Cooperation - Real-time notification service

Summary

In the real-time notification service, the server transfers and exposes notifications regularly to user’s browsers that supports HTML5 and the manager manages the notifications.

In this component, the server sends the PUSH alarm regularly in the designated time.

The real-time notification service is realized through server sent event of the HTML5.

The browsers that supports the server sent event were distinguished from the ones that does not support it and they were classified as the subject of receiving and refusing the notification message on user’s browsers.

\* Function Flow

The development and testing of the mobile screen of this component is completed. It is also noted in development since the supports on HTML5 and CSS3 vary between the PC browser and mobile browser in the same brand.

The tested device and browsers are GalaxyS2, GalaxyS, iPad2, iPhone4, GalaxyTab1 and the web browsers, Firefox, Safari, Opera Mobile.

The test results are as following.

Android

The operation of HTML5, CSS3 and components of the Firefox(6.xx) and opera mobile was stable in android OS 2.3 (gengerbread).

However, in the lower version browsers, android 2.1, 2.2, the operation of components was unstable and UI realization was not smooth due to insufficient HTML5 and CSS3 supports.

For Opera Mobile, it caused problems in colors and event processing in UI. Therefore, it was excluded from the subject. Galaxy, GalxyS2 and GalaxyTab1 were chosen as test mobile devices.

iOS

The iOS is one of three supported browsers and was tested in versions 4.2 and 4.3. The support for HTML5 and CSS3 were relatively smooth.

iPhone4 and iPad2 were chosen as test mobile devices.

Description

The real-time notification service is composed of management functions – real-time notice list inquiry, detailed inquiry, register and delete – and user functions – accept or refuse real-time notification service, check messages and receive or refuse messages. (Refer to Related Functions)

Class Diagram



Related sources

Related Tables

|  |  |  |
| --- | --- | --- |
| Table name | Table name (English ) | Notes |
| Real-time notification | C OMTNREALTIMENOTIC E | Manage real-time notification |

Login / membership management / permission management

Real-time notification service requires the administrator authentication in real-time notification list inquiry, detail inquiry, registration, modification and deletion.

For authentication, you can connect with login, membership management and permission management components. If you use Spring Security for authentication, make change as in the followings.

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

<!--Service class that can check the authenticated user’s LoginVO, permissions and authentication -->

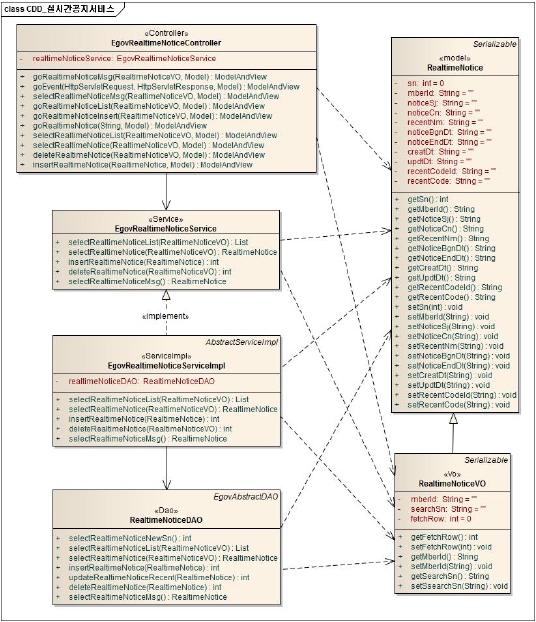
<bean id="egovUserDetailsHelper" class="egovframework.com.cmm.util.EgovUserDetailsHelper">

