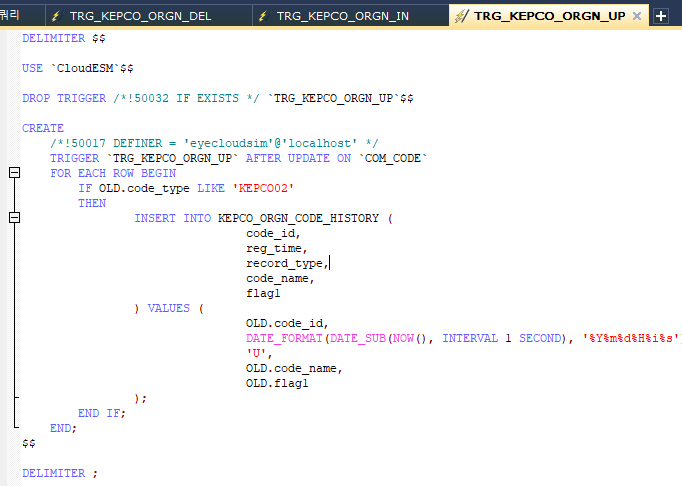
Delete 트리거



Insert 트리거



Update 트리거

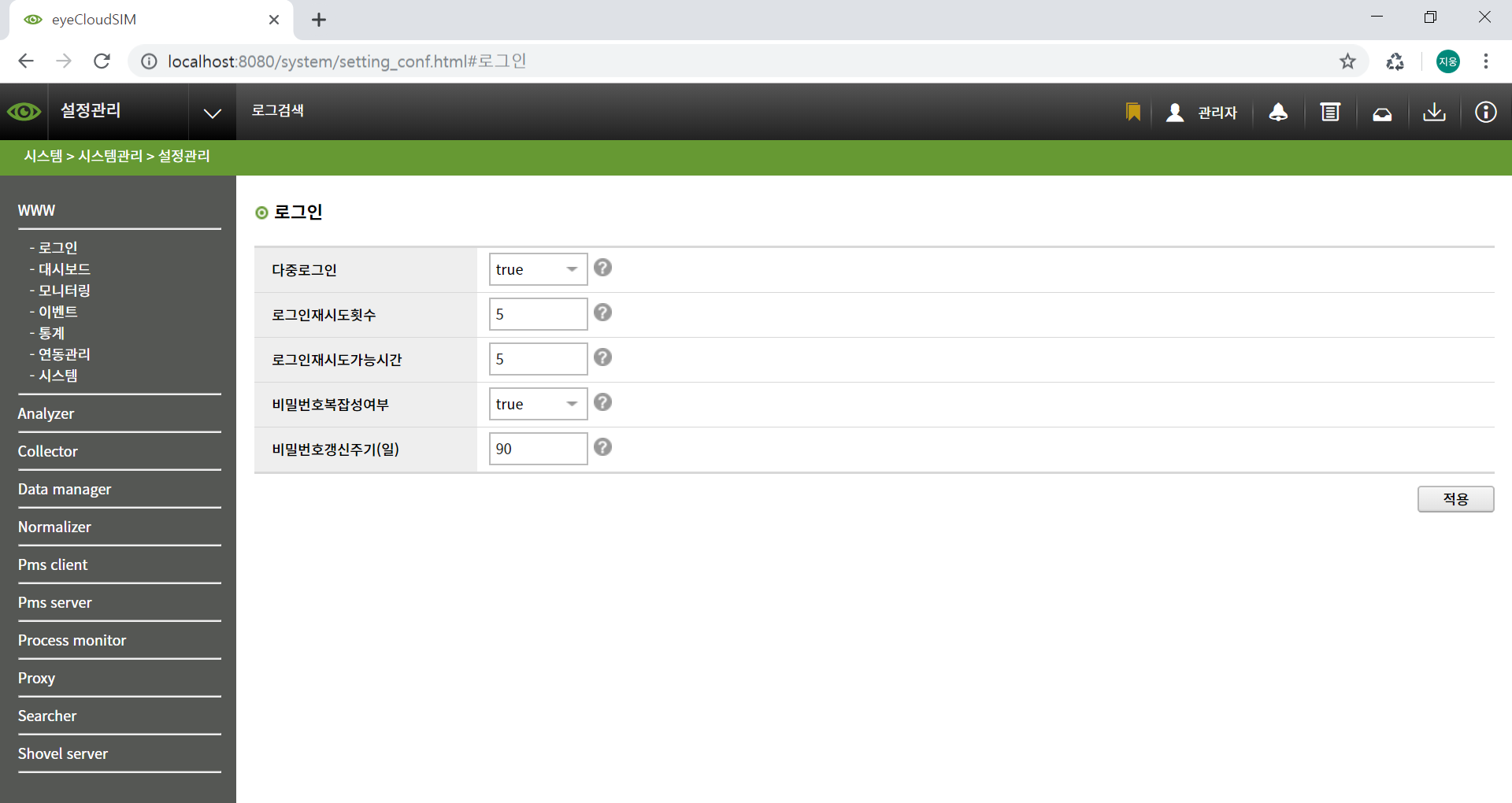


\* 주요사항 - Values에 들어가는 OLD.\* or NEW.\* 은 트리거가 바라보는 테이블의 컬럼이여야 한다.

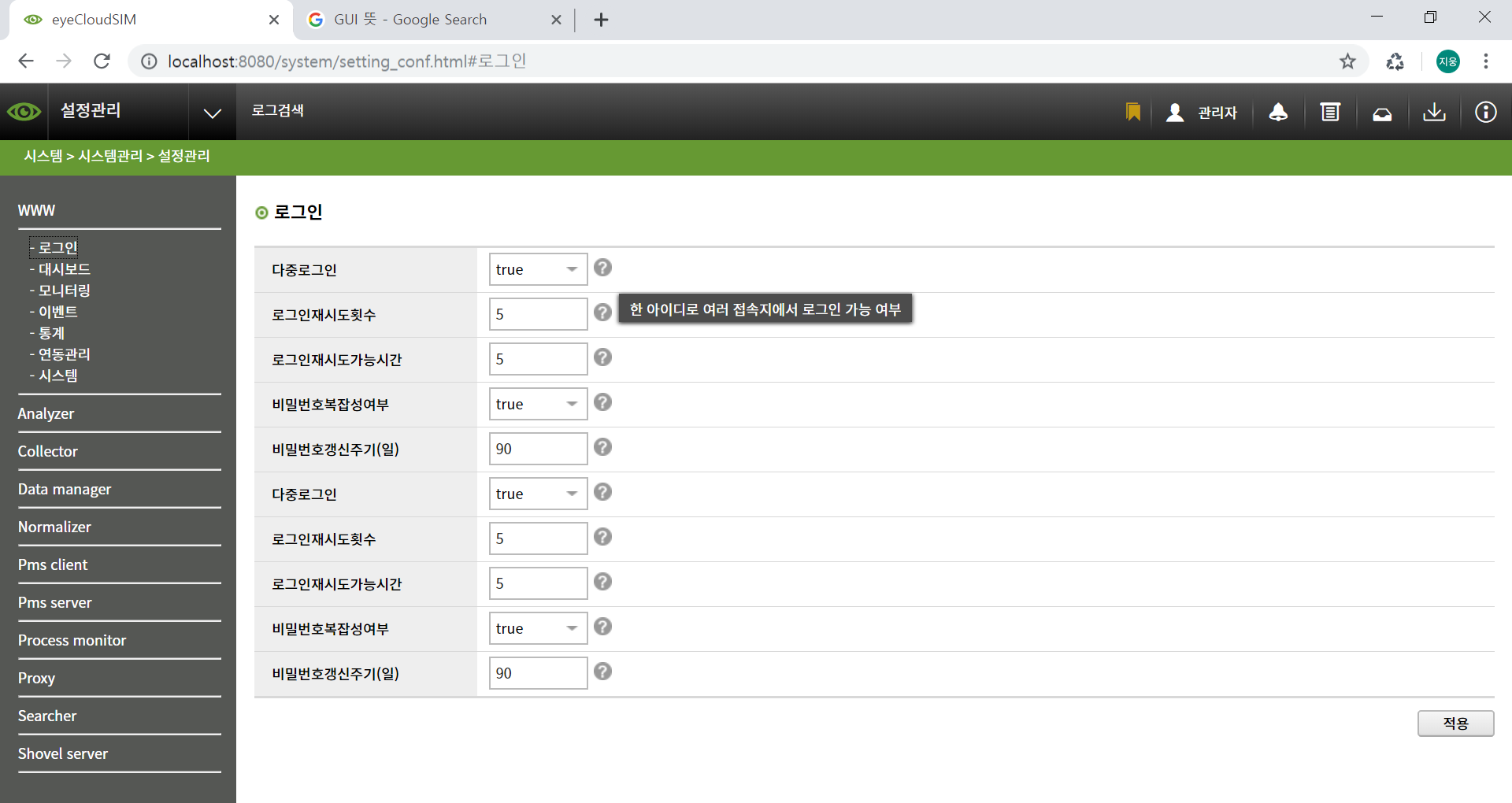
설명 :

1. TRG\_KEPCO\_GRGN\_UP이라는 트리거를 만드는데 이 트리거는 COM\_CODE라는 테이블을 바라보고 있다.
2. For each row begin을 통해서 for문처럼 행동한다. 즉 하나씩 확인을 한다.
3. If문을 통해서 COM\_CODE에 code\_type LIKE ‘KEPCO002’라는것만 INSERT, DELETE, UPDATE를 TALBE KEPCO\_ORGN\_CODE\_HISTORY에 한다.

