

## 개요

실시간 공지 서비스는 HTML5를 지원하는 사용자 브라우저에게 서버가 주기적으로 공지내용을 전송하며, 전송된 공지내용을 사용자 브라우저에서 주기적으로 노출하는 기능 및 관리자가 공지내용을 관리하는 기능을 제공한다.

본 컴포넌트 입장에서 바라본 활용사례로는 서버측에서 브라우저로 지정 시간에 맞도록 주기적으로 공지내용을 PUSH하여 브라우저가 이를 수신하는 기능을 제공하는 것이다.

실시간 공지 서비스는 HTML5의 server sent event를 이용하여 구현하였다.

구현 내용을 살펴보면 server sent event 지원 브라우저와 미지원 브라우저를 구분하여 사용자의 모바일 브라우저상에서 공지 메시지를 수신 및 거부하며, 신규메시지 수신을 대상으로 구현하였다.

\* 기능흐름



본 컴포넌트의 모바일 화면은 모바일 디바이스에 탑재된 브라우저를 대상으로 개발 및 테스트 검증이 완료되었다. 또한 동일 제조사의 동일한 브랜드라도 PC용 브라우저와 mobile용 브라우저의 HTML5, CSS3 에 대한 지원이 상이함으로 개발시 유의가 필요함을 밝혀둔다.

검증 대상 디바이스와 브라우저로는 각각 갤럭시S2, 갤럭시S, iPad2, iPhone4, 갤럭시탭1 등의 디바이스와 android용 웹 브라우저, Firefox, Safari, opera mobile 등의 모바일 브라우저를 사용하였다.

검증결과는 아래와 같다.

**Android**  
android용 웹 브라우저, Firefox(6.xx), opera mobile은 android OS 2.3(gingerbread) 에서 HTML5, CSS3 및 컴포넌트의 실행이 안정적으로 실행되는 것을 확인 하였다.  
반면 android OS 2.3(gingerbread) 하위 브라우저인 android 2.1, 2.2에서는 컴포넌트 실행에 있어 매끄럽지 못한 HTML5, CSS3 지원으로 인하여 컴포넌트의 UI 구현이 원활하지 못한 모습을 보였다.

opera mobile의 경우 UI처리시 색상 및 이벤트 처리에 문제를 야기하였으며 그로인해 opera mobile은 3중 지원브라우저 대상에서 제외 되었다.  
테스트 모바일 디바이스로는 갤럭시, 갤럭시S2, 갤럭시탭1을 채택하였다.