<property name="egovUserDetailsService">

<ref bean="egovUserDetailsSecurityService" />

</property>

</bean>



|  |  |  |
| --- | --- | --- |
| Type | Target source names | Note |
| C ontroller | egovframework.mbl.com.rns.web.EgovRealtimeNoticeC ontroller.java | Real-time notification C ontroller C lass |
| Service | egovframework.mbl.com.rns.service.EgovRealtimeNoticeService.java | Real-time notification Service C lass |
| ServiceImpl | egovframework.mbl.com.rns.service.impl.EgovRealtimeNoticeServiceImpl.java | Real-time notification ServiceImpl C lass |
| Model | egovframework.mbl.com.rns.service.RealtimeNotice.java | Real-time notification Model C lass |
| VO | egovframework.mbl.com.rns.service.RealtimeNoticeVO.java | Real-time notification VO C lass |
| DAO | egovframework.mbl.com.rns.service.impl.RealtimeNoticeDAO.java | Real-time notification Dao C lass |
| JSP | /WEB-INF/jsp/egovframework/mbl/com/rns/EgovMobileRealtimeNoticeList.jsp | Real-time notification List inquiry page |
| JSP | /WEB-INF/jsp/egovframework/mbl/com/rns/EgovMobileRealtimeNoticeInsert.jsp | Real-time notification Registration page |
| JSP | /WEB-INF/jsp/egovframework/mbl/com/rns/EgovMobileRealtimeNotice.jsp | Real-time notification Detail inquiry page |
| JSP | /WEB-INF/jsp/egovframework/mbl/com/rns/EgovMobileRealtimeNoticeData.jsp | Real-time notification server sent event support page |
| JSP | /WEB-INF/jsp/egovframework/mbl/com/rns/EgovMobileRealtimeNoticeMsg.jsp | Real-time notification message inquiry page |
| QUERY XML | resources/egovframework/sqlmap/mbl/com/rns/EgovRealtimeNotice\_SQL\_XXX.xml | Real-time notification QUERY XML |
| Validator Rule XML | resources/egovframework/validator/validator-rules.xml | XML that defines Validator Rule |
| Validator XML | resources/egovframework/validator/mbl/com/rns/EgovRealtimeNotice.xml | Real-time notification Validator XML |

<!-- Bean that will use Security-based authentication -->

<bean id="egovUserDetailsSecurityService" class="egovframework.com.sec.ram.service.impl.EgovUserDetailsSecurityServiceImpl"/>

web.xml

<!-- 2. Spring Security Setup -->

<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 end -->

<!-- 4. Filter setup for Spring Security logout setup -->

<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. Filter setup for Spring Security login setup -->

<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>

As in the above, add Spring Security filter to web.xml.

For details on login/membership management/permission management, refer to the followings.

Login: general login

Membership management: user management

Permission management: permission management

The following pages need the user/administrator permissions.

Settings and Notes

\* The currently distributed mobile common component adopts HTML5 and CSS3 and this means that it doesn’t support the standards for mobile web site validity check of Mobile OK and W3C. For details on correct validity check, please contact the competent professional institute..

\* Security verification is under progress on the currently distributed mobile component. We plan to complete security verification as soon as possible to release a new security-patched version.

Real-time notification mobile list inquiry

Business rules

Real-time notification service uses ajax communication based json and form transfer types to process data.

Page navigation is totally based on the form type.

You can retrieve Real-time notification service message lists.

By clicking View More button, you can add another five messages to a retrieved list.

You can move to Real-time notification service registration screen.

Related codes

N/A

Screen and execution manual



|  |  |  |  |
| --- | --- | --- | --- |
| Action | URL | Controller method | QueryID |
| Move to inquiry page | /mbl/com/rns/goRealtimeNoticeList.mdo | goRealtimeNoticeList |  |
| Inquiry | /mbl/com/rns/selectRealtimeNoticeList.mdo | selectRealtimeNoticeList | “RealtimeNoticeDAO.selectRealtimeNoticeList” |
| View more | /mbl/com/rns/selectRealtimeNoticeList.mdo | selectRealtimeNoticeList | “RealtimeNoticeDAO.selectRealtimeNoticeList” |
| Move to Write page | /mbl/com/rns/goRealtimeNoticeInsert.mdo | goRealtimeNoticeInsert |  |
| Move to Detail page | /mbl/com/rns/goRealtimeNotice.mdo | goRealtimeNotice |  |

|  |  |  |
| --- | --- | --- |
| Permission type | Page path | Description |
| administrator | /mbl/com/rns/goRealtimeNoticeList.mdo | real-time notification list inquiry screen |
| administrator 리자 | /mbl/com/rns/goRealtimeNotice.mdo | real-time notification detail inquiry screen |
| administrator | /mbl/com/rns/goRealtimeNoticeInsert.mdo | real-time notification registration screen |
| administrator | /mbl/com/rns/deleteRealtimeNotice.mdo | real-time notification deletion |

Real-time notification mobile Registration

Business Rules

Real-time notification service uses ajax communication based json and form transfer types to process data.

Page navigation is totally based on the form type.

The starting and ending dates of publishing duration are required. The ending date should not be earlier than the starting date.

If the ending date of the publishing duration is earlier than the starting date, the notification will be published though registered.

Real-time notifications are entered and stored after validity check.

