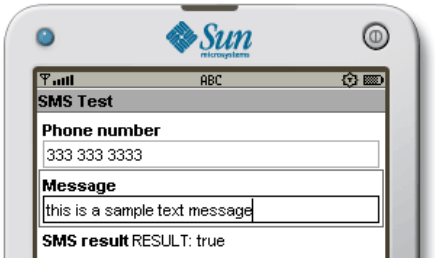
# Como enviar uma mensagem SMS com Java ME

A Wireless Messaging API (WMAPI) permite que os aplicativos Java ME acessem os recursos de comunicação wireless, como envio e recebimento de mensagens SMS e MMS. Este artigo explica como utilizá-lo para enviar uma mensagem de texto simples.

## Overview

A seguinte dica Java ME explica um método de envio de mensagens de texto usando a WMAPI. No programa, o objeto `TextMessage` representa uma nova mensagem de texto. Já o método `setPayloadText()` define os caracteres na mensagem.




```
public boolean sendSms(String number, String message){
  boolean result = true;
  try {
    //define o endeço da mensagem
    String addr = "sms://"+number;
    // abre a conexão
    MessageConnection conn = (MessageConnection) Connector.open(addr);
    // prepara a mensagem
    TextMessage msg = (TextMessage)conn.newMessage(MessageConnection.TEXT_MESSAGE);
    // define o texto
    msg.setPayloadText(message);
    // envia a mensagem
    conn.send(msg);
    conn.close();
  } catch (SecurityException se) {
      // Provavelmente o usuário não tenha permissão para enviar sms
      // Você pode querer lidar com isso de forma diferente
      result = false;
  } catch (Exception e) {
      result = false;
  }
  return result;
}
```

### Download

Você pode baixar um exemplo MIDlet mostrando o código apresentado neste artigo aqui.

### Notas

- Lembre-se que uma única mensagem tem um tamanho máximo (140 bytes, o que se traduz em 160 caracteres de 7 bits). Se a sua mensagem é muito longa para caber em uma mensagem, ela pode ser dividida em um máximo de três SMS, isso é feito automaticamente pela aplicação WMA.
- **Operações com potencial de bloqueio**, como o envio de uma mensagem de texto, **deve sempre ser realizada em uma Thread separada**, para evitar o bloqueio da Thread do MIDlet principal.

## Links relacionados

- WMAPI JavaDocs
- Wireless messaging section on Java ME Developer's Library


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.