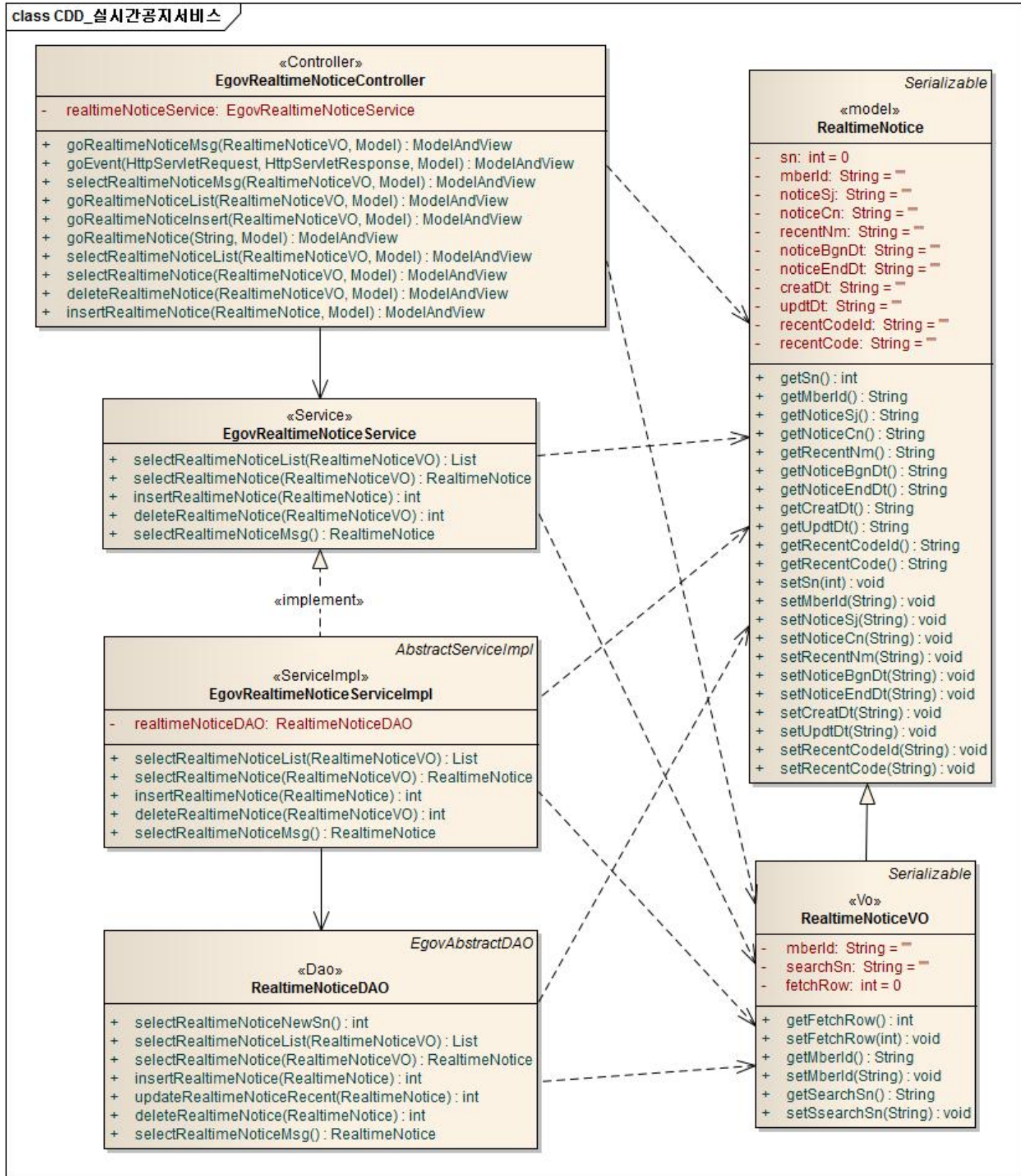
**iOS**  
iOS는 3중 지원브라우저 대상중 하나로 Safari 4.2, 4.3 버전에서 검증을 하였으며 비교적 원활한 HTML5, CSS3를 지원을 하였다.  
테스트 모바일 디바이스로는 iPhone4, iPad2를 채택하였다.

- 협업 - 실시간 공지 서비스
  - 개요
  - 설명
    - 설정 및 주의사항
    - 실시간 공지 모바일 목록 조회
    - 실시간 공지 모바일 등록
    - 실시간 공지 모바일 상세 조회
    - 실시간 공지 모바일 메시지 수락확인
    - 실시간 공지 모바일 메시지 확인
  - 참고자료

## 설명

실시간 공지 서비스는 관리기능인 실시간 공지 목록조회, 상세조회, 등록, 삭제로 구성되어 있으며, 사용자기능에서는 실시간 공지 서비스 수락/거절, 메시지 확인, 메시지수신/메시지거부 기능으로 구성되어 있다.(관련기능 부분참조)