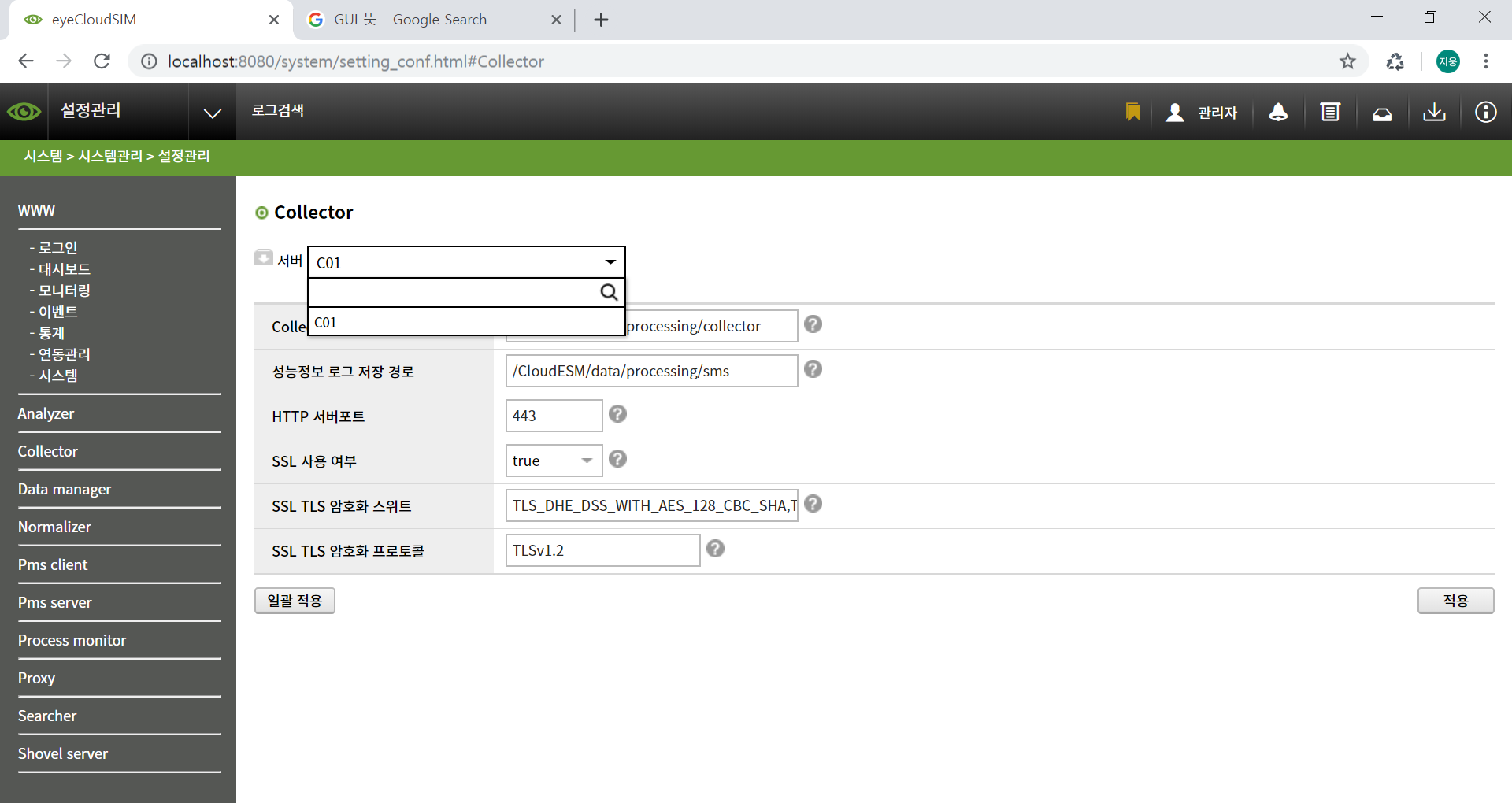
설정관리 GUI 개발



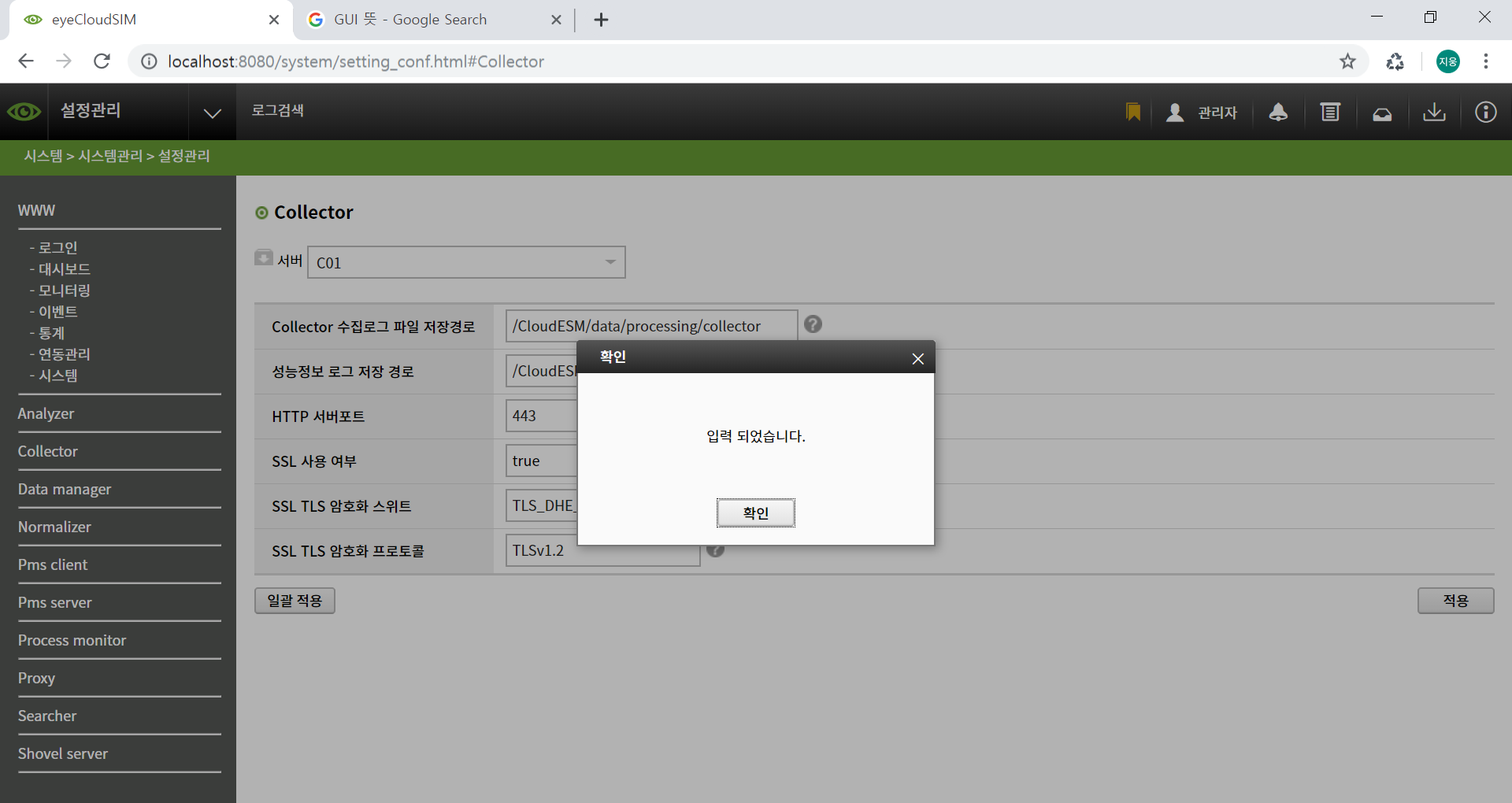
* 기본 메인 화면



* 물음표 마우스 툴팁 생성



* 서버 별 검색 가능한 Select box

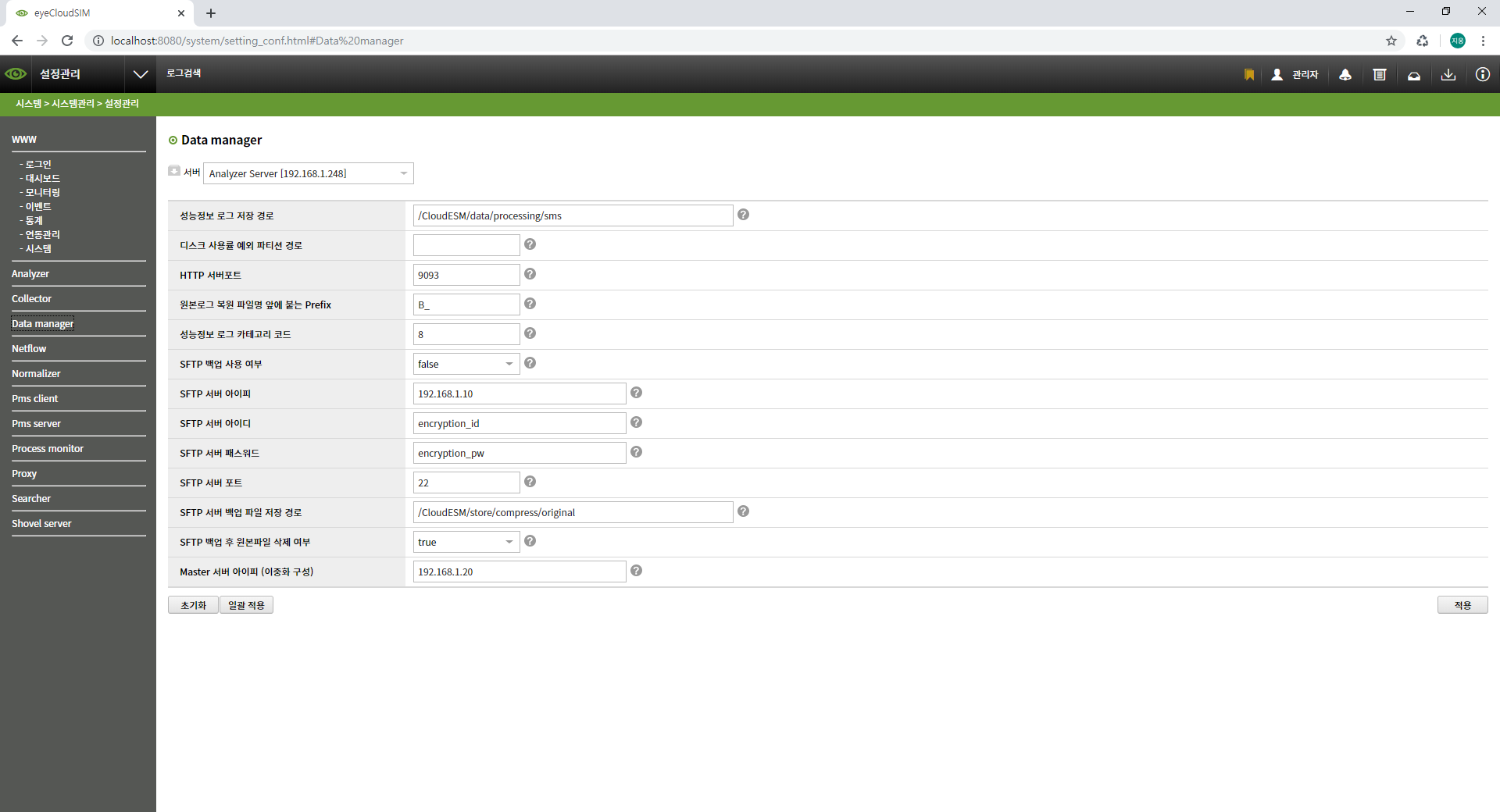


* 적용 버튼 클릭시 DB에 정보를 저장

파일 : Java – SettingConfController.java, SettingConfDAO.java, SettingConfMapper.xml