Related codes

N/A

Screen and execution manual



|  |  |  |  |
| --- | --- | --- | --- |
| Action | URL | Controller method | QueryID |
| Register | /mbl/com/rns/insertRealtimeNotice.mdo | insertRealtimeNotice | “RealtimeNoticeDAO.insertRealtimeNotice” |
| Cancel |  |  |  |
| List | /mbl/com/rns/goRealtimeNoticeList.mdo | goRealtimeNoticeList |  |

Real-time notification mobile detail inquiry

Business Rules

Real-time notification service uses ajax communication based json and form transfer types to process data.

Page navigation is totally based on the form type.

Real-time message details are being inquired.

Related codes

N/A

Screen and execution manual



|  |  |  |  |
| --- | --- | --- | --- |
| Action | URL | Controller method | QueryID |
| Delete | /mbl/com/rns/deleteRealtimeNotice.mdo | deleteRealtimeNotice | “RealtimeNoticeDAO.deleteRealtimeNotice” |
| List | /mbl/com/rns/goRealtimeNoticeList.mdo | goRealtimeNoticeList |  |

Real-time notification mobile message acceptance confirmation

Business Rules

Real-time notification service uses ajax communication based json and form transfer types to process data. Page navigation is totally based on the form type.

Set Y to acceptance and N to refusal for message reception.

In Real-time notification service message acceptance confirmation, for browsers that support server sent event, messages are received according to the call cycle designated on the server page by the browser.

For a non-supporting browser, designate the call cycle on the client, not on the server side.

Whether server sent event is supported can be check from 'window.EventSource' specified in Related codes.

The acceptance of the real-time notification message is being checked.

As soon as whether to accept or refuse is determined, Real-time notification service’s server sent event function is activated.

If you click “Accept” on the notification acceptance confirmation popup, you will receive messages from the server regularly.

If you click “Refuse”, messages sent regularly from the server will not be notified on the client.

You can use the “Receive/Refuse to Receive’ in the tab at the bottom to change the message receiving setting.

You can use related codes 1 and 2 at the bottom to check what subjects are for or not for HTML5’s server sent event and measures to take when the framework is applied.

In processing notification messages, you have to pay attentions to Related codes1’s decodeURI(event.data)' and Related codes2’s 'URLEncoder.encode(noticeMsg, “utf-8”)'.

If URI en/decoding is not used, Korean is not processed correctly.

It has been confirmed that encoding like servlet’s setCharacterEncoding('euc-kr') can process Korean on the sever but cannot on the browser correctly.

Related codes1. js

function fn\_serverSentEvent() {

// eventURI = "<c:url value='/mbl/com/rns/goEvent.mdo'/>";

eventURI = contextPath + "/mbl/com/rns/goEvent.mdo";

// 1. Server-sent event supported

if (!!window.EventSource) {

var eventSrc = new EventSource(eventURI);

eventSrc.addEventListener('message', function(event) {

fn\_showMsg(decodeURI(event.data));

});

// 2. Server-sent event not supported

} else {

fn\_msgContent(eventURI);

}

}

Related codes2. java controller

// 1. Inquire the real-time notification transfer message

RealtimeNotice oRealtimeNotice = (RealtimeNotice)realtimeNoticeService.selectRealtimeNoticeMsg();

if(oRealtimeNotice != null) {

// 2. Notification message combination

// 'data:' will be non-executable if it is changed to a key word required for 는 server sent event implementation..

String noticeMsg = oRealtimeNotice.getSn() + "|" + oRealtimeNotice.getNoticeSj() + "|" + oRealtimeNotice.getNoticeCn();

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

model.addAttribute("retry", "5000");