### 클래스 다이어그램



## 관련소스

유형	대상소스명	비고
Controller	egovframework.mbl.com.rns.web.EgovRealtimeNoticeController.java	실시간 공지 Controller Class
Service	egovframework.mbl.com.rns.service.EgovRealtimeNoticeService.java	실시간 공지 Service Class
ServiceImpl	egovframework.mbl.com.rns.service.impl.EgovRealtimeNoticeServiceImpl.java	실시간 공지 ServiceImpl Class
Model	egovframework.mbl.com.rns.service.RealtimeNotice.java	실시간 공지 Model Class
VO	egovframework.mbl.com.rns.service.RealtimeNoticeVO.java	실시간 공지 VO Class
DAO	egovframework.mbl.com.rns.service.impl.RealtimeNoticeDAO.java	실시간 공지 Dao Class
JSP	/WEB-INF/jsp/egovframework/mbl/com/rns/EgovMobileRealtimeNoticeList.jsp	실시간 공지 목록조회 페이지
JSP	/WEB-INF/jsp/egovframework/mbl/com/rns/EgovMobileRealtimeNoticeInsert.jsp	실시간 공지 등록 페이지
JSP	/WEB-INF/jsp/egovframework/mbl/com/rns/EgovMobileRealtimeNotice.jsp	실시간 공지 상세조회 페이지
JSP	/WEB-INF/jsp/egovframework/mbl/com/rns/EgovMobileRealtimeNoticeData.jsp	실시간 공지 server sent event 지원 페이지
JSP	/WEB-INF/jsp/egovframework/mbl/com/rns/EgovMobileRealtimeNoticeMsg.jsp	실시간 공지 메시지 조회 페이지
QUERY XML	resources/egovframework/sqlmap/mbl/com/rns/EgovRealtimeNotice_SQL_XXX.xml	실시간 공지 QUERY XML
Validator Rule XML	resources/egovframework/validator/validator-rules.xml	Validator Rule을 정의한 XML
Validator XML	resources/egovframework/validator/mbl/com/rns/EgovRealtimeNotice.xml	실시간 공지 Validator XML

## 관련테이블

테이블명	테이블명(영문)	비고
실시간 공지	COMTNREALTIMENOTICE	실시간 공지를 관리

## 로그인/회원관리/권한관리

실시간 공지 서비스는 **실시간 공지 목록조회, 상세조회, 등록, 수정, 삭제**에서 관리자 인증이 필요하다. 인증을 위해 로그인/회원관리/권한관리 컴포넌트와 연계하여 사용할 수 있다. Spring Security를 통해 인증을 할 경우 다음과 같이 변경하면 된다.

### src/main/java/resources/egovframework/spring/com/context-egovuserdetailshelper.xml

```

<!-- 인증된 유저의 LoginVO, 권한, 인증 여부를 확인 할 수있는 서비스 클래스-->
<bean id="egovUserDetailsHelper" class="egovframework.com.cmm.util.EgovUserDetailsHelper">
  <property name="egovUserDetailsService">
    <ref bean="egovUserDetailsService" />
  </property>
</bean>

<!-- 시큐리티를 이용한 인증을 사용할 빈 -->
<bean id="egovUserDetailsService" class="egovframework.com.sec.ram.service.impl.EgovUserDetailsServiceImpl"/>

```

## web.xml

```
<!-- 2. 스프링 시큐리티 설정-->
<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<listener>
  <listener-class>org.springframework.security.ui.session.HttpSessionEventPublisher</listener-class>
</listener>
<!-- security 끝 -->

<!-- 4. Spring Security 로그아웃 설정을 위한 필터 설정 -->
<filter>
  <filter-name>EgovSpringSecurityLogoutFilter</filter-name>
  <filter-class>egovframework.com.sec.security.filter.EgovSpringSecurityLogoutFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>EgovSpringSecurityLogoutFilter</filter-name>
  <url-pattern>/uat/uia/actionLogout.do</url-pattern>
</filter-mapping>

<!-- 4. Spring Security 로그인 설정을 위한 필터 설정 -->
<filter>
  <filter-name>EgovSpringSecurityLoginFilter</filter-name>
  <filter-class>egovframework.com.sec.security.filter.EgovSpringSecurityLoginFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>EgovSpringSecurityLoginFilter</filter-name>
  <url-pattern>*.do</url-pattern>
</filter-mapping>
```

web.xml에 위 내용과 같은 Spring Security 관련 필터를 추가 하여야 한다.

로그인/회원관리/권한관리에 대한 자세한 내용은 다음을 참조한다.

- 로그인 : [일반로그인](#)
- 회원관리 : [사용자관리](#)
- 권한관리 : [권한관리](#)

사용자/관리자 권한이 필요한 페이지는 다음과 같다.

