



Software House SPECIALIST IN SAGE AND NOTES/DOMINO DEVELOPMENT

Using HTML5 Notification API in XPages

September 15, 2014

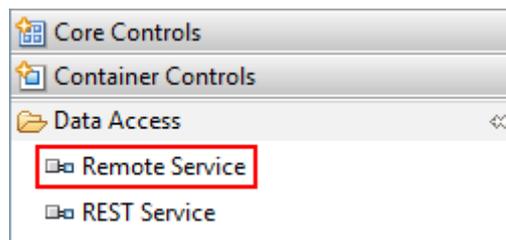
HTML5 introduces [Web Notifications API](#) which "allows alerting the user outside the context of a web page of an occurrence, such as the delivery of email". This opens up quite a few possibilities for developers like alerting a user when a document needs approval irrespective of the XPage he is currently on.

In this example I will use the Extension Library control Remote Service to poll a view to check if any documents require approval and if yes then show notification to user.

REMOTE SERVICE CONTROL

The Remote Service control [allows calling SSJS code from client side JavaScript code without a full page refresh](#). Instead the output of the SSJS code is returned asynchronously. Tim Tripcony gave a beautiful explanation [here](#) on how this control should be used and its usage scenarios.

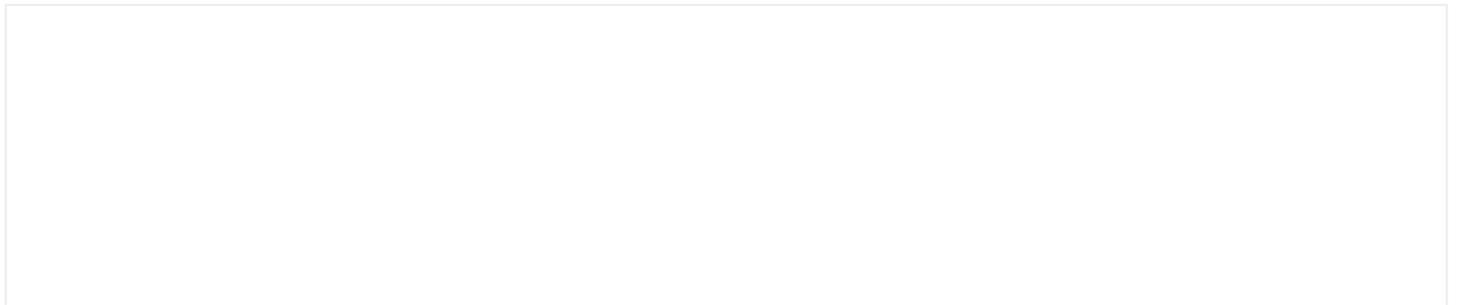
You can find the Remote Service control under Data Access group in your Control view. Drag it on to your XPage.



Set the properties for Remote Service as displayed below.

Property	Value
binding	
id	jsonRpcService1
loaded	
methods	
remoteMethod [0]	
arguments	
loaded	
name	checkApprovals
script	var currentDb=NobisDatabase = session.getCu...
pathInfo	
rendered	
renderType	
serviceName	service
state	
styling	
disableTheme	
themId	

Notice the value of property *serviceName*. We will be using this property in our client side JavaScript code call the remote method. In the *script* property of remote method we will write the SSJS script which will be checking if there is any document with Status as "Submitted" and return the count of those documents. So our code for remote service would be:



Now we call our remote service at regular intervals and notify the user if there are any documents to approve. Here I am checking every 5 seconds (which would not be suitable for production).

```
var intervalTime = 5000; // Check for new documents pending approval every 5 seconds

setInterval(function() {
    var deferred = service.checkApprovals();
    deferred.addCallback(function(result) {
        if (result > 0) {
            notifyUser(result);
        } else {
            console.log("No documents pending approval.");
        }
    });
}, intervalTime);
```

WEB NOTIFICATION API

This [article on Mozilla Developer Network](#) describes how to use the notification API. First we check if the browser supports notifications API by this statement.

```
if (!("Notification" in window)) { ... }
```

If the browser supports it then we ask for permission from user to show notifications.

```
Notification.requestPermission(function (permission) {
    if (permission === "granted") {
        ...
    }
});
```

So our function *notifyUser* would come to something like this:

```
function notifyUser(result) {
    if (!("Notification" in window)) { // If browser does not support notification
        console.log("Desktop notifications not supported in this browser");
    } else if (Notification.permission === "granted") { // If browser supports notification and user has granted
        createNotification(result);
    } else if (Notification.permission !== "denied") { // If browser supports notification but user is yet to gr
        Notification.requestPermission(function (permission) {
            if (permission === "granted") {
                createNotification(result);
            }
        });
    }
}
```

To create notification we call the following statement:

```
var notification = new Notification(TITLE, BODY);
```

So our *createNotification* would be this:

```
function createNotification(result) {
    title = "New documents for approval"
    options = {
        body: result + " document(s) require your approval.",
        icon: "notificationicon.png"
    };
    var notification = new Notification(title, options);
}
```

We store the image *notificationicon.png*, shown in above code snippet, in our database itself as file (or image) resource. Now when you open your XPage you would get notifications in browser in case you have documents pending for approval.



Notifications as shown in Google Chrome and Firefox respectively

The above demo only works in latest versions of Google Chrome and Mozilla Firefox.

PUTTING EVERYTHING TOGETHER

We put the entire code together and our XPage looks like this.

```
This page will show notification every 5
seconds if there is at least one document in the vApprovalForm view
with Status as "Submitted".
```

```
    0) {
        notifyUser(result);
    } else {
        console.log("No documents pending approval.");
    }
});
}, intervalTime)

function notifyUser(result) {
    if (!("Notification" in window)) { // If browser does not support notification
        console.log("Desktop notifications not supported in this browser");
    } else if (Notification.permission === "granted") { // If browser supports notification and user has granted
        createNotification(result);
    } else if (Notification.permission !== "denied") { // If browser supports notification but user is yet to gr
        Notification.requestPermission(function (permission) {
            if (permission === "granted") {
                createNotification(result);
            }
        });
    }
}

function createNotification(result) {
    title = "New documents for approval"
    options = {
        body: result + " document(s) require your approval.",
        icon: "notificationicon.png"
    };
};
```

```
var notification = new Notification(title, options);  
}]]>
```

You can also download the demo database and try it out: [Notification Demo](#).

Share:



+ MORE

Like 0



Share



Pin

September 15, 2014 [<http://www.pipalia.co.uk/using-html5-notification-api-xpages/>] | by Naveen Maurya | in [html 5](#), [xpages](#) .

2 Comments

2 thoughts on “Using HTML5 Notification API in XPages”



Stephan Wissel

September 16, 2014 at 2:41 am

Nice one! In a live application you could modify the code to show a footer in the regular app instead. That would make it degrade gracefully



Naveen Maurya

Post author

September 17, 2014 at 4:08 pm

Thanks Stephen!
