Anu's crazy world

Not just an another WordPress blog

Sending SMS app in J2ME

Posted on July 13, 2011

Greetings friends 🙂 Today I am going to develop a mobile application which has SMS (Short Message Service) sending functionality. Specialty is I am going to use an optional package named as Wireless Messaging API to accomplish my task. There are some other optional packages as well. I will explain them with suitable applications. Today lets see some of the functions providing by Wireless Messaging API.

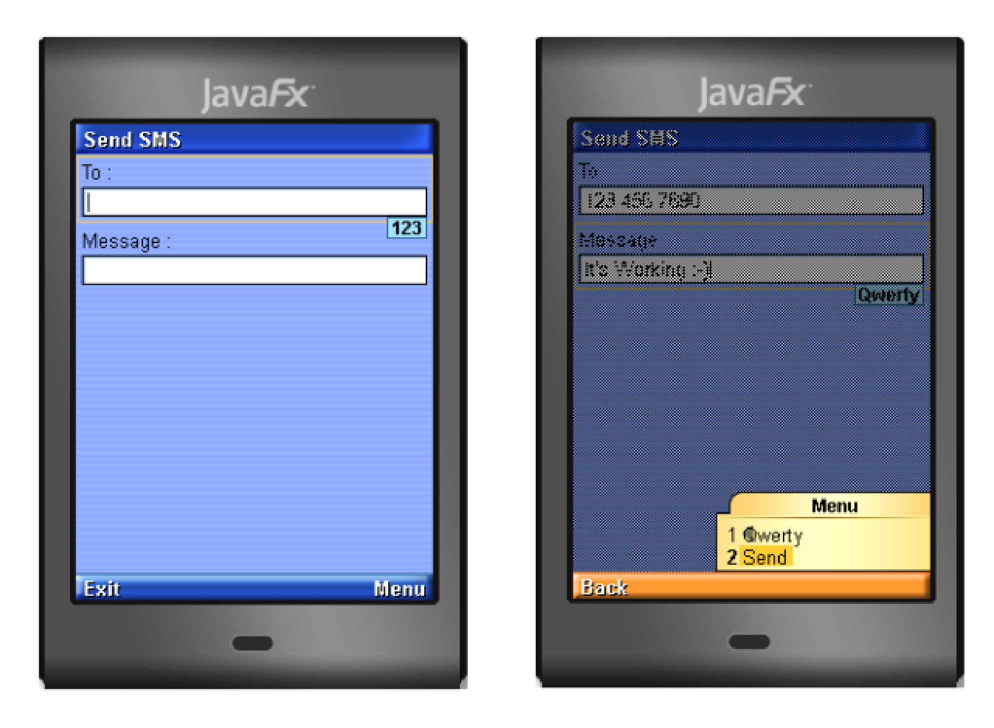
The class that you are going to newly meet today is,

- Connector

I have to remind you something. Some of these classes and interfaces that I am mentioning is in J2SE and others are J2ME. So the interfaces that new in this post are,

- MessageConnection
- TextMessage

I am not much concern about the user interface here. But I have uploaded some screen shots to help you to understand the code sample.



123 456 7890

It's Working :-)

Figure 1

Scenario is very simple. You give a phone number, then write the message and finally press on the *Send* command. You may confuse how to check whether this application is working or not. There are two options that you can try. First one is copy the code, clean and build, deploy and then install it in your phone. The second method is, you can use two different phone emulators. I will explain it in one of my future post. Till then try with the first method 🙂