권한 종류	페이지 경로	설명
관리자	/mbbl/com/rns/goRealtimeNoticeList.mdo	실시간 공지 목록조회 화면
관리자	/mbbl/com/rns/goRealtimeNotice.mdo	실시간 공지 상세조회 화면
관리자	/mbbl/com/rns/goRealtimeNoticeInsert.mdo	실시간 공지 등록 화면
관리자	/mbbl/com/rns/deleteRealtimeNotice.mdo	실시간 공지 삭제

## 설정 및 주의사항

- \* 현재 배포된 모바일 공통컴포넌트는 HTML5, CSS3를 적용하고 있다.
- \* 그에 따라, 현재 모바일OK, W3C의 모바일 웹 사이트 유효성 검사의 표준에 맞지 않으며, 정확한 유효성 검사에 대한 문의는 해당 전문기관에 문의가 필요하다.
- \* 현재 배포된 모바일 공통컴포넌트 V 2.0은 소스코드에 대한 보안성 검증이 완료 되었다.

## 실시간 공지 모바일 목록 조회

### 비즈니스 규칙

실시간 공지 서비스의 기능 수행은 ajax통신을 통한 json 및 form 전송 방식으로 통해 데이터를 처리하며, 페이지 이동은 전적으로 form 전송 방식으로 이루어지고 있다.

실시간 공지 서비스 대상 메시지 목록을 조회 할 수 있다.  
단 더보기 버튼을 클릭하면 기존의 조회 목록에 신규로 메시지를 5건씩 추가하여 조회 할 수 있다.  
실시간 공지 서비스 등록 화면으로 이동 할 수 있다.

### 관련코드

N/A

## 관련화면 및 수행매뉴얼

Action	URL	Controller method	QueryID
조회페이지로이동	/mbl/com/rns/goRealtimeNoticeList.mdo	goRealtimeNoticeList	
조회	/mbl/com/rns/selectRealtimeNoticeList.mdo	selectRealtimeNoticeList	"RealtimeNoticeDAO.selectRealtimeNoticeList"
더보기	/mbl/com/rns/selectRealtimeNoticeList.mdo	selectRealtimeNoticeList	"RealtimeNoticeDAO.selectRealtimeNoticeList"
글쓰기페이지로이동	/mbl/com/rns/goRealtimeNoticeInsert.mdo	goRealtimeNoticeInsert	
상세페이지이동	/mbl/com/rns/goRealtimeNotice.mdo	goRealtimeNotice	



## 실시간 공지 모바일 등록

### 비즈니스 규칙

실시간 공지 서비스의 기능 수행은 ajax통신을 통한 json 및 form 전송 방식으로 통해 데이터를 처리하며, 페이지 이동은 전적으로 form 전송 방식으로 이루어지고 있다.

게시기간 시작일, 종료일 입력은 필수이며 종료일은 시작일보다 작을 수 없다.  
게시기간 종료일이 현재일보다 작은 경우 등록이 이루어져도 공지메시지로 게시가 되지 않는다.  
실시간 공지 서비스 글을 입력, 유효성 체크 후 저장 처리한다.

### 관련코드

N/A

## 관련화면 및 수행매뉴얼

Action	URL	Controller method	QueryID
등록	/mbl/com/rns/insertRealtimeNotice.mdo	insertRealtimeNotice	"RealtimeNoticeDAO.insertRealtimeNotice"
취소			
목록	/mbl/com/rns/goRealtimeNoticeList.mdo	goRealtimeNoticeList	





## 실시간 공지 모바일 상세 조회

### 비즈니스 규칙

실시간 공지 서비스의 기능 수행은 ajax통신을 통한 json 및 form 전송 방식으로 통해 데이터를 처리하며, 페이지 이동은 전적으로 form 전송 방식으로 이루어지고 있다.

공지메시지 상세를 조회하고 있다.

### 관련코드

N/A

## 관련화면 및 수행매뉴얼

Action	URL	Controller method	QueryID
삭제	/mbl/com/rns/deleteRealtimeNotice.mdo	deleteRealtimeNotice	"RealtimeNoticeDAO.deleteRealtimeNotice"
목록	/mbl/com/rns/goRealtimeNoticeList.mdo	goRealtimeNoticeList	



## 실시간 공지 모바일 메시지 수락확인

### 비즈니스 규칙

실시간 공지 서비스의 기능 수행은 ajax통신을 통한 json 및 form 전송 방식으로 통해 데이터를 처리하며, 페이지 이동은 전적으로 form 전송 방식으로 이루어지고 있다.