Js – setting\_conf.js

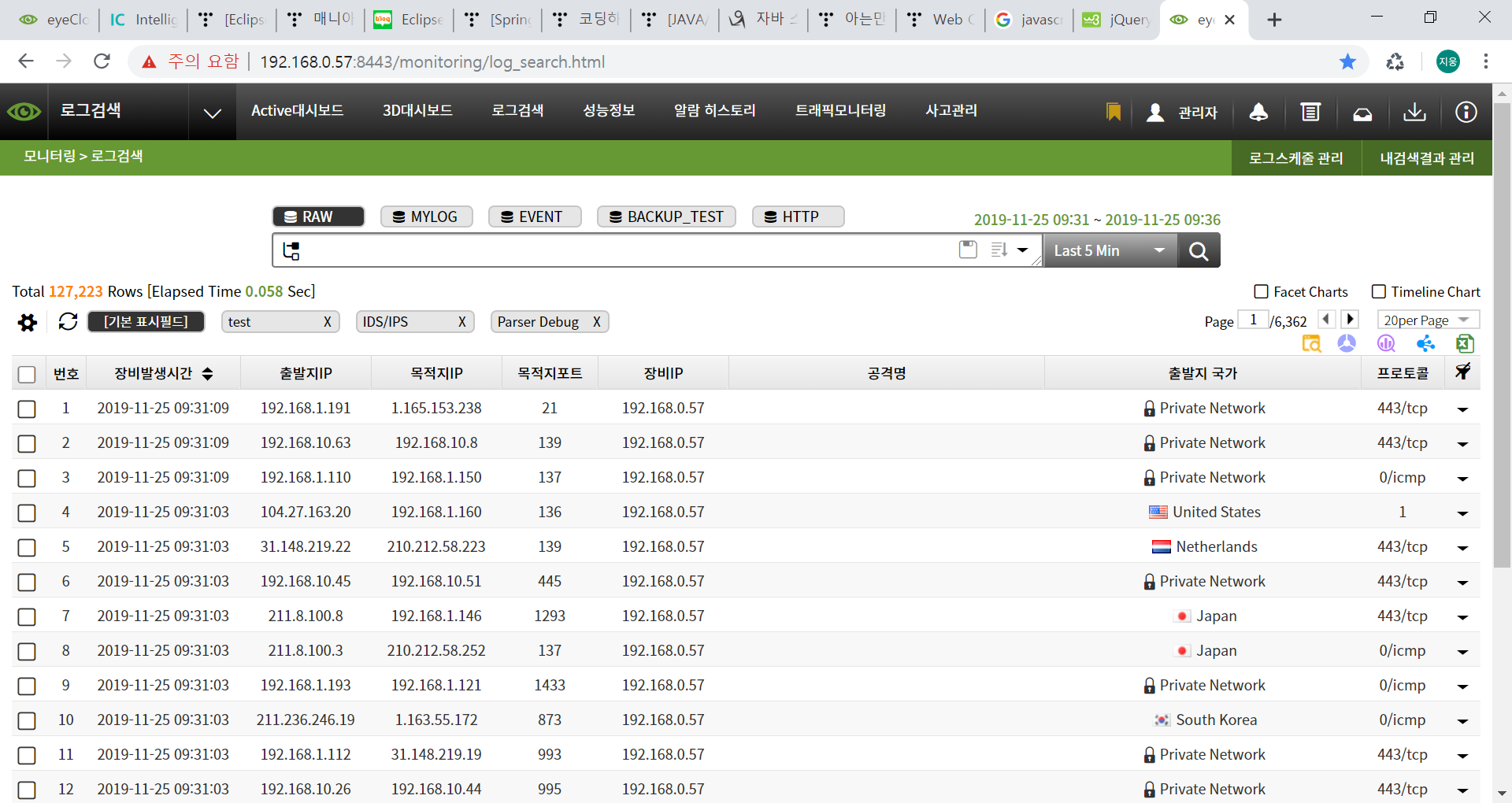
Jsp – setting\_conf.jsp

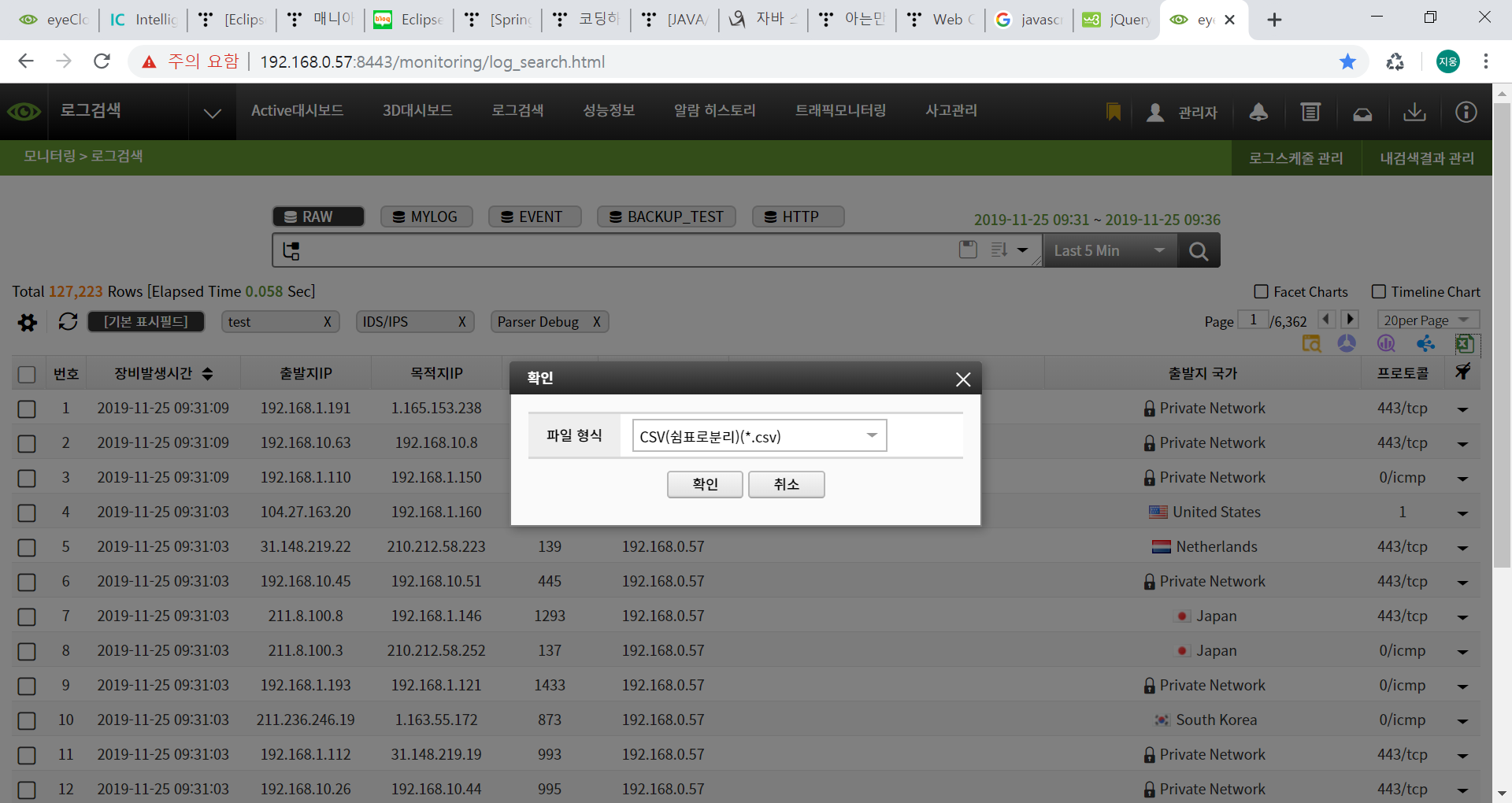


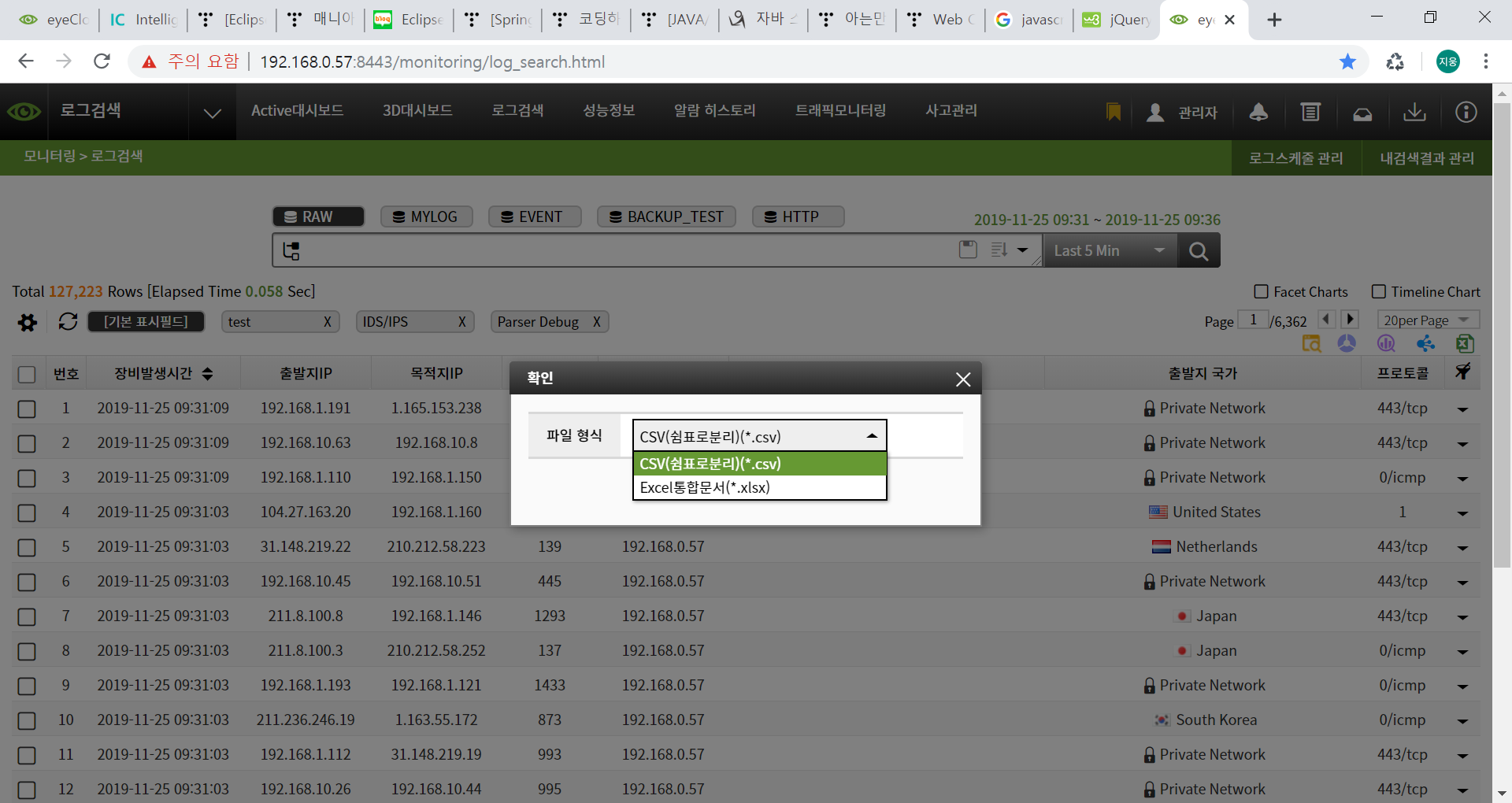
설정을 기본으로 초기화 시킬 수 있는 버튼

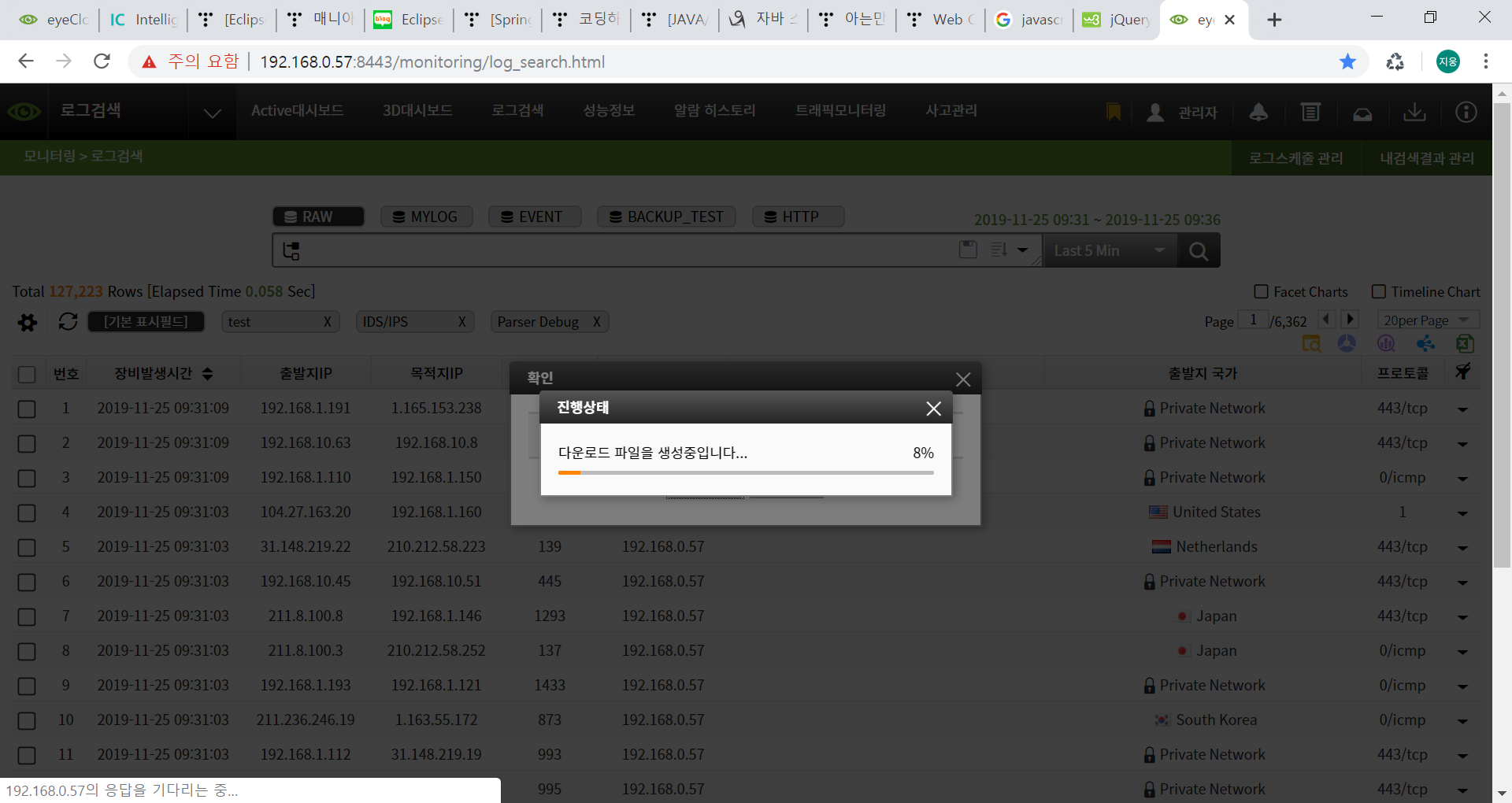
* 초기화 버튼 클릭시 기존에 사용자가 설정해 놓은 값들을 기본 값으로 설정

