Exporting and importing calendar events in Symbian Web Runtime

This code example demonstrates how to export and import calendar entries in Symbian Web Runtime, using the Calendar Platform Service (introduced in S60 5th Edition).

Overview

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

The "Export" button handler uses the `IDataSource.Export(criteria, callback)` method from the Calendar Service API to export the selected calendar entries to the file.

The "Import" button handler uses the `IDataSource.Import(criteria, callback)` method from the Calendar Service API to import calendar events from the file.

The methods used for exporting and importing calendar entries are asynchronous.

To interrupt the execution of the asynchronous exporting or importing process, the method `IDataSource.Cancel(criteria)` is used. Its `criteria` parameter specifies the transaction ID of the asynchronous method to be canceled.

Source: Relevant HTML components

```
<input type="radio" name="eventType" id="meeting" checked
    onclick="showEntries();" />
<label for="meeting">Meetings</label><br />

<input type="radio" name="eventType" id="anniversary"
    onclick="showEntries();" />
<label for="anniversary">Anniversaries</label><br />

<input type="radio" name="eventType" id="dayEvent"
    onclick="showEntries();" />
<label for="dayEvent">Day Events</label><br />

<input type="radio" name="eventType" id="reminder"
    onclick="showEntries();" />
<label for="reminder">Reminders</label><br />

<input type="radio" name="eventType" id="toDo"
    onclick="showEntries();" />
<label for="toDo">ToDos</label><br />

<label for="entriesList">Entries:</label><br />
<select size="2" id="entriesList"></select><br />

<label for="fileSelection">File:</label><br />
<p id="fileSelection">
    C:\Data\Others\<input type="text" size="20" id="file" />
</p>

<p>File Type:</p>
<input type="radio" name="fileType" id="iCal" checked />
<label for="iCal">ICal</label><br />
<input type="radio" name="fileType" id="vCal">
<label for="vCal">VCal</label><br />

<input type="button" id="export" onclick="exportEntry();" value="Export">
<input type="button" id="import" onclick="importEntry();" value="Import">
<input type="button" id="cancel" onclick="cancelAction();" value="Cancel">
<br />
<p id="state"></p><br />
```

Source: JavaScript file

```
var serviceObj = null;

// Contains current export or import transaction id value
var currentTransactionId = 0;

window.onload = init;

// Initializes the widget
function init() {
    // Obtain the service object
    try {
        serviceObj = device.getServiceObject("Service.Calendar",
            "IDataSource");
    } catch (ex) {
        alert("Service object cannot be found.");
        return;
    }
}

function showEntries() {
    // Showing entries is omitted here for brevity. Refer to the See also
    // section of the code snippet for more information.
    // ...
```


(C) Copyright Nokia 2013. All rights reserved.

```
}

/**
 * Imports all entries from the specified file to the calendar.
 */
function importEntry() {
    var criteria = new Object();
    criteria.Type = "CalendarEntry";
    criteria.Data = new Object();

    // Name of the file from which calendar events will be imported
    var fileName = "C:\\Data\\Others" + document.getElementById("file").value;
    criteria.Data.FileName = fileName;
    // Get format of the file.
    if (document.getElementById("iCal").checked) {
        criteria.Data.Format = "ICal";
    } else {
        criteria.Data.Format = "VCal";
    }

    // Import the data
    try {
        var result = serviceObj.IDataSource.Import(criteria, importCallback);
        // Store current transaction id to global variable
        currentTransactionId = result.TransactionID;
        document.getElementById("state").innerHTML = "Importing...";
    } catch (ex) {
        alert("Error in importing entries: " + ex);
    }
}

/**
 * Callback function for import asynchronous call.
 * @param transId A number representing the transaction that called the
 * callback handler.
 * @param eventCode A number representing the callback return status.
 * @param result An object for holding the callback return value
 */
function importCallback(transId, eventCode, result) {
    // Check result of importing
    if (result.ErrorCode != 0) {
        alert("Error in importing: " + result.ErrorMessage);
    } else {
        alert("Data was imported successfully!");
    }

    document.getElementById("state").innerHTML = "";
    showEntries();
}

/**
 * Exports the selected entries from calendar to file.
 */
function exportEntry() {
    // Get the name of the selected entry
    var entriesList = document.getElementById("entriesList");
    var entryList = new Array();
    for (var i = 0; i < entriesList.length; i++) {
        if (entriesList.options[i].selected) {
            entryList[i] = entriesList.options[i].value;
            break;
        }
    }

    // Criteria indicating the data to be exported and the destination file
    var criteria = new Object();
    criteria.Type = "CalendarEntry";
    criteria.Data = new Object();
    criteria.Data.LocalIdList = entryList;
    // Destination file name
    var fileLocalName = document.getElementById("file").value;
    // Check the file name correctness
    var regPattern = new RegExp("^[a-zA-Z]+\.?[a-zA-Z]+$");
    if (!(regPattern.test(fileLocalName))) {
        alert("Incorrect file name: " + fileLocalName);
        return;
    }
    // Create full destination file name
    var fileName = "C:\\Data\\Others\\" + fileLocalName;
    criteria.Data.FileName = fileName;
    // Set the format of the destination file.
    if (document.getElementById("iCal").checked) {
        criteria.Data.Format = "ICal";
    } else {
        criteria.Data.Format = "VCal";
    }

    try {
        // Export the selected entries
        alert("Export begins");
        var result = serviceObj.IDataSource.Export(criteria, exportCallback);
        currentTransactionId = result.TransactionID;
        document.getElementById("state").innerHTML = "Exporting...";
    } catch (ex) {
        alert ("Error in exporting entries: " + ex);
    }
}

/**
 * Callback function for export asynchronous call.
 * @param transId A number representing the transaction that called the
 * callback handler.
 * @param eventCode A number representing the callback return status.
 * @param result An object for holding the callback return value
 */
function exportCallback(transId, eventCode, result) {
    // Check result of exporting
    if (result.ErrorCode != 0) {
        alert("Error in exporting: " + result.ErrorMessage);
    } else {
        alert("Data was exported successfully!");
    }

    document.getElementById("state").innerHTML = "";
    showEntries();
}

/**
 * Cancels the current action (exporting or importing).
 */
```

 (C) Copyright Nokia 2013. All rights reserved.

```
function cancelAction() {
    // Criteria determining the transaction which is to be canceled
    var criteria = new Object();
    // Get transaction id for the transaction which is to be canceled
    var transactionId = currentTransactionId;
    if (transactionId == undefined) {
        return;
    }
    criteria.TransactionID = transactionId;
    // Cancel the current action
    var result = calendarServiceObject.Cancel(criteria);
    if (result.ErrorCode !== 0) {
        alert("Cancel error: " + result.ErrorMessage);
    }
    showEntries();
}
```

Postconditions

The selected calendar entries are exported into or imported from the file specified by the user.

Supplementary material

You can view the source file and executable application in the attached ZIP archive. The archive is available for download at Media:Listing calendars and events in WRT.zip.

See also

Listing calendars in Symbian Web Runtime

 (C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.