Receiving system logs info in Symbian Web Runtime

This code snippet shows how to retrieve information from system logs in Symbian Web Runtime, using the Logging Platform Service (introduced in S60 5th Edition).

Overview

The `device.getServiceObject("Service.Logging", "IDataSource")` method is used to obtain access to the service object for the Contact Service API.

After setting the values for `criteria.Type`, the `IDataSource.GetList(criteria)` method is used to retrieve items from system logs.

Source file: LoggingView.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <script type='text/javascript' src="LoggingView.js"></script>
        <title></title>
    </head>
    <body onload="init()">
        <div id="inputdata">
            <select id="eventtype">
                <option>EKLogCallEventType</option>
                <option>EKLogDataEventType</option>
                <option>EKLogFaxEventType</option>
                <option>EKLogShortMessageEventType</option>
                <option>EKLogPacketDataEventType</option>
            </select>
            <button onclick="doRefresh()">Refresh</button><br/>
        </div>
        <hr/>
        <!-- table for showing existing items -->
        <table id="maintable" cellspacing='1' cellpadding='1' border='1'>
        <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
        </table>
    </body>
</html>
```

Source file: LoggingView.js

```
// logging system service object
var logger;
// system error code
var noItemsErrorCode = 1012;

/**
 * Initializes the widget
 */
function init() {
    // Obtain the service object
    try {
        logger = device.getServiceObject("Service.Logging", "IDataSource");
    } catch(err) {
       alert( "error receiving Logging service handle" );
       return;
    }

    // Refreshing info
    doRefresh();
}

/**
 * Refreshes information in table
 */
function doRefresh() {
    var criteria = new Object();
    criteria.Type = "Log";
    criteria.Filter = new Object();
    criteria.Filter.EventType =
        document.getElementById('eventtype').selectedIndex;

    // making a call to service to receive a list of suitable items
    var logs;
    try {
        logs = logger.IDataSource.GetList(criteria);
    }catch(err) {
        alert("error receiving log items");
        return;
    }

    var table = document.getElementById('maintable');
    // cleaning table and adding header
    while(table.rows[0] != undefined) {
        table.deleteRow(0);
    }
    table.insertRow(0);
    table.rows[0].innerHTML = "<th>EventType</th><th>EventDuration</th>" +
        "<th>Subject</th><th>Description</th>";

    if(logs.ErrorCode != 0) {
```


(C) Copyright Nokia 2013. All rights reserved.

```
        // If the error code is not "no suitable items", show an alert.
        // Otherwise just leave the function.
        if(logs.ErrorCode != noItemsErrorCode) {
            alert( logs.ErrorMessage );
        }
        return;
    } else {
        // filling table with received data
        var table = document.getElementById('maintable');
        var i;  // iterator
        var item; // pointer to item in received list
        for(i = 0; (item = logs.ReturnValue.getNext()) != undefined; ++i) {
            table.insertRow(i + 1);
            var result = "<td>";
            if(item['EventType'] != undefined) {
                result += item['EventType'];
            }
            result += "</td><td>";
            if(item['EventDuration'] != undefined) {
                result += item['EventDuration']
            }
            result += "</td><td>";
            if(item['Subject'] != undefined) {
                result += item['Subject'];
            }
            result += "</td><td>";
            if(item['Description'] != undefined) {
                result += item['Description'];
            }
            result += "</td>";
            table.rows[i + 1].innerHTML = result;
        }
    }
}
```

Postconditions

- After loading a table with all logs is shown.
- Pressing the "Refresh" button refreshes the table.
- The drop-down list allows you to choose what type of logs you would like to see.

Supplementary material

You can view the source file and executable application in the attached ZIP archive. The archive is available for download at Media:Receiving system logs info in WRT.zip.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.