수락/거절 시 각각의 EVENT에 따라 알림상태를 **Y**, **N** 으로 처리하여 향후 메시지 수신 여부를 설정한다.

실시간 공지서비스 메시지의 수락확인의 경우 server sent event 지원 브라우저의 경우에는 최초 연결 후 브라우저에 의해 서버페이지 상에 지정된 호출 주기로 메시지를 수신받는다.

미 지원 브라우저의 경우에는 서버측에 별도의 호출 주기를 설정하지 않고 클라이언트 상에 호출 주기를 지정한다. server sent event 지원여부는 관련코드 상에 기술되어 있는 'window.EventSource' 를 통해 확인을 할 수 있다.

또한 실시간 공지 메시지 수락여부를 확인하고 있으며, 수락/거절을 결정하는 순간 실시간 공지 서비스의 **server sent event** 기능이 구동된다.

공지 수락 확인 팝업에서 '수락'을 클릭하면 서버에서 전송하는 메시지를 주기적으로 수신하게 된다. '거절'을 클릭할 경우 서버상에서 주기적으로 전송하는 메시지를 클라이언트상에서 팝업으로 알림을 하지않게된다. 단 하단의 탭의 '수신/수신거부'를 통해 수락/거절시 설정된 메시지 수신상태를 변경할 수 있다.

하단의 관련코드 1,2를 통해 HTML5의 server sent event 적용대 및 미적용 대상, 프레임워크 적용시 처리 방안에대해 확인 할 수 있다. 공지메시지 처리시 주의해서 보아야 할 부분은 '관련코드1'의 'decodeURI(event.data)'와 '관련코드2'의 'URLLEncoder.encode(noticeMsg, "utf-8")' 이다.

URI en/decoding 을 이용하지 않을 경우 한글이 올바르게 처리가 되지않는다. (servlet의 setCharacterEncoding('euc-kr') 이런식의 encoding을 통해서서는 서버상에서는 한글이 처리될 수 있으나 브라우저상에서는 올바로 출력이 되지 않는 것을 확인 하였다.)

## 관련코드1. js

```
function fn_serverSentEvent() {

    // eventURI = "<c:url value='/mbl/com/rns/goEvent.mdo' />";
    eventURI = contextPath + "/mbl/com/rns/goEvent.mdo";
    // 1. Server-sent event 지원
    if (!!window.EventSource) {
        var eventSrc = new EventSource(eventURI);
        eventSrc.addEventListener('message', function(event) {
            fn_showMsg(decodeURI(event.data));
        });

        // 2. Server-sent event 미지원
    } else {
        fn_msgContent(eventURI);
    }
}
```

## 관련코드2. java controller

```
// 1. 실시간 공지 전송 메시지를 조회
RealtimeNotice oRealtimeNotice = (RealtimeNotice)realtimeNoticeService.selectRealtimeNoticeMsg();
if(oRealtimeNotice != null) {
    // 2. 공지 메시지 조합
    // 'data:' 는 server sent event 구현을 위해 필요한 키워드로 변경 하면 실행 불가함..
    String noticeMsg = oRealtimeNotice.getSn() + "|" + oRealtimeNotice.getNoticeSj() + "|" + oRealtimeNotice.getNoticeCn();

    model.addAttribute("data", URLEncoder.encode(noticeMsg, "utf-8"));
    model.addAttribute("retry", "5000");
}

modelAndView.setViewName("/mbl/com/rns/EgovMobileRealtimeNoticeData");
```

## 관련화면 및 수행매뉴얼

Action	URL	Controller method	QueryID
수락확인	/mbl/com/rns/goRealtimeNoticeMsg.mdo	goRealtimeNoticeMsg	





## 실시간 공지 모바일 메시지 확인

### 비즈니스 규칙

실시간 공지 서비스의 기능 수행은 ajax통신을 통한 json 및 form 전송 방식으로 통해 데이터를 처리 하며, 페이지 이동은 전적으로 form 전송 방식으로 이루어지고 있다.

