Finding and viewing contacts in Java ME

Overview

This snippet demonstrates how to search for a specific contact in phone contact list.

After user has enterd the last name of the contact he or she wants to find, the application retrieves the currect contact list and searches for the required value using the method `Contact.items()`.

Source file: FindViewContact.java

```
import java.util.Enumeration;
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.TextField;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Displayable;
import javax.microedition.pim.ContactList;
import javax.microedition.pim.Contact;
import javax.microedition.pim.PIM;
import javax.microedition.pim.PIMItem;
import javax.microedition.pim.PIMException;

public class FindViewContact extends MIDlet implements CommandListener {

    //Main form.
    private Form form;
    private Display display;
    private Command exitCommand;
    private Command searchCommand;
    // Contains family name for searching.
    private TextField searchName;

    /**
     * Constructor. Constructs the object and initializes displayables.
     */
    public FindViewContact() {
        form = new Form("Find/View PIM Contact");

        // Adding command buttons
        searchCommand = new Command("Search", Command.OK, 0);
        exitCommand = new Command("Exit", Command.EXIT, 1);
        form.addCommand(searchCommand);
        form.addCommand(exitCommand);

        searchName = new TextField("Enter family name:", "", 20,
                                   TextField.ANY);

        form.append(searchName);


        form.setCommandListener(this);
        display = Display.getDisplay(this);
        display.setCurrent(form);
    }

    /**
     * From CommandListener.
     * Called by the system to indicate that a command has been invoked on a
     * particular displayable.
     * @param cmd the command that was invoked
     * @param displayable the displayable where the command was invoked
     */
    public void commandAction(Command cmd, Displayable displayable) {
        if (cmd == exitCommand) {
            notifyDestroyed();
        } else if (cmd == searchCommand) {
            searchContact();
        }
    }

    /**
     * Called when user entered family name. Displays search results.
     */
    private void searchContact() {
        ContactList contacts = null;
        try {
            // Retrieve the contact list.
            contacts = (ContactList) PIM.getInstance().
                    openPIMList(PIM.CONTACT_LIST, PIM.READ_ONLY);
            Enumeration result = null;

            //Create new temporary contact for searching.
            Contact searchItem = contacts.createContact();
            String[] name =
                    new String[contacts.stringArraySize(Contact.NAME)];
            if (contacts.isSupportedArrayElement(
                    Contact.NAME, Contact.NAME_FAMILY)) {
                name[Contact.NAME_FAMILY] = searchName.getString();
            }
            searchItem.addStringArray(Contact.NAME, PIMItem.ATTR_NONE, name);

            //Get enumeration of matching contacts.
            result = contacts.items(searchItem);

            // Clean results list.
            int size = form.size();
            for (int i = 1; i < size; i++) {
```


(C) Copyright Nokia 2013. All rights reserved.

```
                form.delete(1);
            }
            Contact item = null;
            //If nothing was found.
            if (!result.hasMoreElements())
            {
                form.append("No contacts found for: \""
                            + searchName.getString()+"\"");
            }
            while (result.hasMoreElements()) {
                // Display new results.
                item = (Contact) result.nextElement();
                form.append(item.getStringArray(Contact.NAME,
                        Contact.ATTR_NONE)[Contact.NAME_GIVEN]);
                form.append(" ");
                form.append(item.getStringArray(Contact.NAME,
                        Contact.ATTR_NONE)[Contact.NAME_FAMILY]);
                form.append("\n");
            }
            contacts.removeContact(searchItem);
        } catch (PIMException e) {
            //TODO: Error handling
        }
    }

    /**
     * From MIDlet.
     * Called when the MIDlet is started.
     */
    public void startApp() {
        // No implementation required.
    }

    /**
     * From MIDlet.
     * Called to signal the MIDlet to enter the Paused state.
     */
    public void pauseApp() {
        // No implementation required.
    }

    /**
     * From MIDlet.
     * Called to signal the MIDlet to terminate.
     * @param unconditional whether the MIDlet has to be unconditionally
     * terminated
     */
    public void destroyApp(boolean unconditional) {
        // No implementation required
    }
}
```

Postconditions

After the user has enterd the last name family name of a contact into a text field and pressed «search», the search results are shown in below the input field. If any errors occurs, a message describing the error is diplayed.

Supplementary material

The source file and the executable application are available for download at Media:Finding and viewing contacts in J2ME.zip.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.