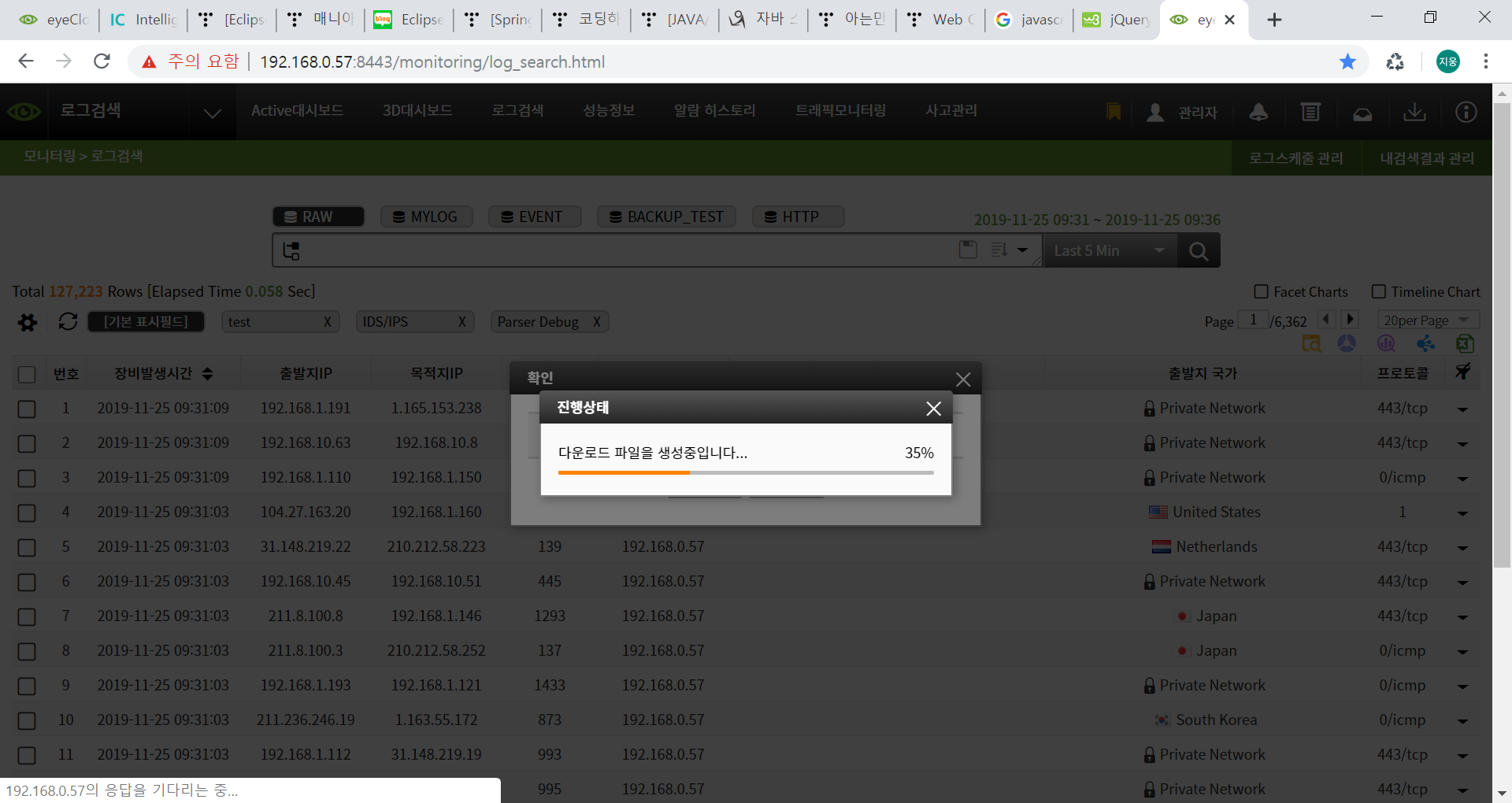
**Excel, CSV 다운로드**

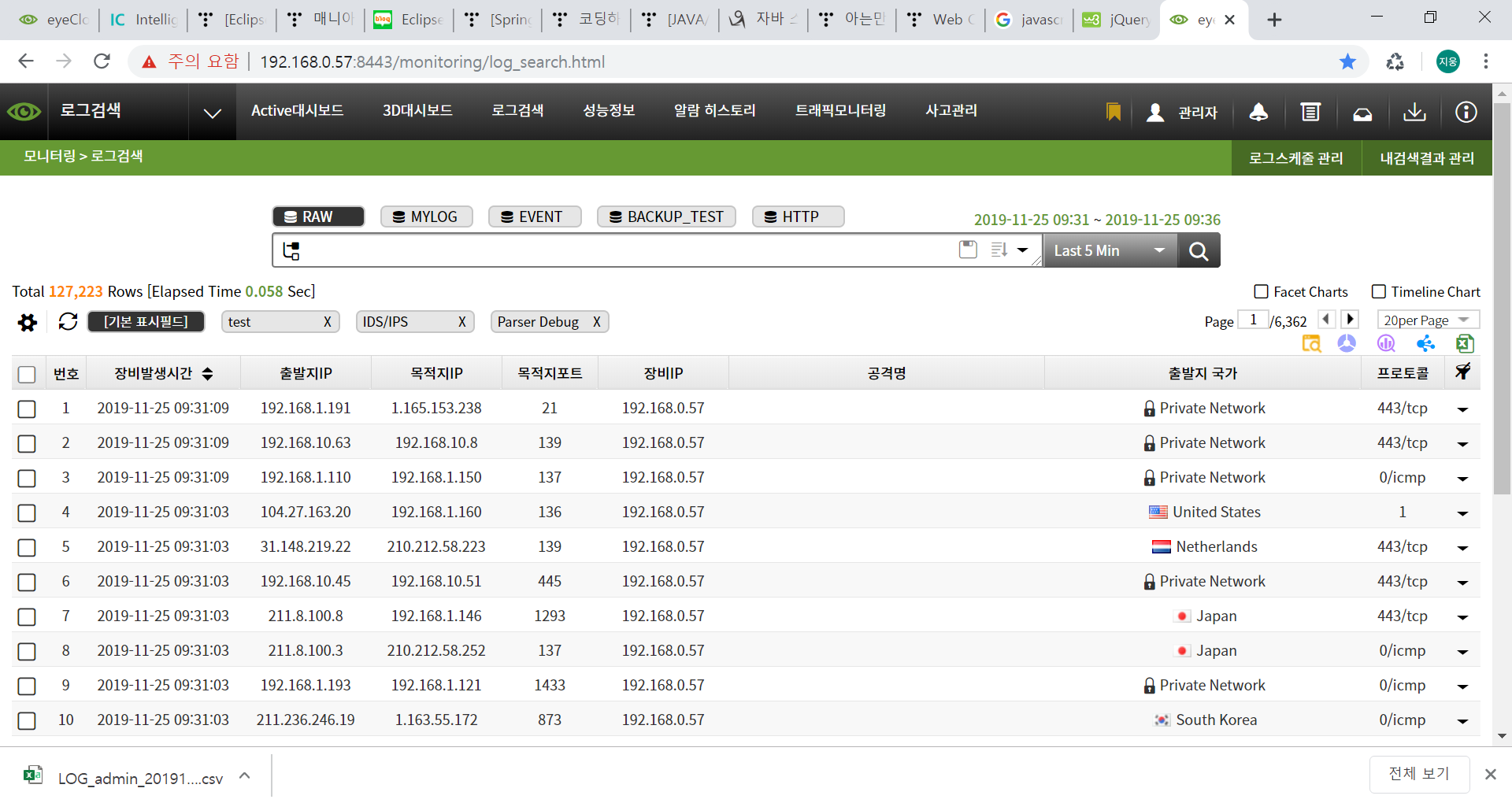




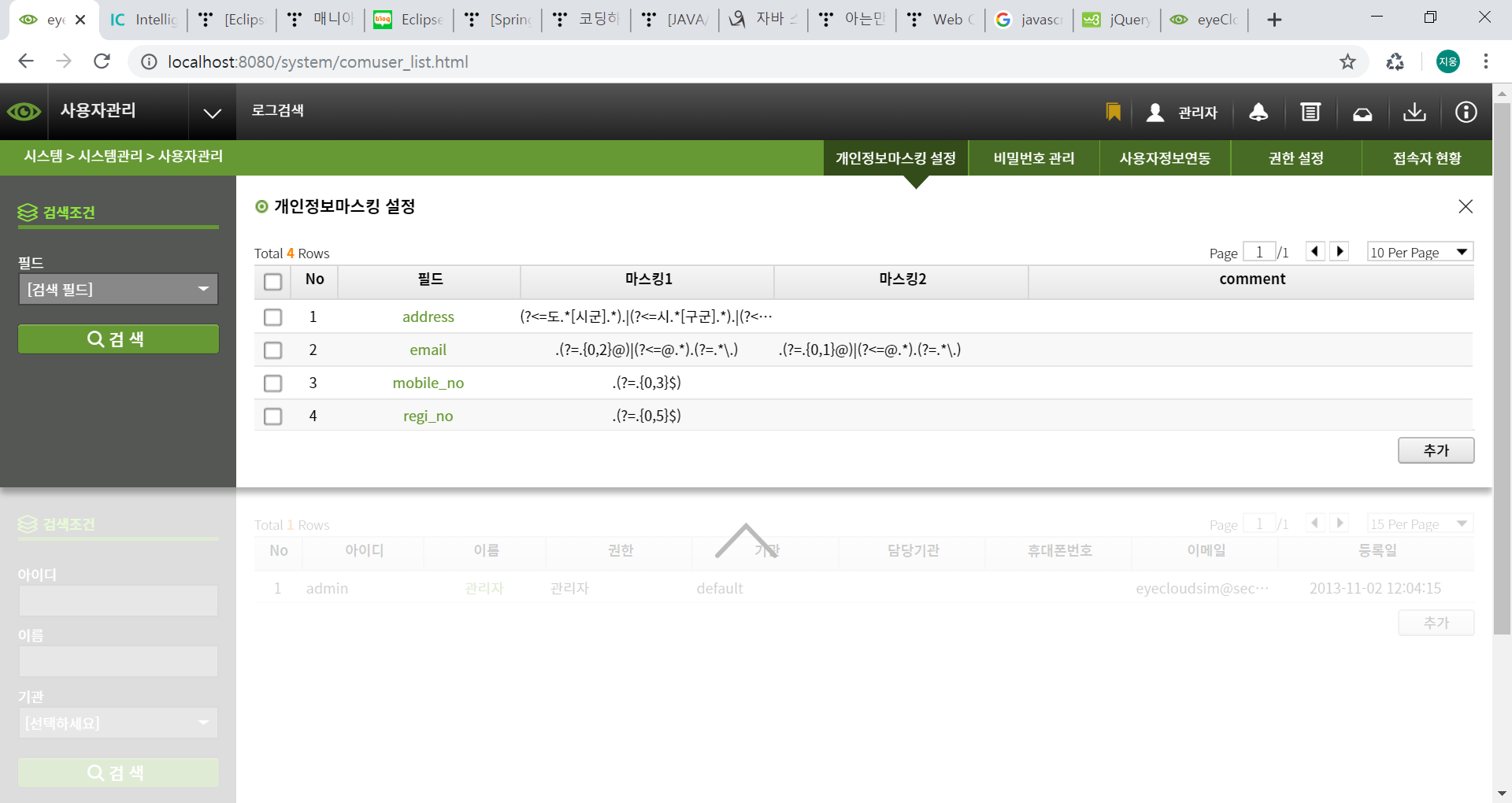


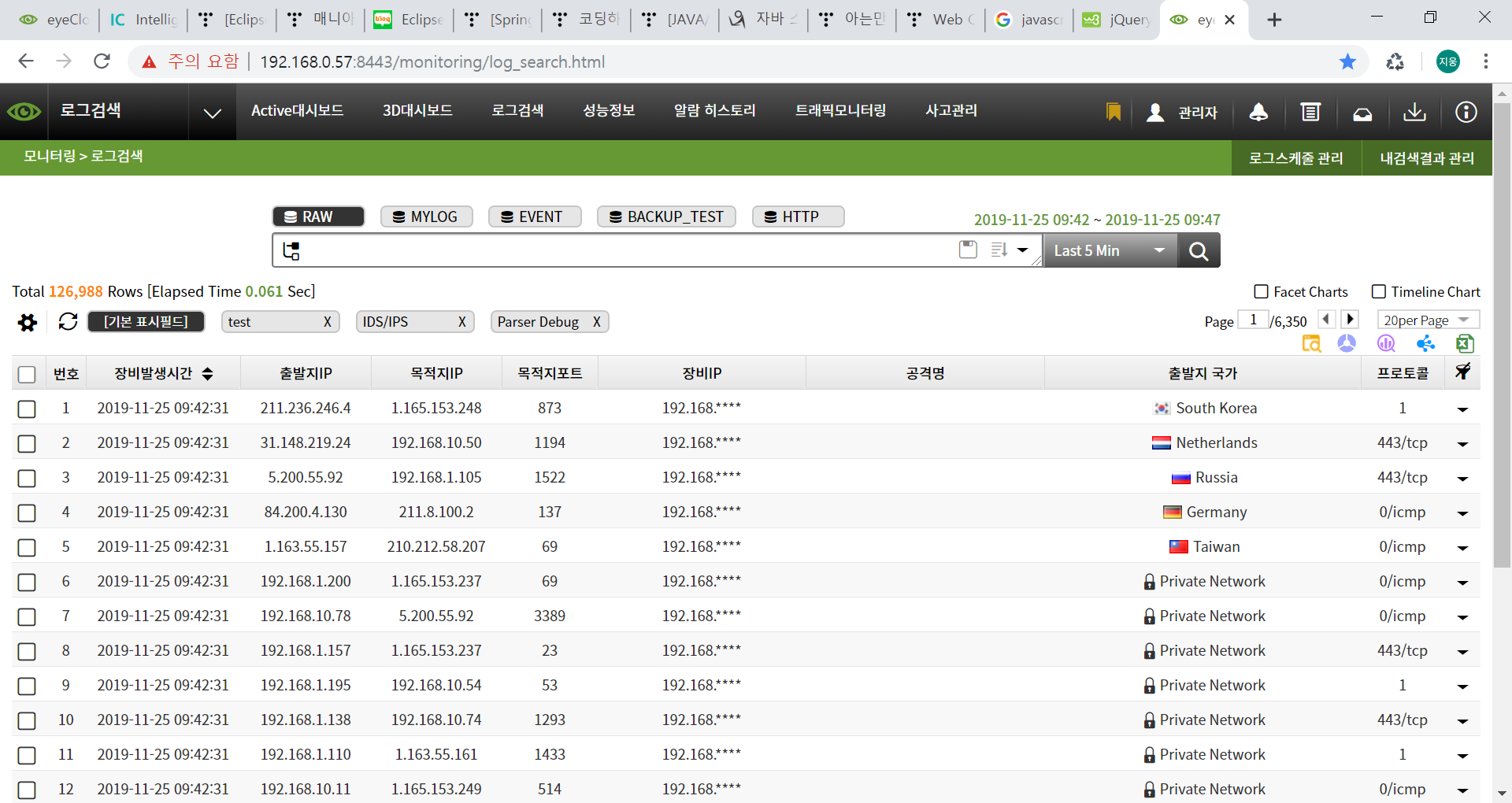




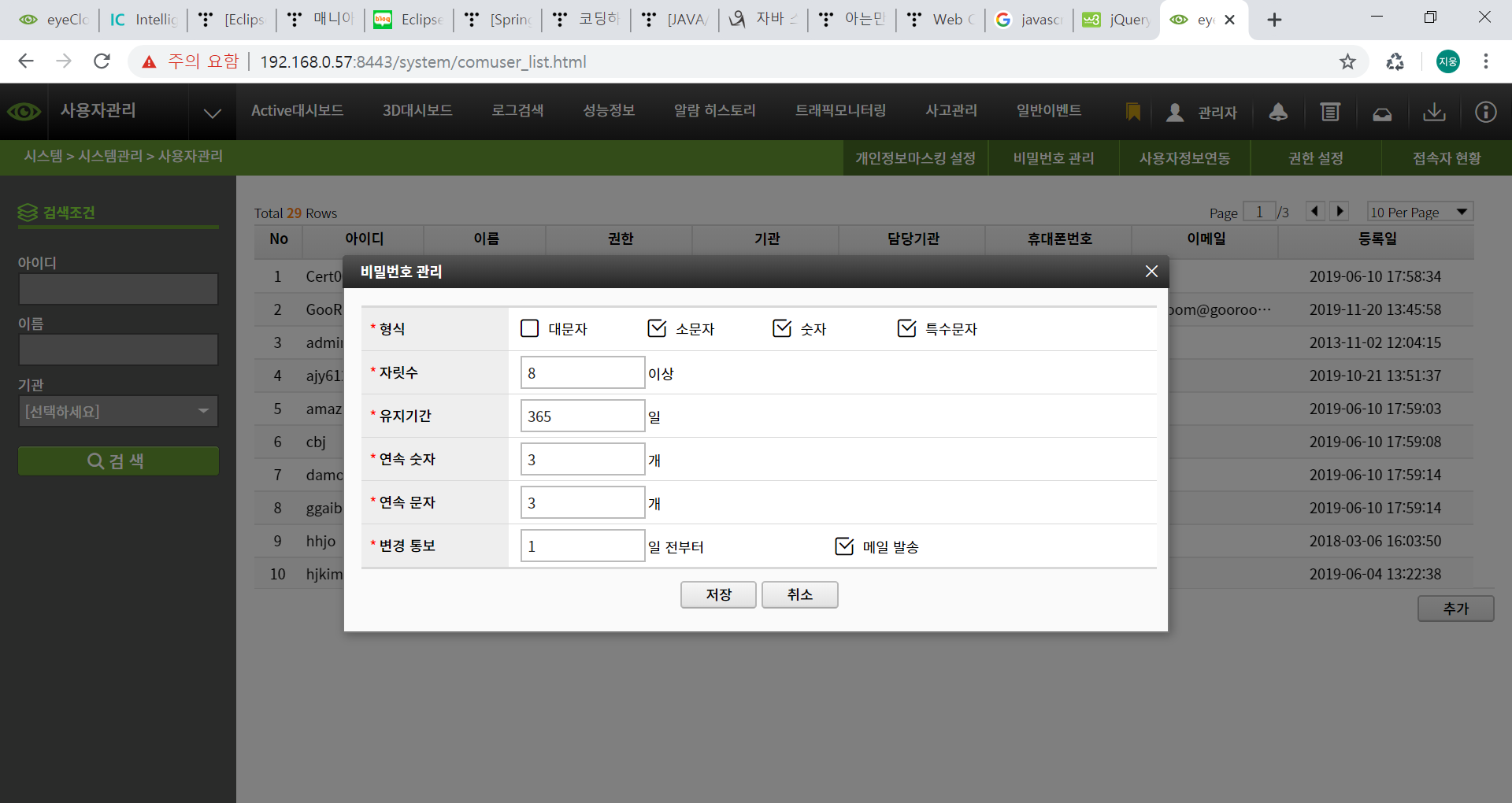


