Managing phonebook groups in Symbian Web Runtime

This code snippet demonstrates how to add and remove phonebook groups using the Contact Platform Service for Web Runtime. The service was introduced in S60 5th Edition.

Source: Relevant HTML components

```
<div id="bodyContent" class="bodyContent">
    <label for="groupName">Group name:</label>
    <input id="groupName" type="text"></input><br />
    <input type="button" onclick="addGroup();" value="Add" /><br />
    <input type="button" onclick="removeGroup();" value="Remove" />

    <hr />
    <!-- Table for showing existing items -->
    <table id="maintable" cellspacing="1" cellpadding="1" border="1">
        <tr><td>1</td></tr>
    </table>
</div>
```

Source file: JavaScript file

```
var phonebook = null;

var noItemsErrorCode = 1012;

window.onload = init;

// Initializes the widget
function init() {
    // ...

    // Initialize the service object
    try {
        phonebook = device.getServiceObject("Service.Contact", "IDataSource");
    } catch (ex) {
        alert("Error while initializing Contact service object");
        return;
    }

    // Refresh the information in the table
    doShow();
}

/**
 * Refreshes the information in the table
 */
function doShow() {
    // Criteria object for making a GetList call to receive groups information
    var criteria = new Object();
    criteria.Type = "Group";
    var groups = null;
    // Get list of groups
```


(C) Copyright Nokia 2014. All rights reserved.

```
    try {
        groups = phonebook.IDataSource.GetList(criteria);
    } catch (ex) {
        alert("Error in retrieving group list");
        return;
    }

    var table = document.getElementById("maintable");
    // Clean the table and add a header
    while (table.rows[0] != undefined) {
        table.deleteRow(0);
    }
    table.insertRow(0);
    table.rows[0].innerHTML = "<th>Groups:</th>";

    if (groups.ErrorCode != 0) {
        // If the error code is not "no suitable items", show an alert;
        // Otherwise just leave the function
        if (groups.ErrorCode != noItemsErrorCode) {
            alert(groups.ErrorMessage);
        }
        return;
    } else {
        // Fill the table with the received data
        var item;
        for (var i = 0; (item = groups.ReturnValue.getNext()) != undefined;
            i++) {
            table.insertRow(i + 1);
            var result = "<td>";
            if (item.GroupLabel != undefined) {
                result += item.GroupLabel;
            }
            result += "</td>";
            table.rows[i + 1].innerHTML = result;
        }
    }
}

/**
 * Adds the specified item to groups
 */
function addGroup() {
    var groupName = document.getElementById("groupName").value;
    if (groupName == "") {
        alert("You must enter a name for the group.");
        return;
    }

    // Check that duplicate groups are not added
    // Criteria object for making a GetList call to receive groups information
    var criteria = new Object();
    criteria.Type = "Group";
    var groups = null;
    // Get list of groups
    try {
        groups = phonebook.IDataSource.GetList(criteria);
    } catch (ex) {
```


(C) Copyright Nokia 2014. All rights reserved.

```
        alert("Error in retrieving group list");
        return;
    }

    if (groups.ErrorCode != 0) {
        // If the error code is not "no suitable items", show an alert;
        // Otherwise just leave the function
        if (groups.ErrorCode != noItemsErrorCode) {
            alert(groups.ErrorMessage);
        }
        return;
    } else {
        // Fill the table with the received data
        var item;
        while ((item = groups.ReturnValue.getNext()) != undefined) {
            if (item.GroupLabel != undefined) {
                if (item.GroupLabel == groupName) {
                    alert("Group with this name already exists.");
                    return;
                }
            }
        }
    }

    criteria = new Object();
    criteria.Type = "Group";
    criteria.Data = new Object();
    // Adding the groupName parameter to the new item
    criteria.Data.GroupLabel = document.getElementById('groupName').value;

    groups = null;
    try {
        groups = phonebook.IDataSource.Add(criteria);
    } catch(err) {
        alert("Error in adding the group.");
        return;
    }
    if (groups.ErrorCode != 0) {
        alert(groups.ErrorMessage);
    }
    // Refresh the information in the table
    doShow();
    alert("Group added.");
}

function removeGroup() {
    var groupName = document.getElementById("groupName").value;
    if (groupName == "") {
        alert("You must enter a name for the group.");
        return;
    }

    // Check that the groups exists
    var groupId = -1;
    // Criteria object for making a GetList call to receive groups information
    var criteria = new Object();
```


(C) Copyright Nokia 2014. All rights reserved.

```
    criteria.Type = "Group";
    var groups = null;
    // Get list of groups
    try {
        groups = phonebook.IDataSource.GetList(criteria);
    } catch (ex) {
        alert("Error in retrieving group list");
        return;
    }

    if (groups.ErrorCode != 0) {
        // If the error code is not "no suitable items", show an alert;
        // Otherwise just leave the function
        if (groups.ErrorCode != noItemsErrorCode) {
            alert(groups.ErrorMessage);
        }
        return;
    } else {
        // Fill the table with the received data
        var item;
        while ((item = groups.ReturnValue.getNext()) != undefined) {
            if (item.GroupLabel != undefined) {
                if (item.GroupLabel == groupName) {
                    groupId = item.id;
                    break;
                }
            }
        }
    }

    if (groupId == -1) {
        alert("No such group.");
        return;
    }

    criteria = new Object();
    criteria.Type = "Group";
    criteria.Data = new Object();
    // Add the groups list to criteria. You can add more than one group.
    criteria.Data.IdList = new Array();
    criteria.Data.IdList[0] = groupId;

    groups = null;
    try {
        groups = phonebook.IDataSource.Delete(criteria);
    } catch(err) {
        alert("Error in removing the group.");
        return;
    }
    if (groups.ErrorCode != 0) {
        alert(groups.ErrorMessage);
    }
    // Refresh the information in the table
    doShow();
    alert("Group removed.");
}
```


(C) Copyright Nokia 2014. All rights reserved.

Postconditions

The adding and removing of groups is demonstrated.

Supplementary material

This code snippet is part of the stub concept, which means that it has been patched on top of a template application in order to be more useful to developers. The version of the WRT stub application used as a template in this snippet is v1.1.

- The patched, executable application that can be used to test the features described in this snippet is available for download at Media:Managing phone groups in WRT.zip.
- You can view all the changes that are required to implement the above-mentioned features. The changes are provided in unified diff and colour-coded diff (HTML) formats in Media:ManagingPhoneGroupsWRT.diff.zip.
- For general information on applying the patch, see Using Diffs.
- For unpatched stub applications, see Example app stubs with logging framework.

(C) Copyright Nokia 2014. All rights reserved.