



ID	CS001242	Creation date	December 18, 2008
Platform	S60 5th Edition	Tested on devices	
Category	Web Runtime (WRT)	Subcategory	Messaging

Keywords (APIs, classes, methods, functions): device.getServiceObject(), Service.Messaging.Send()

Overview

This code snippet shows how to send SMS messages synchronously and asynchronously using the Messaging Platform Service for S60 Web Runtime introduced in S60 5th Edition.

To obtain access to the service object for the Messaging Service API, the `device.getServiceObject("Service.Messaging", "IMessaging")` method is used.

After setting the correct values for the message type (`criteria.MessageType`), recipient (`criteria.To`), and body text of the message (`criteria.BodyText`), the `IMessaging.Send(criteria)` method is used to send the SMS.

Source

```
<?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>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <script type="text/javascript" src="sendMessage.js" />
    <title></title>
  </head>
  <body>
    <div id="bodyContent" class="bodyContent">
      <input type="checkbox" id="async" />
      <label for="async">Send asynchronously</label><br />
      <label for="phoneNumber">Phone number:</label><br />
      <input type="text" id="phoneNumber" size="12" maxlength="12" />
      <br />
      <label for="message">Message:</label><br />
      <textarea id="message" cols="40" rows="3"></textarea><br />
      <input type="button" value="Send" onclick="sendSMS();" />
    </div>
  </body>
</html>
```

```

    </div>
  </body>
</html>

```

Source: sendMessage.js

```

var serviceObj = null;

window.onload = init;

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

function sendSMS() {
  var criteria = new Object();
  //Setting the type of the message
  criteria.MessageType = "SMS";

  var phoneNumber = document.getElementById("phoneNumber").value;
  if (phoneNumber != null) {
    //Setting the "To" field of the message, can't be empty
    criteria.To = phoneNumber;
  } else {
    alert("Phone number is empty");
    return;
  }

  var messageText = document.getElementById("message").value;
  if (messageText != null) {
    //Setting the body text field of the message, can't be empty
    criteria.BodyText = messageText;
  } else {
    alert("Text is empty");
    return;
  }

  try {
    if (document.getElementById("async").checked == false) {
      //Send the message synchronously
      var result = serviceObj.IMessaging.Send(criteria);
      checkError(result);
    } else {
      //Send the message asynchronously
      serviceObj.IMessaging.Send(criteria, onSendDone);
    }
  } catch(exception) {
    alert("SendSMS error: " + exception);
  }
}

// Called when asynchronous message sending has completed

```

CS001242_-_Sending_SMS_messages_in_WRT

```
function onSendDone(transId, eventCode, result) {
    checkError(result);
}

function checkError(error) {
    if (error.ErrorCode != 0) {
        alert("Error in sending message");
    } else {
        alert("Message was sent succesfully");
    }
}
```

Postconditions

- When the snippet is started, the user can choose to send the SMS synchronously or asynchronously (check box).
- Input field is used to define the receiver's phone number.
- To send the message press "Send".

Supplementary material

You can view the source file and the executable application in the attached ZIP archive. The archive is available for download at [Media:Sending sms messages in WRT.zip](#).