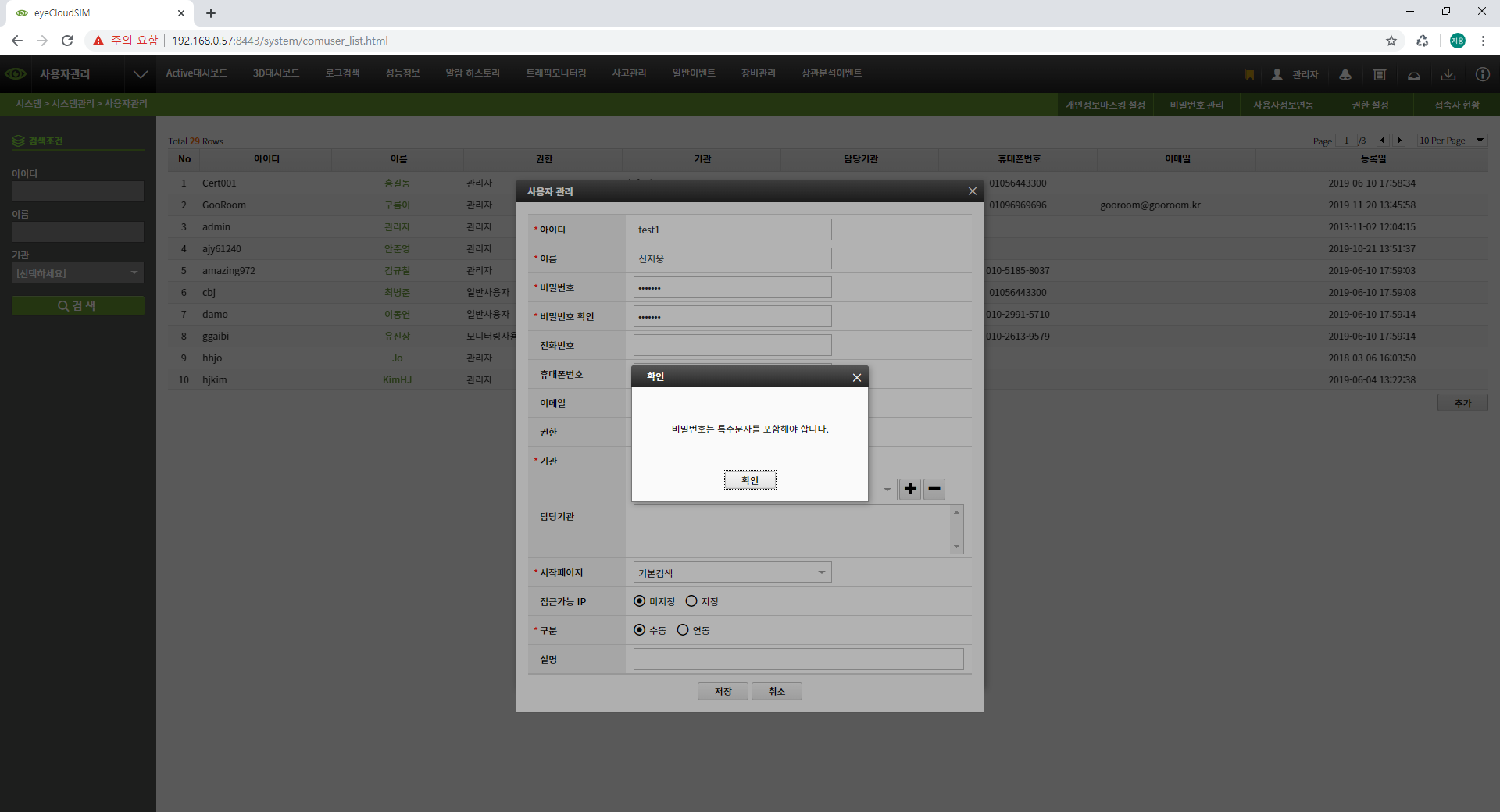
**개인정보 마스킹 설정**





**비밀번호 관리**

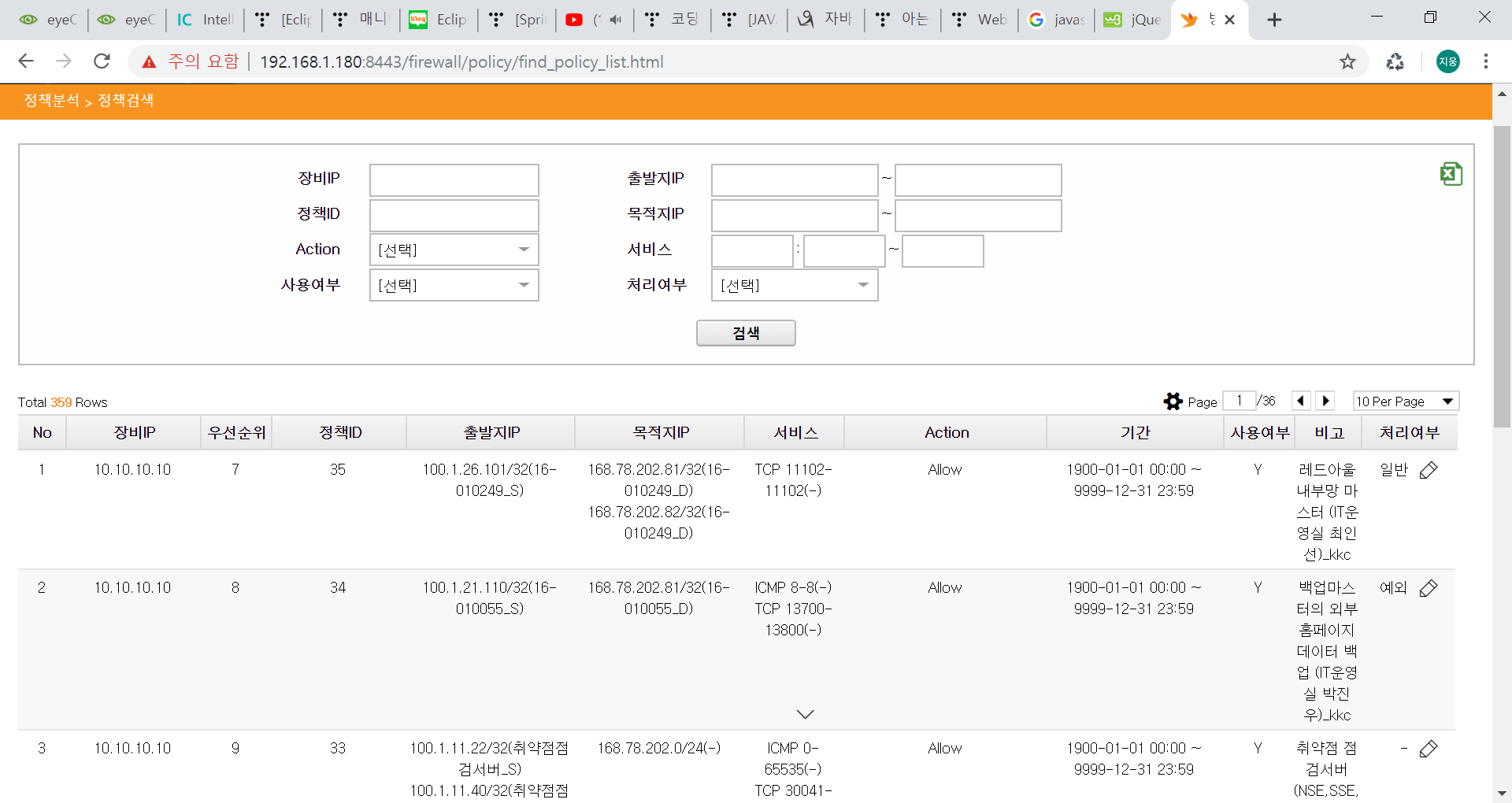


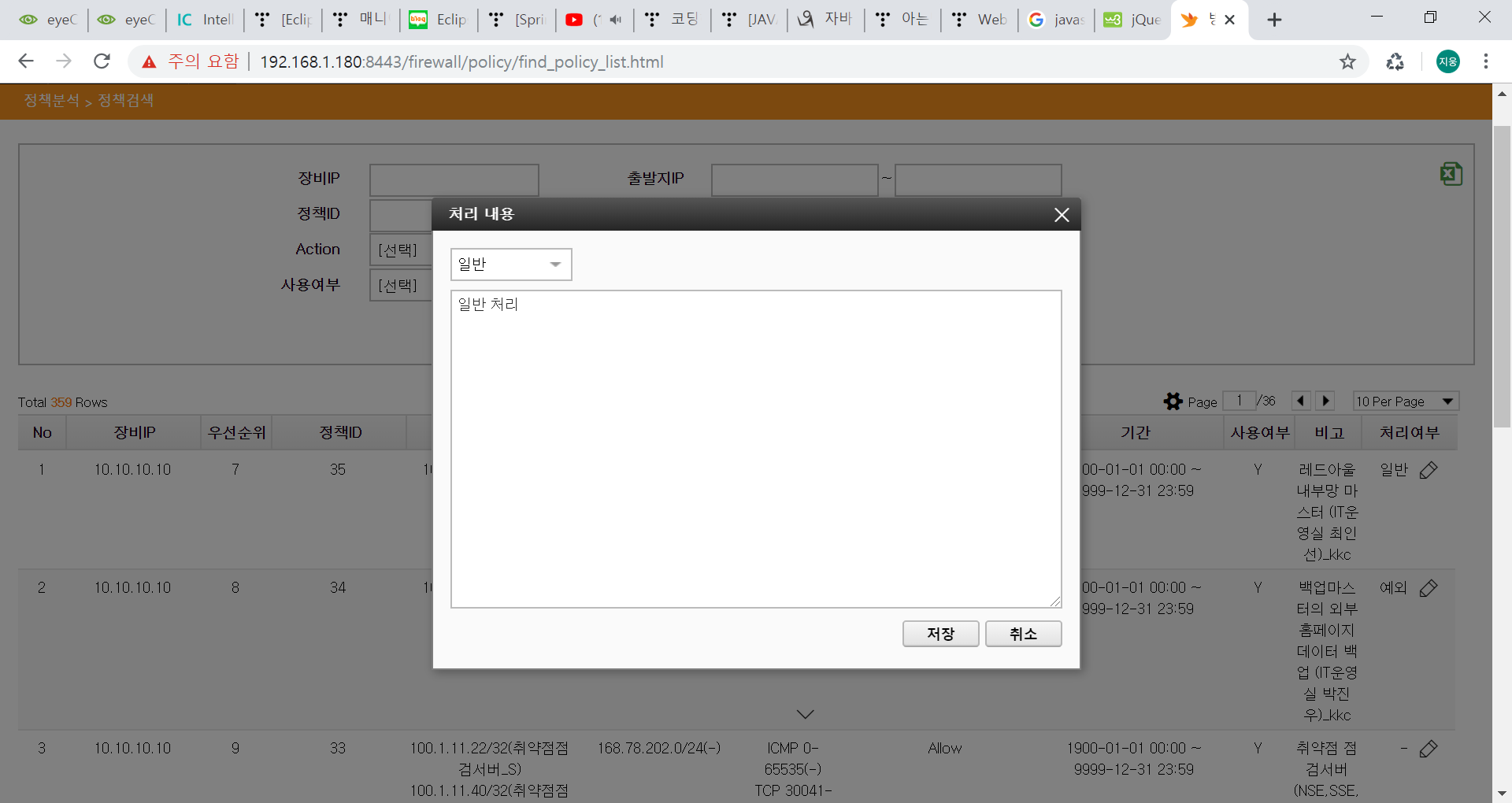


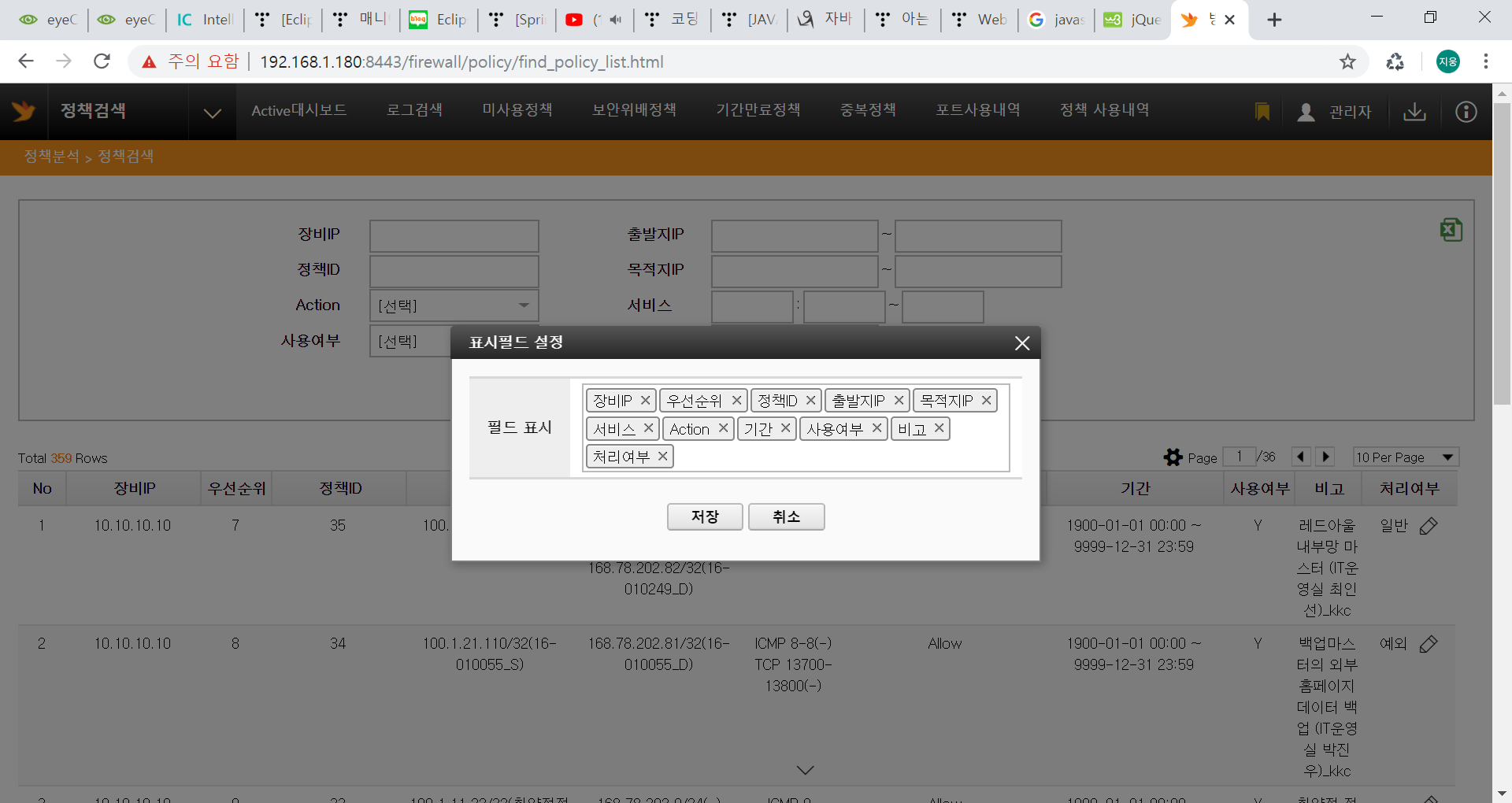
**IBK실시간 대시보드 현황**



**정책검색**

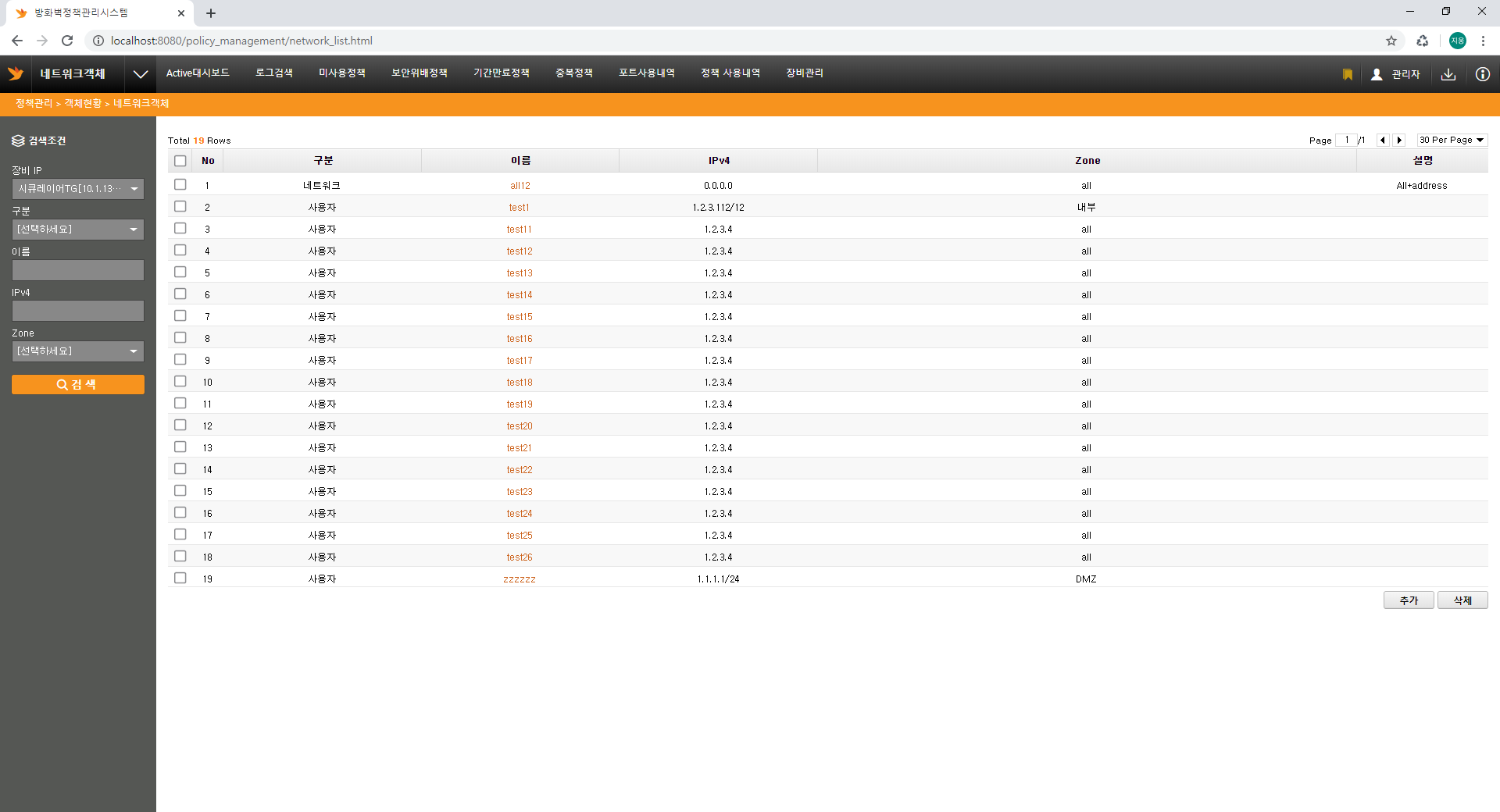




