Listing calendar events in Java ME

The following code snippet demonstrates how to list events in the calendar using the PIM API.

Preconditions

In order to test this snippet there should be at least one event in the database. The application also needs permission to read and write PIM database data.

Source

```
import java.util.Enumeration;
import javax.microedition.lcdui.Choice;
import javax.microedition.lcdui.List;
import javax.microedition.pim.Event;
import javax.microedition.pim.EventList;
import javax.microedition.pim.PIM;
import javax.microedition.pim.PIMException;
import javax.microedition.pim.PIMItem;
```

```
private List listEvents;

/**
 * Holds reference to the currently opened event list.
 */
private EventList eventList;

private static final Command BACK_COMMAND =
        new Command("Back", Command.BACK, 0);
```

```
/**
 * Instantiates the event list.
 */
private void setupEventList() {
    listEvents = new List("Event list", Choice.IMPLICIT);
    listEvents.addCommand(BACK_COMMAND);
    listEvents.setCommandListener(this);
}
```

```
/**
 * Executes the snippet.
 */
private void executeSnippet() {
    openEventList();
}

/**
 * Fills the event list with data read from the PIM database and brings
 * the list to the foreground.
 */
private void openEventList() {
    fillEventList();
```


(C) Copyright Nokia 2014. All rights reserved.

```
    display.setCurrent(listEvents);
}

/**
 * Reads the event list from the PIM database and fills the list component
 * with its information.
 */
private void fillEventList() {
    try {
        printString("Opening event list ...");
        eventList = (EventList)PIM.getInstance().openPIMList(
                PIM.EVENT_LIST, PIM.READ_WRITE);
    } catch (PIMException ex) {
        printString(ex.toString());
        return;
    }
    Enumeration eventItems = null;
    Event event = null;
    try {
        printString("Getting event list items...");
        eventItems = eventList.items();
    } catch (PIMException ex) {
        releaseEventList();
        printString(ex.toString());
        return;
    }
    // Checking event count
    if (!eventItems.hasMoreElements()) {
        printString("Event list is empty!");
    }
    if (listEvents.size() > 0) {
        listEvents.deleteAll();
    }
    while (eventItems.hasMoreElements()) {
        event = (Event)eventItems.nextElement();
        String eventInfo = null;
        try {
            eventInfo =
                    event.getString(Event.SUMMARY, PIMItem.ATTR_NONE);
        } catch (Exception ex) {
            printString(ex.getMessage());
            continue;
        }
        if (eventInfo != null) {
            listEvents.append(eventInfo, null);
        }
    }
}

/**
 * Closes the event list if one is opened.
 */
private void releaseEventList() {
    if (eventList != null) {
        try {
            printString("Closing event list");
            eventList.close();
```


(C) Copyright Nokia 2014. All rights reserved.

```
        } catch (PIMException ex) {
            printString(ex.toString());
        }
    }
    eventList = null;
}
```

```
/**
 * From CommandListener.
 * Called by the system to indicate that a command has been invoked on a
 * particular displayable.
 * @param command the command that was invoked
 * @param displayable the displayable where the command was invoked
 */
public void commandAction(Command command, Displayable displayable) {
    if (command == BACK_COMMAND) {
        display.setCurrent(mainForm);
        // Close the event list
        releaseEventList();
    }
}
```

Postconditions

The events in the calendar are listed on the screen.

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 Java ME 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:ListEvents.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:ListEvents.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.