}

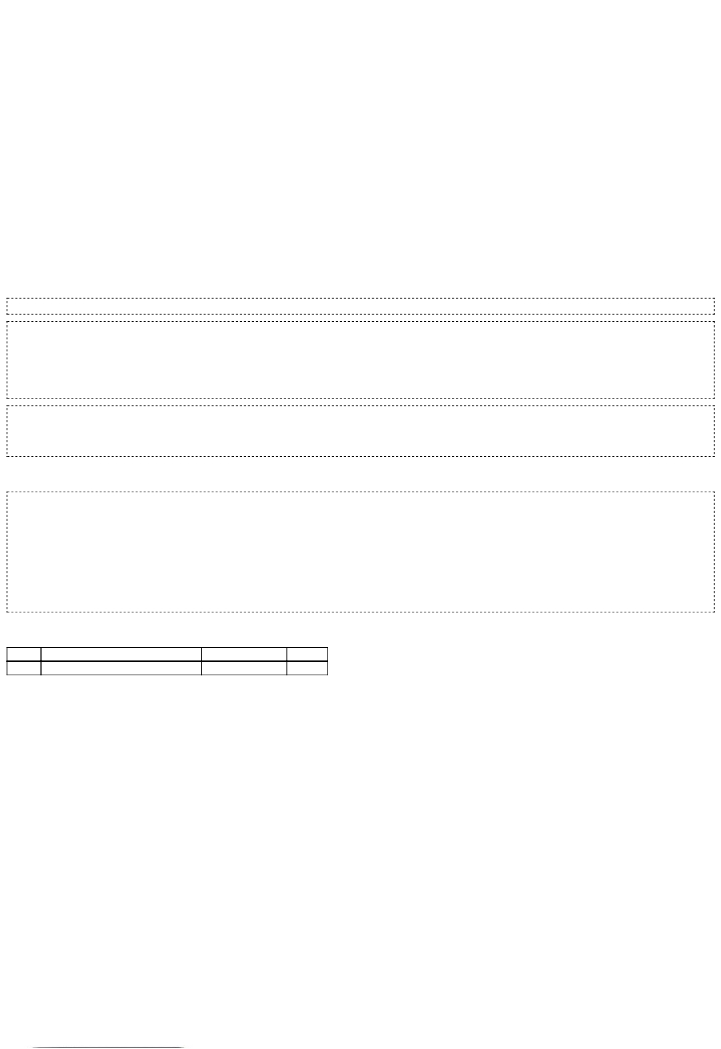
modelAndView.setViewName("/mbl/com/rns/EgovMobileRealtimeNoticeData");

Screen and execution manual

Action URL

Controller method QueryID

Acceptance confirmation /mbl/com/rns/goRealtimeNoticeMsg.mdo goRealtimeNoticeMsg



Real-time notification mobile message confirmation

Business Rules

Real-time notification service uses ajax communication based json and form transfer types to process data.

Page navigation is totally based on the form type.

Real-time notification messages are being checked.

Real-time notification is checked. After checking, the same message will not pop up again. If you click “Close”, the message pops up again in a certain period time.

If a new message is registered, this message pops up and the existing message doesn’t pop up any more.

Currently, among the mobile browsers, FireFox and Safari support HTML5 server sent events.

The standard Android browser doesn’t support the current version (OS version 2.3).

Depending on the controller’s support for server sent events, Real-time notification service provide different processes.

The supporting browsers use the controller’s goEvent method to call the message inquiry selectRealtimeNoticeMsg.

The non-supporting browsers call message inquiry selectRealtimeNoticeMsg directly to receive the message results.

In returning the inquiry result to jsp, the supporting browsers send details retrieved from DB to 'dtata' and ‘retry’ result to 'EgovMobileRealtimeNoticeData'.

The non-supporting browsers return messages retrieved from DB to the called page.

By selecting the Receive/Refuse to Receive in the tab at the bottom, you can re-define whether to receive messages as in Real-time notification message page’s initial acceptance/refusal.

If you click the Confirm button, you will not have a message (pop up) until a new message is registered.

Related codes

//Message inquiry

function fn\_msgContent(eventURI) {

$.ajax({

type

cache

url

dataType

: "POST",

: false,

: eventURI,

: "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);

}

});

}

// 메 시 지 화 면 표 출

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'>제 목</span><span class='right'>" + data[1] + "</span></dt>" + "<dd><span class='left'>내 용</span><span clas

var btmItem = [ {id : 'button1', value: "메 시 지 확 인"}];



jActionSheet(alertMessage, '메 시 지 확 인', 'b', btmItem , function(r) {

if(r == "메 시 지 확 인") {

maxSn = data[0];

}

});

}

}



jActionSheet(alertMessage, 'Check message', 'b', btmItem , function(r) {

if(r == "check message") {

maxSn = data[0];

}

});

}

}

Screen and execution manual

References

Refer to execution environment: ID Generation Service

Distribution and test: Mobile Common Component System and distribution package plan

Related database structure and schema : Related database structure and schema

e govfram e work /m com /realtime notification service.txt · Last m odifie d: 2011/10/10 16:07 by m com

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported

[http://creativecommons.org/licenses/by-nc-sa/3.0/]



|  |  |  |  |
| --- | --- | --- | --- |
| Action | URL | Controller method | QueryID |
| server sent event processing | /mbl/com/rns/goEvent.mdo | goEvent |  |
| Notification message inquiry | /mbl/com/rns/selectRealtimeNoticeMsg.mdo | selectRealtimeNoticeMsg | RealtimeNoticeDAO.selectRealtimeNoticeMsg |