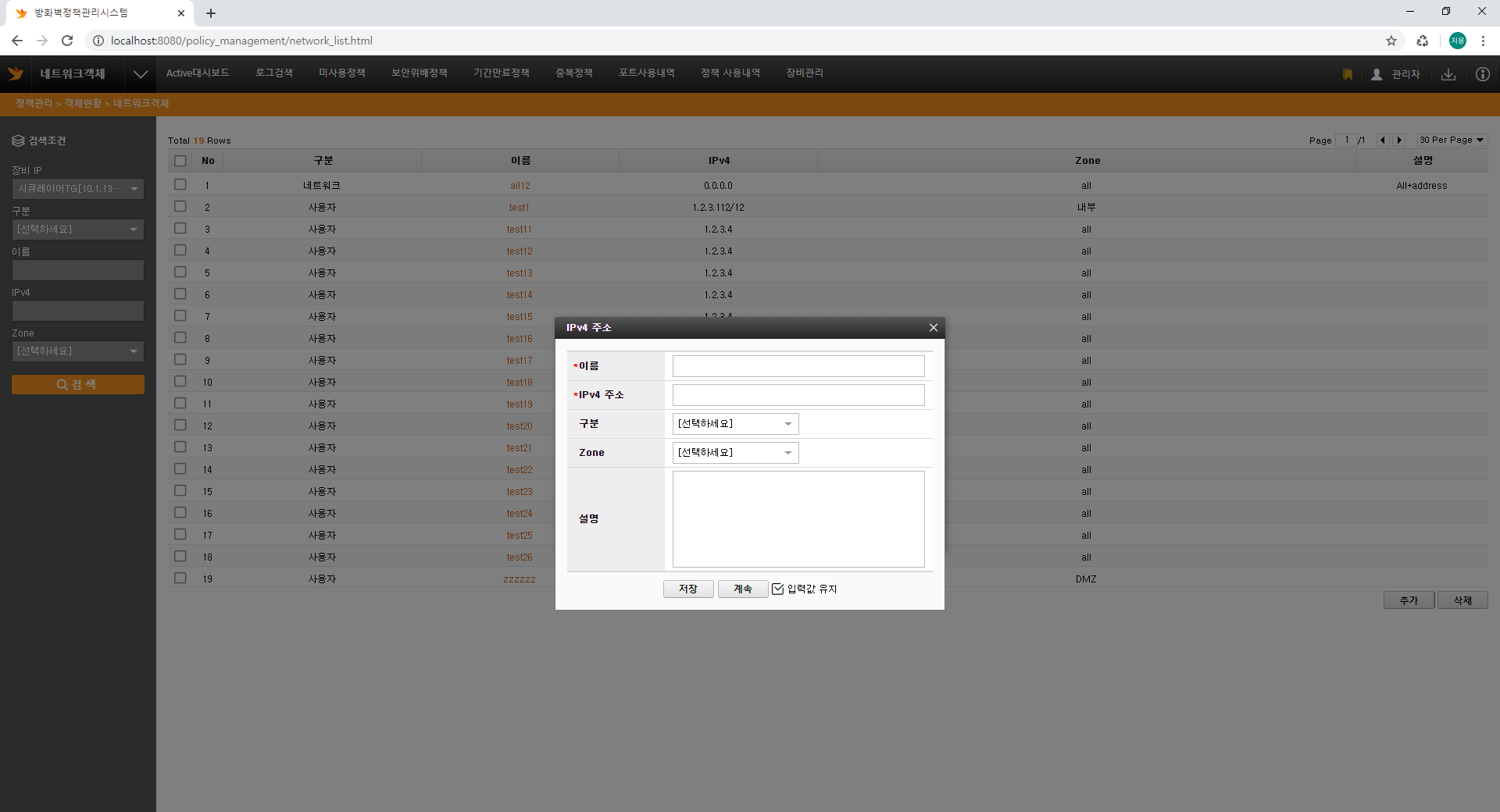


방화벽 보안 개발

* 네트워크 객체별 관리



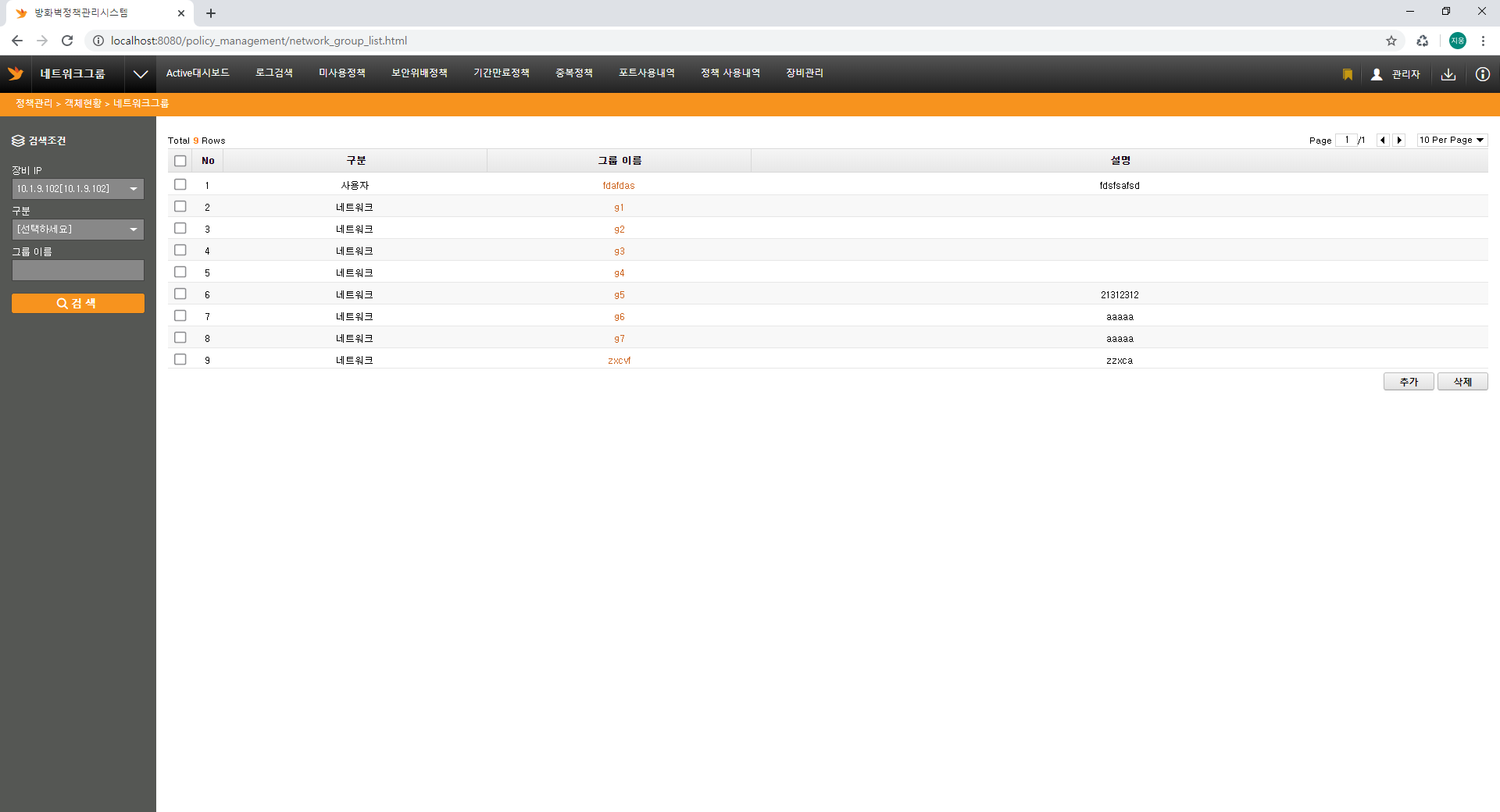
네트워크객체 추가 화면



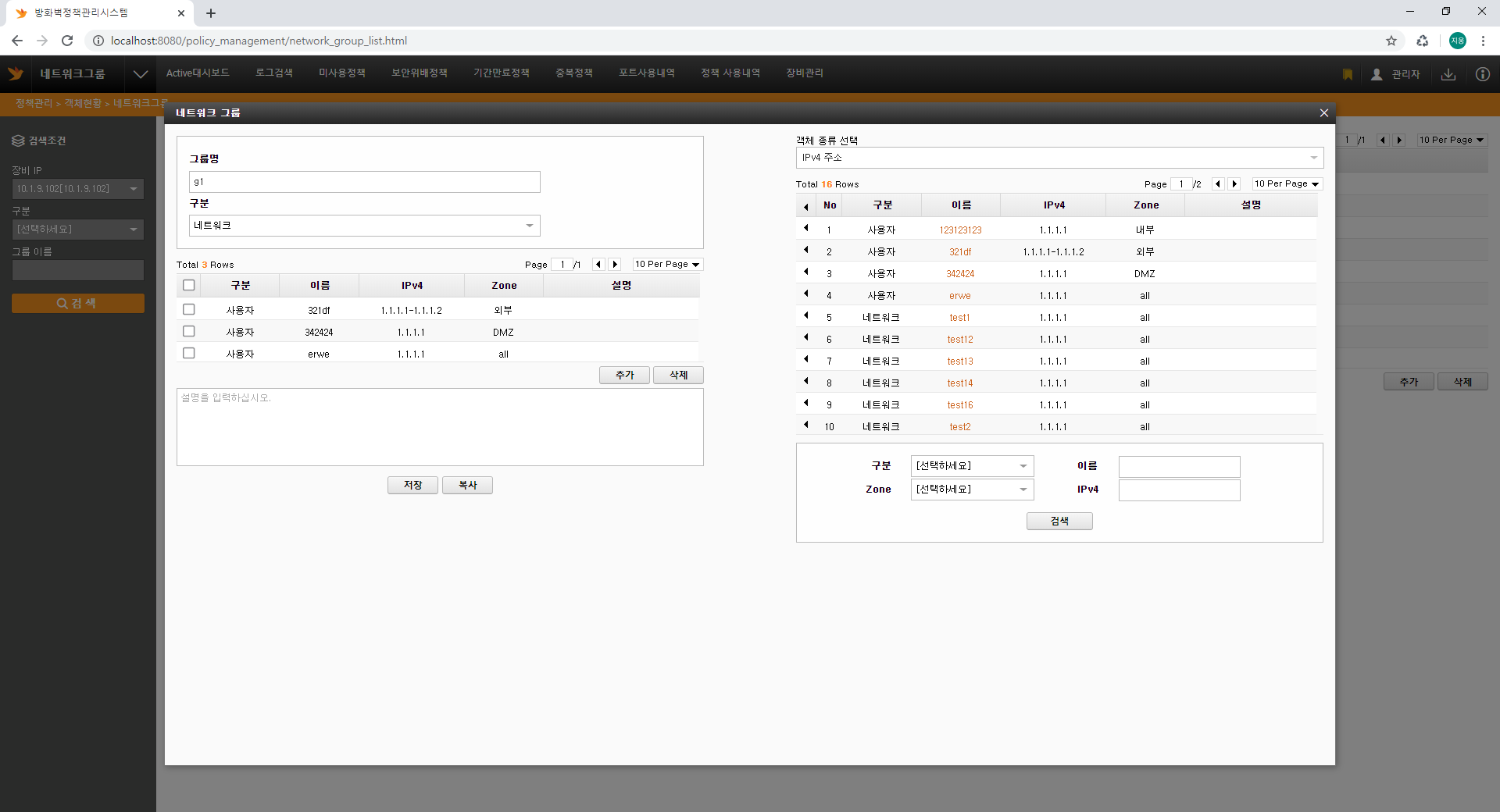
네트워크객체 수정 화면



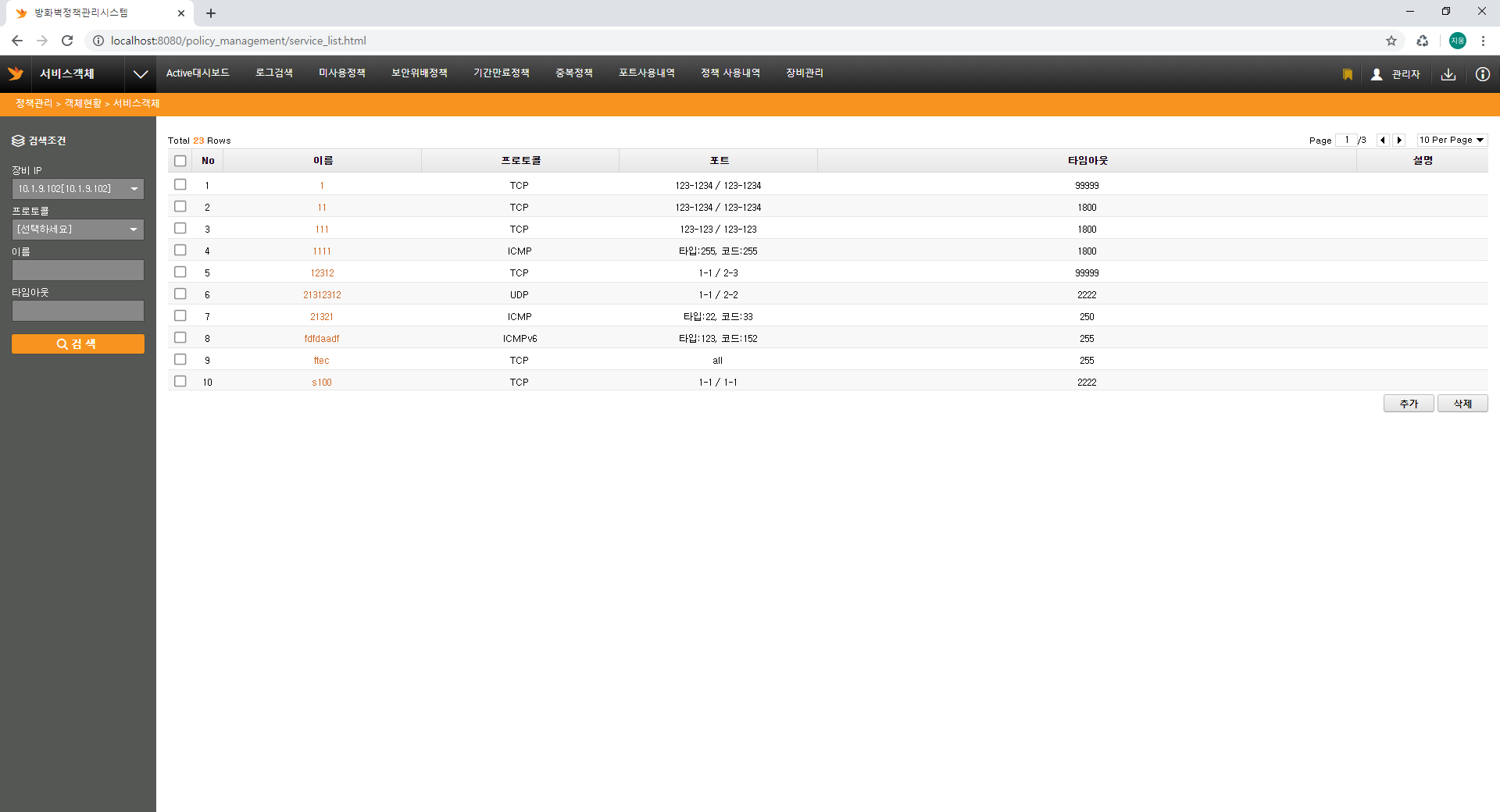