Let’s see the code sample. The main thing you have to focus on the code is how I have used the Wireless Messaging API to open a connection, set pay load and send a SMS.

```
package anuja;

import java.io.IOException;
import javax.microedition.io.Connector;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.TextField;
import javax.microedition.midlet.MIDlet;
import javax.wireless.messaging.MessageConnection;
import javax.wireless.messaging.TextMessage;

/**
 * @author Anuja
 */
public class MySMS extends MIDlet implements CommandListener {
    private Display display;
    private Form smsFrm;
    private TextField phnNoTxtFld;
    private TextField smsTxtFld;
    private Command exitCmd;
    private Command sendCmd;
    private Alert infoAlert;
    private String smsStr;
    private String phnNoStr;

    public void startApp() {
        display = Display.getDisplay(this);

        smsFrm = new Form("Send SMS");
        phnNoTxtFld = new TextField("To : ", "", 20, TextField.PHONENUMBER);
        smsFrm.append(phnNoTxtFld);
        smsTxtFld = new TextField("Message : ", "", 150, TextField.ANY);
        smsFrm.append(smsTxtFld);

        exitCmd = new Command("Exit", "Exit from the app", Command.EXIT, 3);
        smsFrm.addCommand(exitCmd);
        sendCmd = new Command("Send", Command.OK, 4);
        smsFrm.addCommand(sendCmd);

        smsFrm.setCommandListener(this);
        display.setCurrent(smsFrm);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }

    public void commandAction(Command c, Displayable d) {
        if(c == exitCmd){
            notifyDestroyed();
        }else if(c == sendCmd){

            phnNoStr = phnNoTxtFld.getString();
            smsStr = smsTxtFld.getString();
            // MessageConnection is an optional package. That package named as JSR 120: Wireless Messag
            MessageConnection msgCon = null;

            try {
                // Connector class creates new Connection object
                // open() Create and open a Connection.
                msgCon = (MessageConnection) Connector.open("sms://" + phnNoStr);
            } catch (IOException ex) {
                ex.printStackTrace();
            }

            // A TextMessage object is used to send a message containing a java.lang.String.
            // It inherits from the Message interface and adds a text message body.
            TextMessage txtMsg = (TextMessage) msgCon.newMessage(MessageConnection.TEXT_MESSAGE);

            //set your message payload data in to message object
            txtMsg.setPayloadText(smsStr);

            try {
                // send the message
                msgCon.send(txtMsg);

                clearTxt();
            } catch (IOException ex) {
                ex.printStackTrace();
            }finally{
                if(msgCon != null){
                    try {
                        // Close the connection
                        msgCon.close();
                    } catch (IOException ex) {
                        ex.printStackTrace();
                    }
                }
            }
        }
    }

    public void clearTxt(){
        phnNoTxtFld.setString("");
        smsTxtFld.setString("");
        infoAlert = new Alert("Message sent successfully...!");
        display.setCurrent(infoAlert, smsFrm);
    }
}
```

When you are trying to send a SMS, a warning message will display. I haven't handle that issue in this post. The reason for that alert message is, we are going to use users, credit to open a connection.

That's the story of this application. Bye 🙂

Rate this: ☆☆☆☆☆ Rate This

Share this:

Like this: ★ Like

Be the first to like this.

About anujarosha

An undergraduate in the stream of ICT (Information & Communication Technology). A simple person :)

View all posts by anujarosha →

This entry was posted in J2ME examples. Bookmark the permalink.

7 Responses to *Sending SMS app in J2ME*

AJOSEA *says:*

September 26, 2011 at 9:04 am

HELLO I AM NEW HEAR DO YOU HAVE STEPS ON HOW TO DI THIS?
I MEAN SHOULD I USE ECLIPSE?
IM REAAALY REALLLLY NEW.
DO U HAVE ANY PRINTSCREENS? NEW PROJECT? AND SO ON, PLS HELP ME I WANT TO SEE THE DETAILED CODING AND HOW TO'S THANKS MY FRIEND!
THIS IS MY EMAIL

REYES.JOSEA@YAHOO.COM

Reply

anujarosha *says:*

September 26, 2011 at 3:43 pm

Hi,

You can use Eclipse too. But I have used NetBeans IDE. Visit my first blog in J2ME (http://anujarosha.wordpress.com/2011/05/03/lets-learn-j2me-with-mobile-applications/) which describe the configurations that I have used.

This is the complete code for sending SMS application. I have checked it with a Nokia 5700 phone and it's working without any error.

I have commented with in my coding so I guess this is a "detailed coding" 😉

You can post your problems here while you are doing it. Then you get the answers for your "how to's "

Best of luck 😀

Reply

parul *says:*

March 28, 2012 at 12:40 pm

how can i implement phonebook in this code so i can send sms from sim.
reply soon
this is my email id parulcomeng@gmail.com

Reply

aisha *says:*

June 24, 2012 at 11:11 pm

Hi did you use jade leap with j2me or not???

Reply

anujarosha *says:*

June 25, 2012 at 8:03 pm

Hi Aisha,

No, I haven't used it.

Reply

Chithraa *says:*

February 23, 2013 at 12:45 pm

Hiiii anujarosha sir ….. I was given with an assignment in my college ….So i decided 2 do chat box …referred yur post …..I used j2me toolkit 2.2 and run your program,it showing exceptions …..Can yu send me the step by step procedure 2 my mail skcchithraa@gmail.com ..Please help me sir 🙂

Reply

samuel *says:*

April 20, 2013 at 5:07 pm

it is nice

Reply

Anu's crazy world

The Twenty Ten Theme. Blog at WordPress.com.