실시간 공지 메시지를 확인 기능을 제공하고 있다.

실시간 공지 메시지를 확인 기능에서 **메시지 확인** 시에는 동일한 메시지는 다시 팝업되지 않으며 **닫기** 를 클릭하면 지정된 시간 이후에 메시지 창이 다시 팝업이된다.

또한 신규 메시지가 등록시에는 기존의 메시지는 더이상 팝업되지 않고 신규 메시지가 팝업되도록 이루어져 있다.

현재 모바일 브라우저 기준으로 **FireFox, Safari** 등이 **HTML5의 server sent event** 를 지원하며

android 기본 탑재 브라우저는 현재(os version 2.3)버전까지 지원을 하고 있지않다.

controller에서 server sent event 지원 여부에 따라 실시간 공지 서비스는 관련 처리 방식을 다르게 하고 있다.

**지원 브라우저** 의 경우에는 해당 controller의 **goEvent** 메서드를 거쳐 메시지 조회 **selectRealtimeNoticeMsg** 를 호출하여 처리한다.

반면 **미지원 브라우저** 는 메시지 조회 **selectRealtimeNoticeMsg** 를 바로 호출하여 메시지 결과를 반환 받는다.

조회 결과를 jsp로 반환하는 과정에서도 **지원 브라우저**의 경우는 db에서 조회된 내용을 '**dtata**', **retry** 라는 결과를 '**EgovMobileRealtimeNoticeData**' 로 결과를 보내 server sent event를 구성한다.

반면 **미지원 브라우저**의 경우에는 db에서 조회된 메시지 내용을 호출한 페이지로 반환한다.

페이지 하단의 **수신, 수신거부 Tab** 을 선택하면 실시간 공지 메시지 페이지 초기의 **수락/거절** 과 마찬가지로 메시지 수신여부가 재설정 가능하다.

메시지 확인 버튼을 클릭하면 최종적으로 신규 메시지가 등록되기 전까지는 별도의 메시지 알림(팝업)은 없다.

### 관련코드

```
// 메시지 조회
function fn_msgContent(eventURI) {
    $.ajax({
        type      : "POST",
        cache     : false,
        url       : eventURI,
        dataType  : "text",
        success   :function(textData) {
            var srcData = decodeURI(textData).split("\n");
            var retryTime = 5000;
            if (srcData.length == 2) {
                retryTime = srcData[0].split(":")[1];
                var data = srcData[1].split(":")[1];
                fn_showMsg(data);
            }
            setTimeout("fn_serverSentEvent()", retryTime);
        }
    });
}

// 메시지 화면 호출
function fn_showMsg(srcData) {
    var data = srcData.split("|");
    data[1] = data[1].replace(/\+/g, ' ');
    data[2] = data[2].replace(/\+/g, ' ');

    if (data.length == 3 && notiStatus == "Y" && data[0] > maxSn) {
        var alertMessage = "<dl class='mcom-msg'><dt><span class='left'>제목</span><span class='right'>" + data[1] +
            "</span></dt>" + "<dd><span class='left'>내용</span><span class='right'>" + data[2] + "</span></dd><dd></dd></dl>";
        var btmItem = [ {id : 'button1', value: "메시지 확인"}];

        jActionSheet(alertMessage, '메시지 확인', 'b', btmItem , function(r) {
            if(r == "메시지 확인") {
                maxSn = data[0];
            }
        });
    }
}
```

## 관련화면 및 수행매뉴얼

Action	URL	Controller method	QueryID
server sent event 처리	/mbl/com/rns/goEvent.mdo	goEvent	
공지메시지조회	/mbl/com/rns/selectRealtimeNoticeMsg.mdo	selectRealtimeNoticeMsg	RealtimeNoticeDAO.selectRealtimeNoticeMsg



## 참고자료

- 실행환경 참조 : [ID Generation Service](#)
- 배포 및 테스트 관련 : [모바일 공통컴포넌트란](#) 및 [배포 패키지 구성안](#)
- 관련 데이터베이스 구조 및 스키마 : [관련 데이터베이스 구조 및 스키마](#)