네트워크객체를 그룹화해서 관리하는 화면



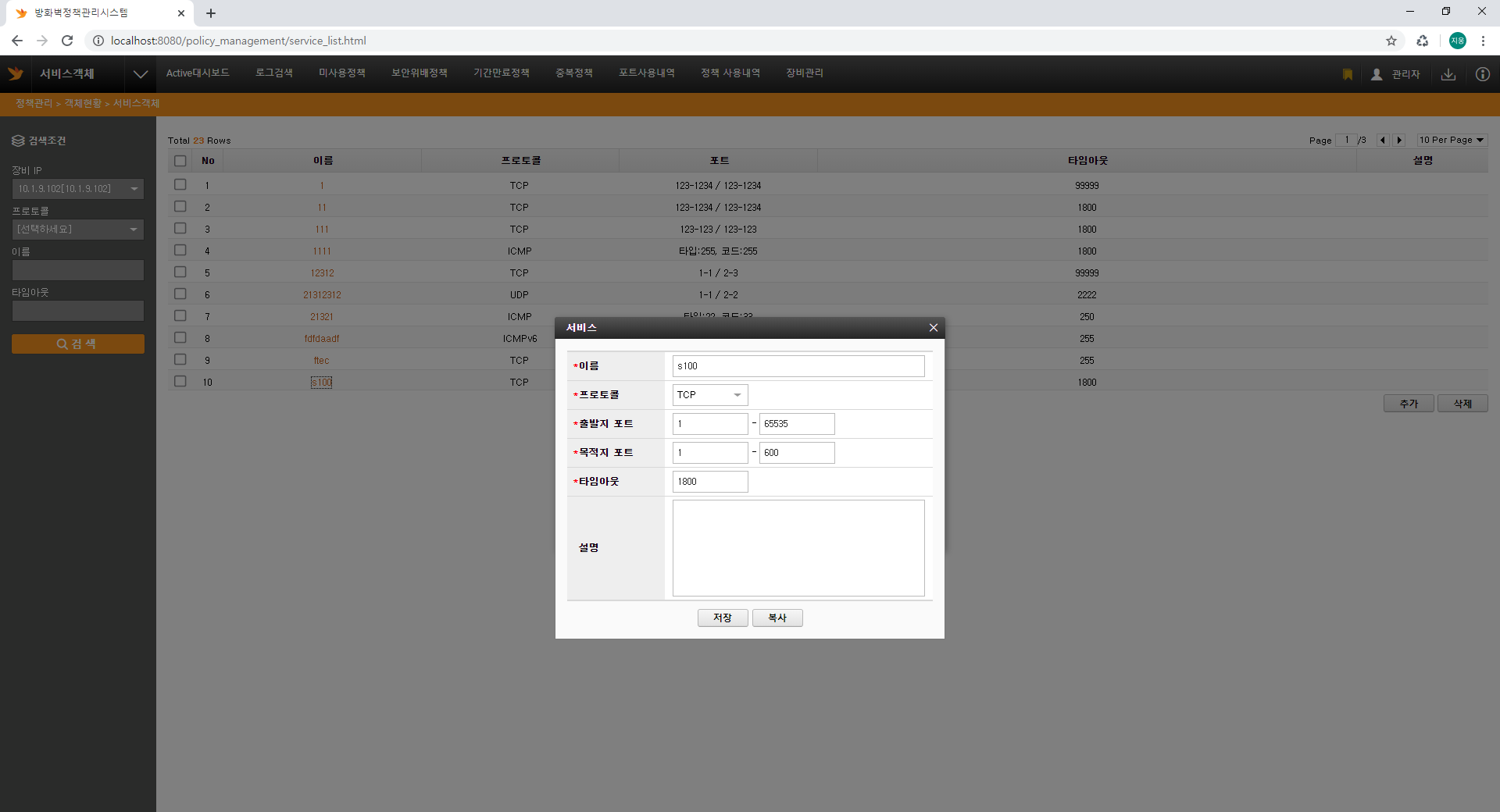
네트워크객체 그룹을 수정 및 추가 화면



서비스객체 리스트 화면

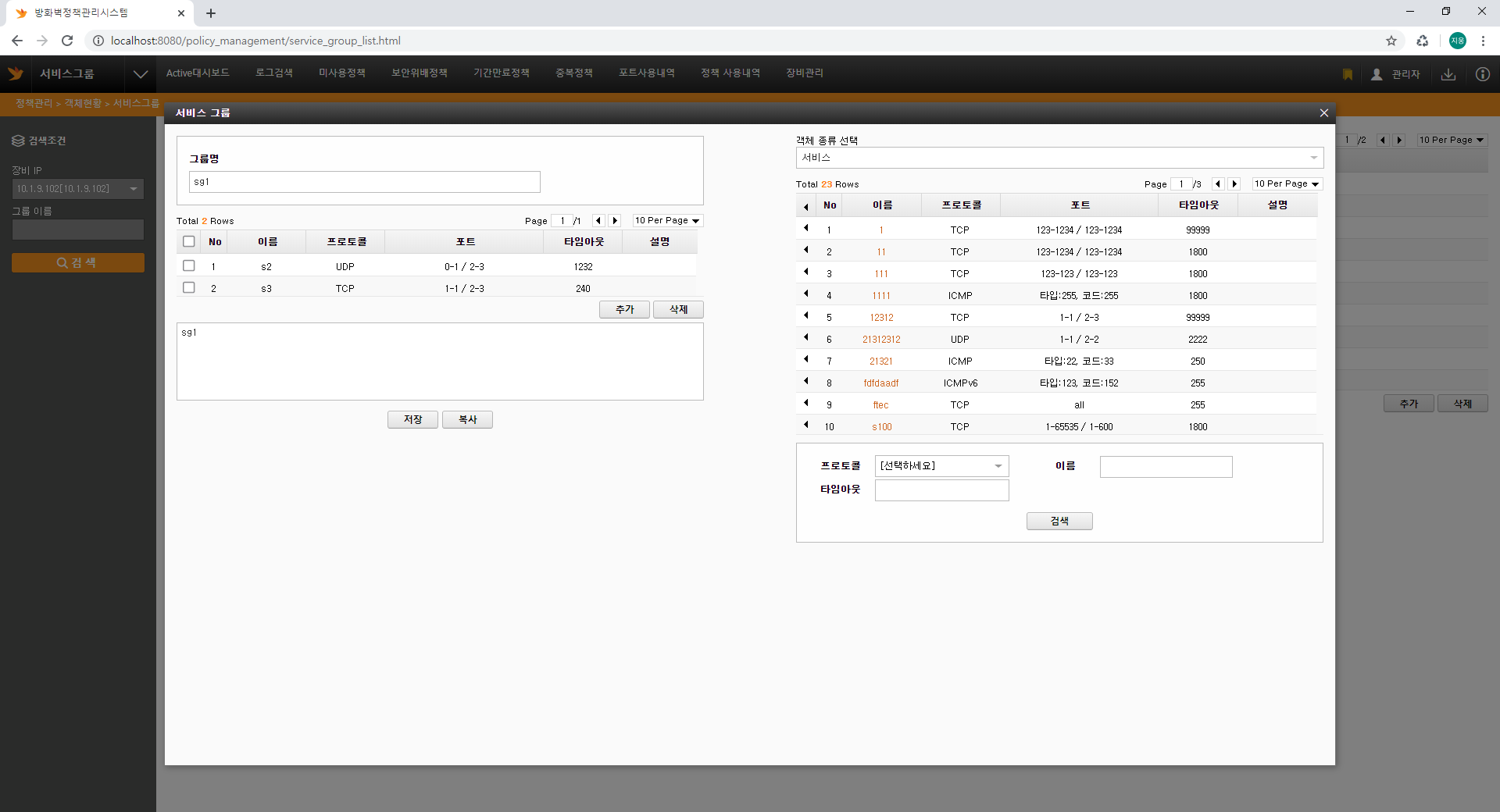


서비스 객체 개별 설정 및 추가화면

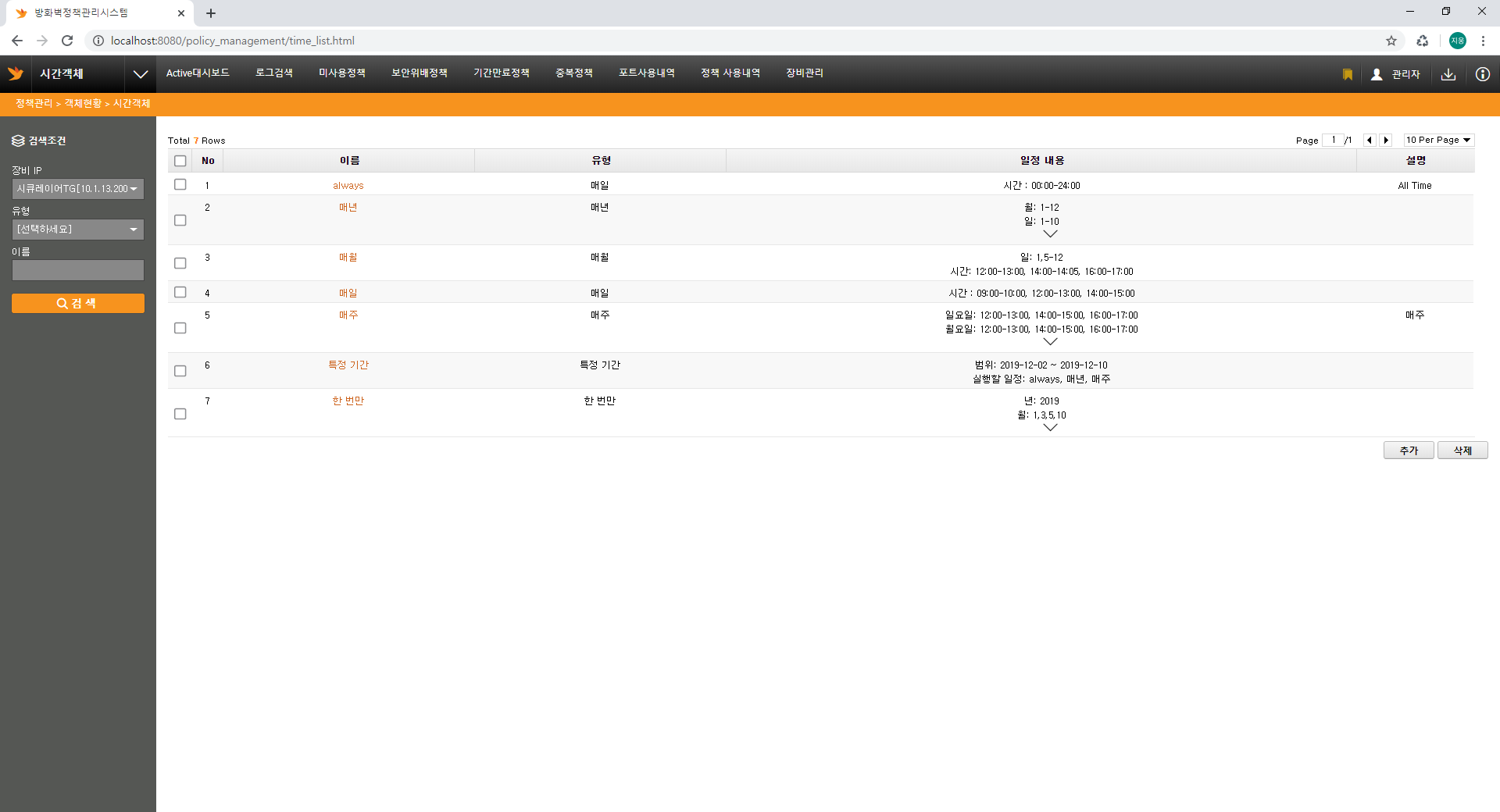


프로토콜 별로 입력하는 form이 다 다름 (TCP,UDP,ICMP,ICMPv6,IP)

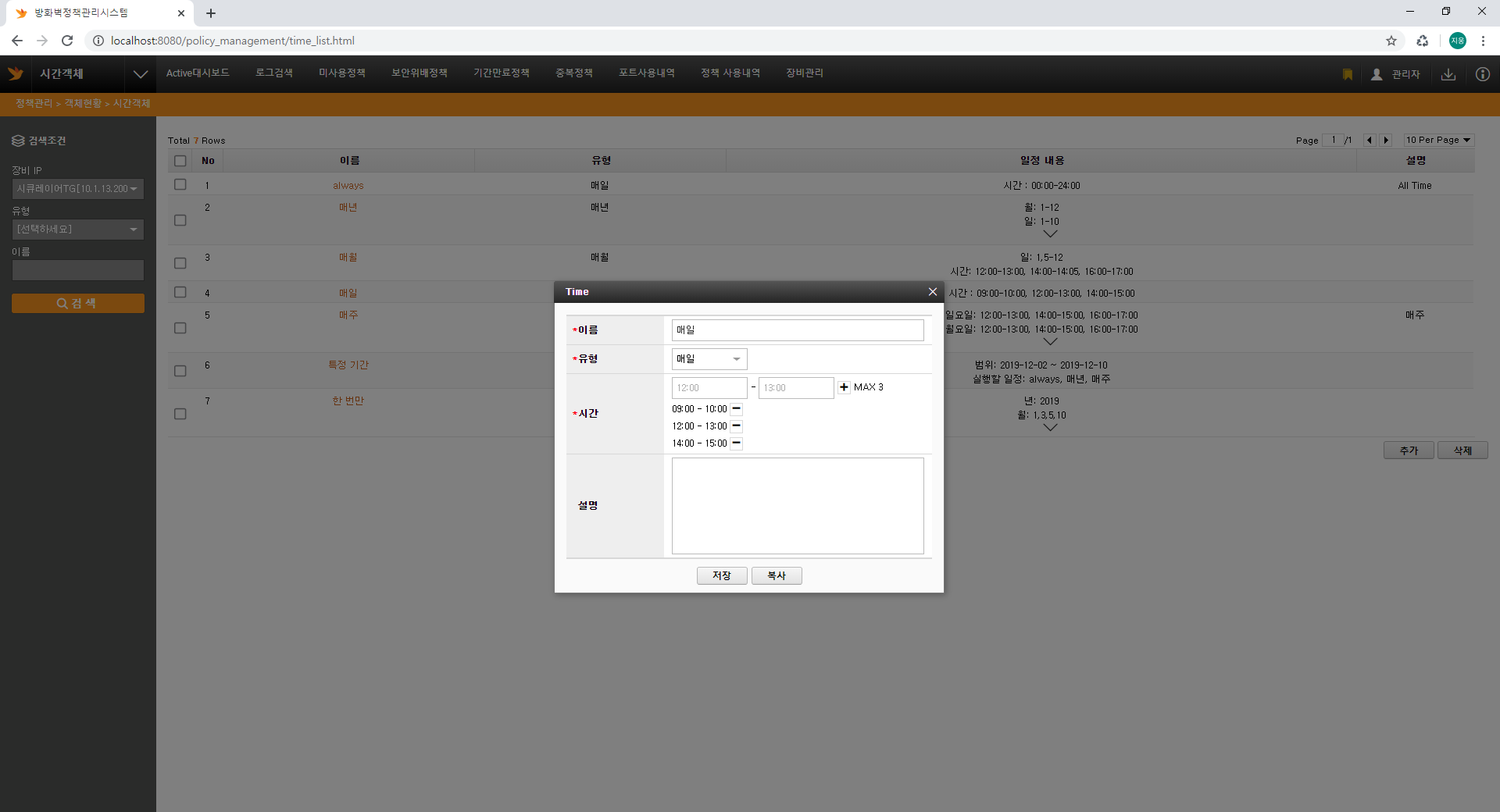
서비스객체 그룹별 관리 화면



시간 객체별 리스트 화면

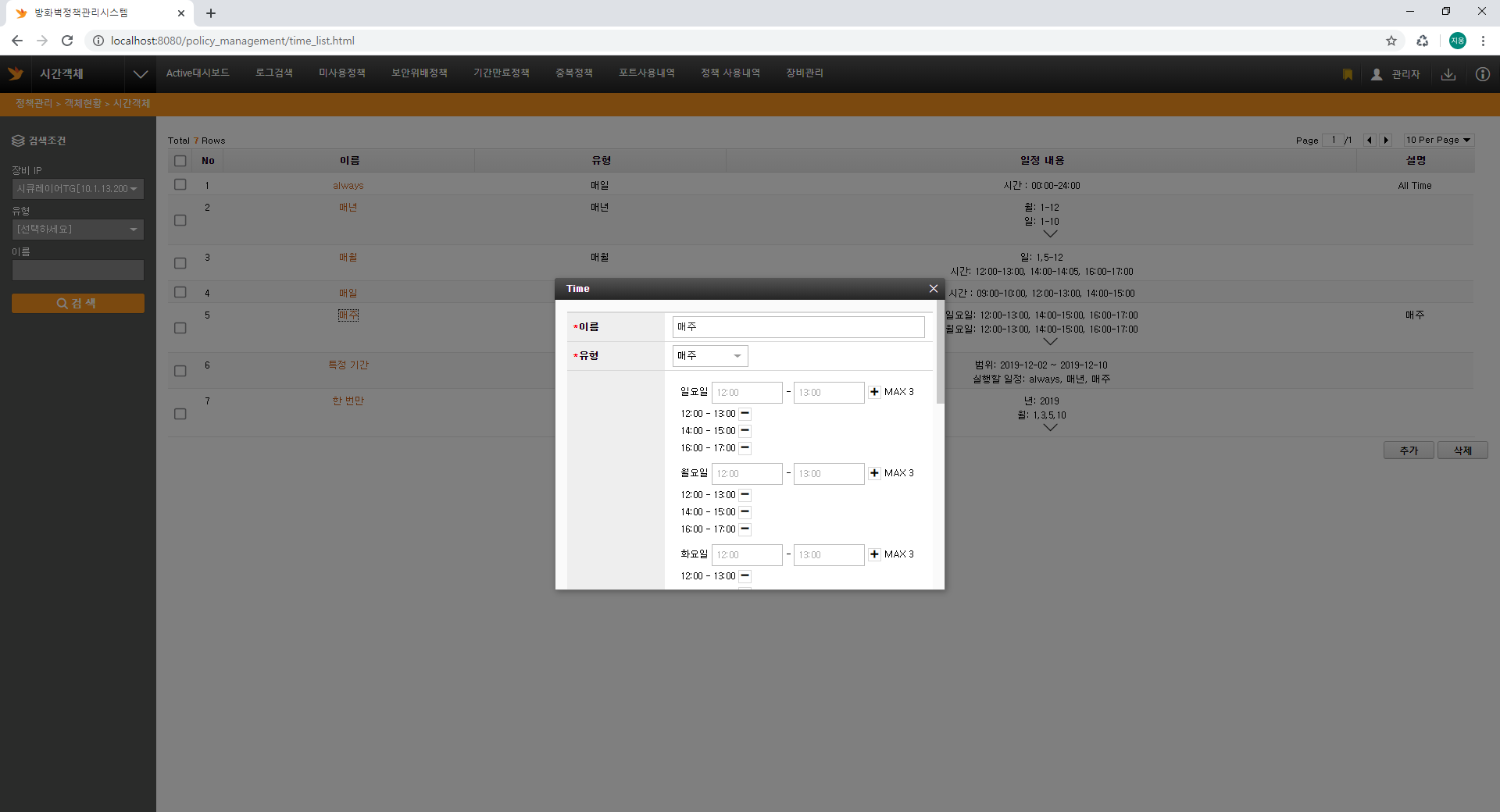


시각객체별 추가 및 수정 화면

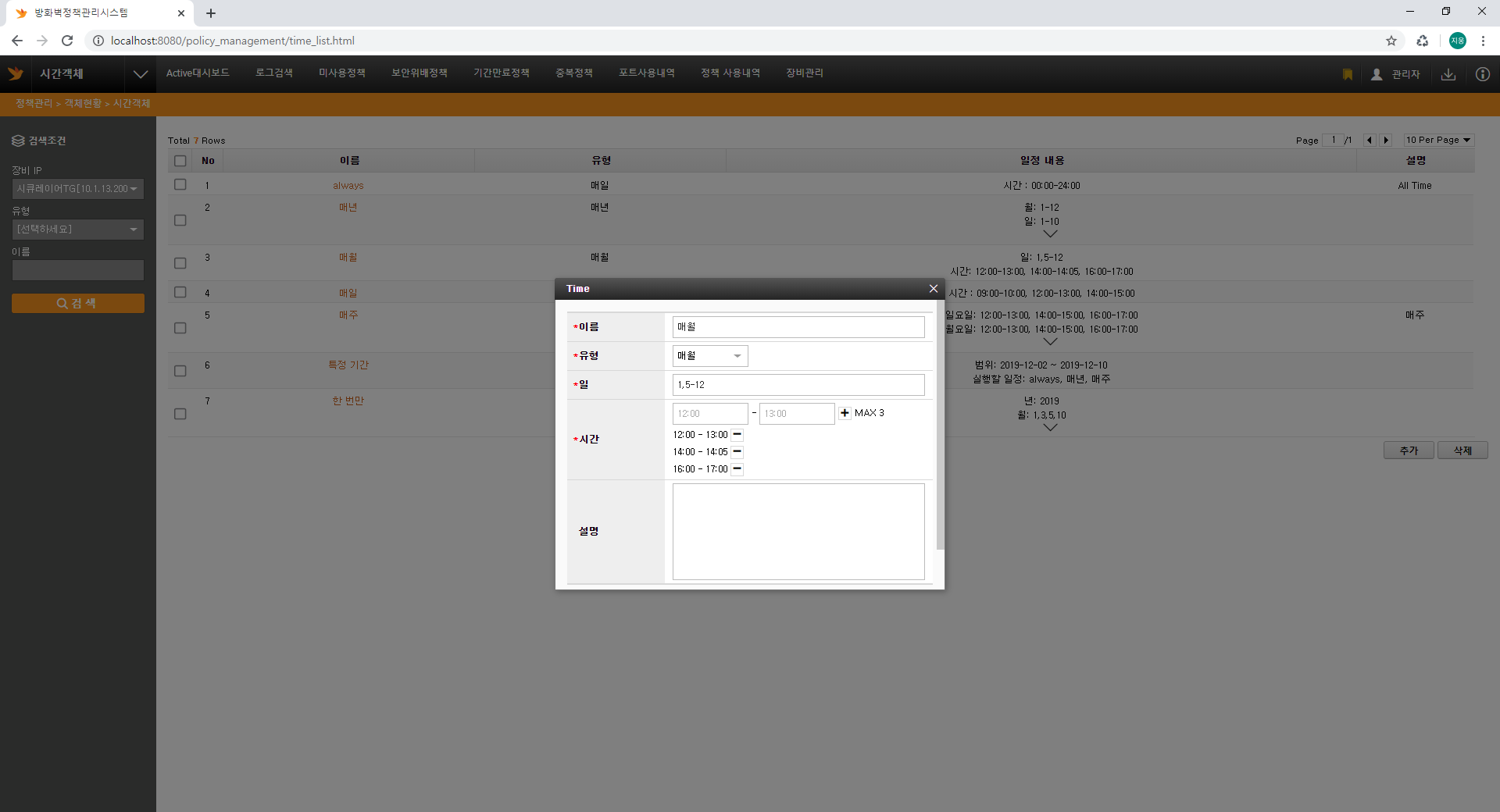


select옵션으로 매일, 매주, 매월, 매년, 한번만, 특정 기간을 선택해서 입력가능

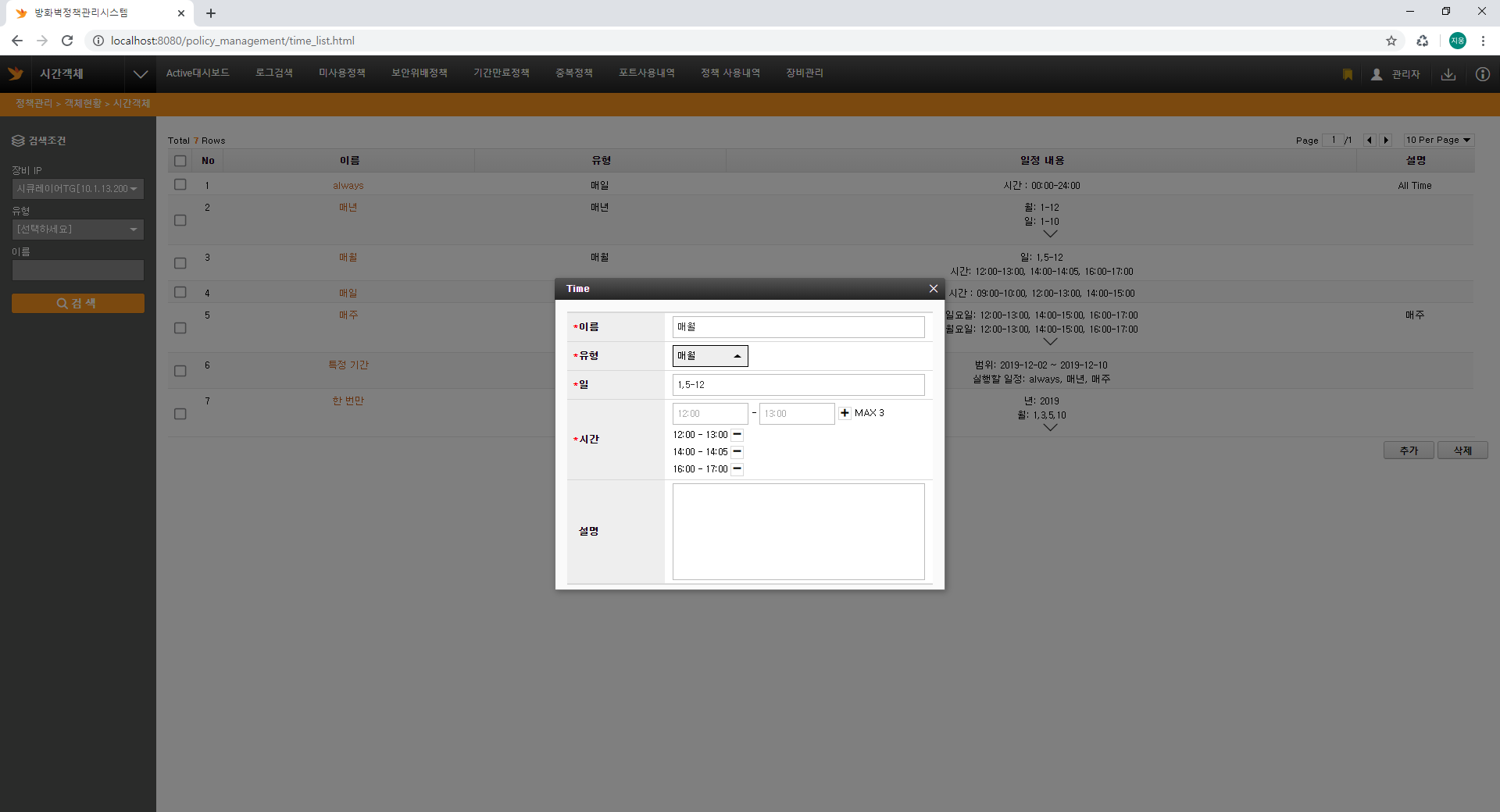
매주:



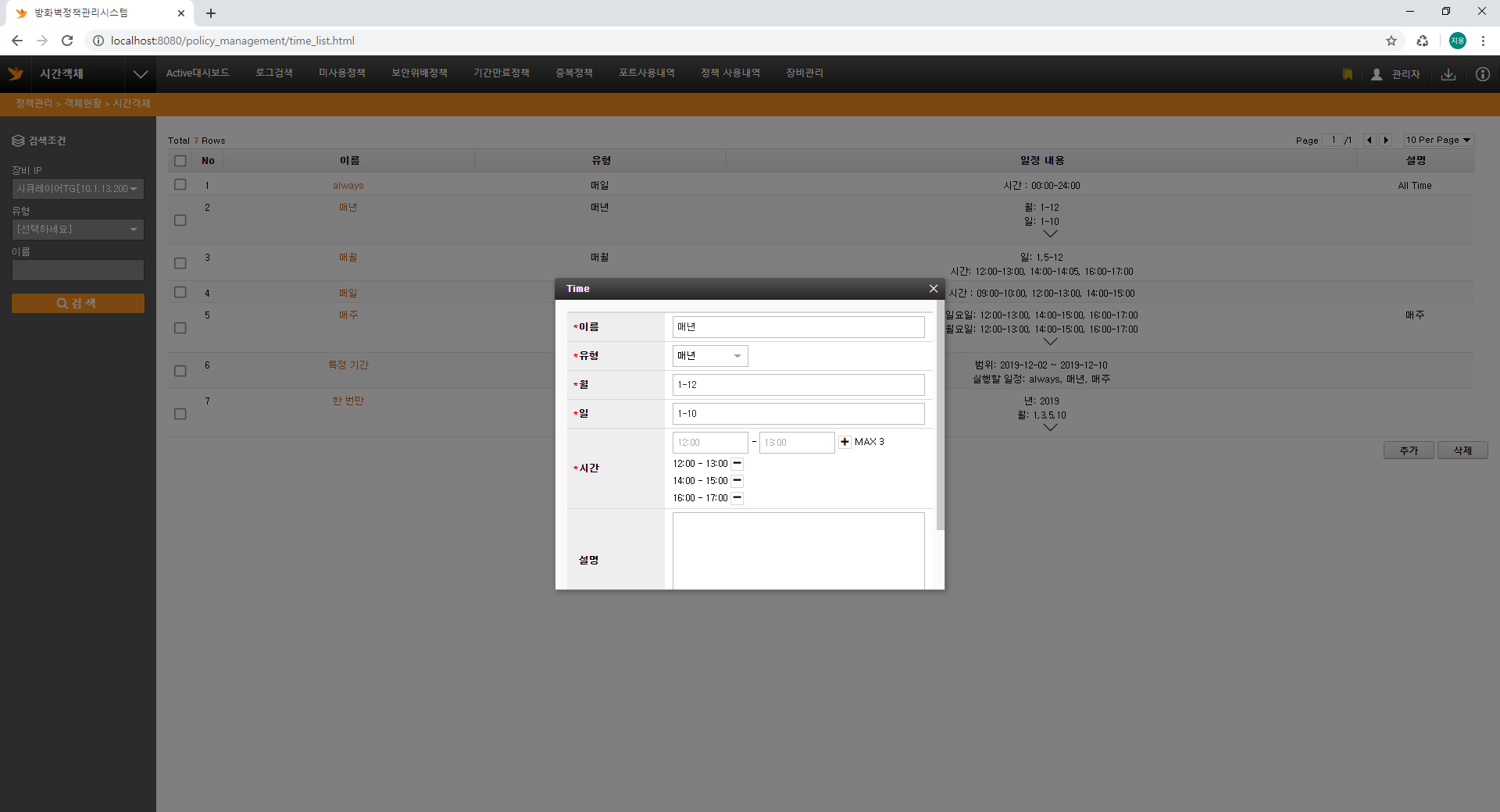
매월:



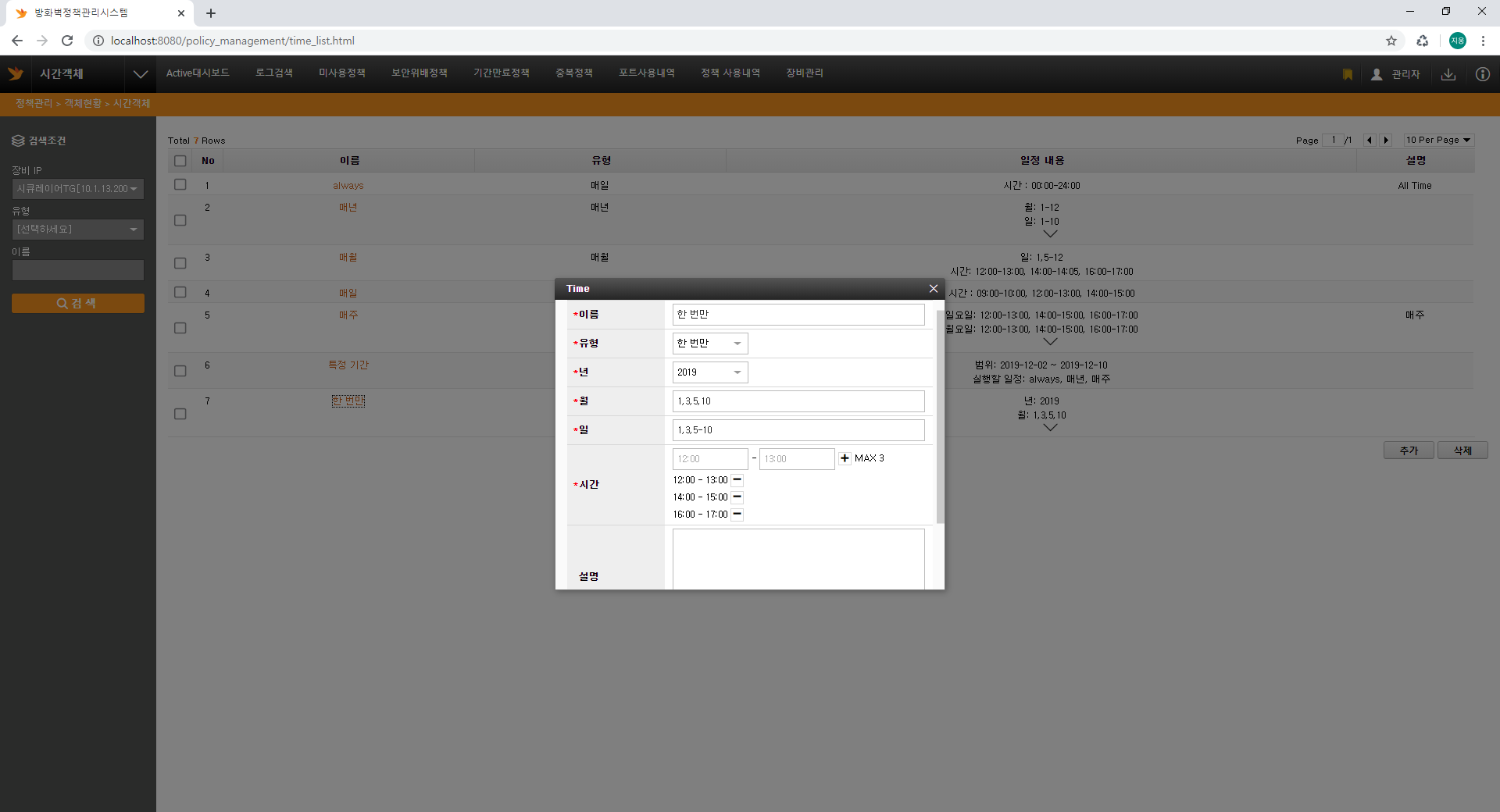
매월:



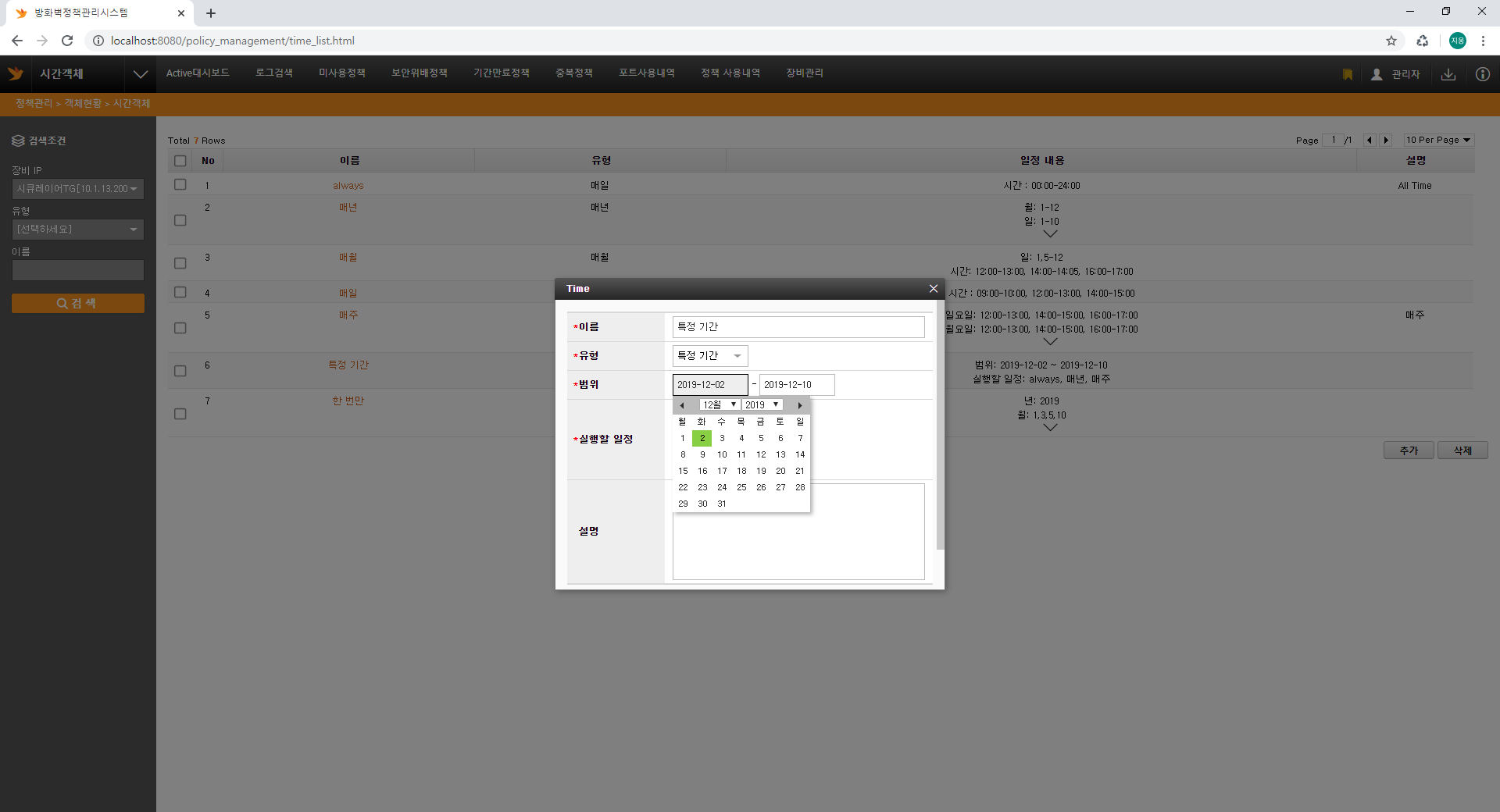
매년:



한번만



특정기간:



달력으로 날짜를 선택해서 관리 가능.