```java
package fi.paatti.mobile.views.listview;


import com.vaadin.data.Item;
import com.vaadin.terminal.ThemeResource;
import com.vaadin.ui.Alignment;
import com.vaadin.ui.Button;
import com.vaadin.ui.Button.ClickEvent;
import com.vaadin.ui.Embedded;
import com.vaadin.ui.HorizontalLayout;
import com.vaadin.ui.Label;
import com.vaadin.ui.NativeButton;
import com.vaadin.ui.VerticalLayout;
import fi.paatti.mobile.applicationinfo.CssStyleNames;
import fi.paatti.mobile.applicationinfo.Resources;


/**
 * A single item in a ListView.
 * <p/>
 * @author Lauri Satokangas, lauri.n.satokangas@student.jyu.fi @date 9.3.2012
 * @since 0.0.1
 */
public class ListItem extends HorizontalLayout {

  private static final long serialVersionUID = 1L;
  private Label headerLabel;
  private ThemeResource itemIcon = null;
  private Object itemId;
  private NativeButton openButton;
  private Embedded iconEmb;
  private VerticalLayout labelLayout;

  /**
   * Constructor.
   * <p/>
   * The ID will be taken from the item and the items text rows will be set using the names in the textRows param.
   * <p/>
   * @param item      The item for the list item.
   * @param idColumn The ID column in the item.
   * @param textRows The property names in the item.
   */
  public ListItem(Item item, String idColumn, String... textRows) {
    init();
    try {
      itemId = item.getItemProperty(idColumn).getValue();
      if (textRows.length > 0) {
        this.headerLabel.setPropertyDataSource(item.getItemProperty(textRows[0]));
      }
      for (int i = 1; i < textRows.length; i++) {
        Label textRow = new Label(item.getItemProperty(textRows[i]).getValue().toString());

        textRow.setStyleName(CssStyleNames.LISTITEM_TEXTROW_LABEL);
        labelLayout.addComponent(textRow);
      }
    } catch (NullPointerException e) {// TODO: logging.
    }
  }

  /**
   * Initialize the list item.
   */
  private void init() {
    setSpacing(true);
    setSizeFull();
    labelLayout = new VerticalLayout();

    headerLabel = new Label();
    headerLabel.setStyleName(CssStyleNames.LISTITEM_HEADER_LABEL);
    headerLabel.setImmediate(true);

    openButton = new NativeButton();
    labelLayout.addComponent(headerLabel);
    labelLayout.setSizeFull();

    addComponent(labelLayout);
    addComponent(openButton);

    setExpandRatio(labelLayout, 1f);
    setExpandRatio(openButton, 0f);
    setItemIcon(itemIcon);

    setComponentAlignment(labelLayout, Alignment.MIDDLE_CENTER);
    setComponentAlignment(openButton, Alignment.MIDDLE_CENTER);
    this.setMargin(true);
  }

  /**
   * Set icon for the item.
   * <p/>
   * @param icon Themeresource containing the icon.
   */
```

```java
  public void setItemIcon(ThemeResource icon) {
    iconEmb = new Embedded();
    iconEmb.setSource(itemIcon);
    addComponent(iconEmb, 0);
    setComponentAlignment(iconEmb, Alignment.MIDDLE_CENTER);
  }

  /**
   * Return the ID of the item.
   * <p/>
   * @return
   */
  public Object getItemId() {
    return itemId;
  }

  /**
   * Add an item click listener to the button in the list item.
   * <p/>
   * @param logic The logic that will be used when the button click event happens.
   */
  public void setItemClickLogic(final ListViewLogic logic) {
    openButton.addListener(new NativeButton.ClickListener() {

      private static final long serialVersionUID = 1L;

      @Override
      public void buttonClick(ClickEvent event) {
        // TODO: tarviiko muuta tietoja ku itemId:n?
        logic.listItemClicked(itemId);
      }
    });
  }

  /**
   * Add a new row of text to the item.
   * @param labelText The text that will be added
   */
  public void addTextRow(String labelText) {
    Label textRow = new Label(labelText);

    textRow.setStyleName(CssStyleNames.LISTITEM_TEXTROW_LABEL);
    labelLayout.addComponent(textRow);
  }

  /**
```

```
   * Removes the open-button from the list item so it can not be opened.
   * <p/>
   * @param openEnabled Can the item be opened.
   */
  public void setOpenButtonEnabled(boolean openEnabled) {
    openButton.setVisible(openEnabled);
    openButton.setEnabled(openEnabled);
  }

  /**
   * Set text for the opening button.
   * <p/>
   * @param text The text to be set.
   */
  public void setOpenButtonText(String text) {
    openButton.setCaption(text);
  }

  /**
   * Set icon for the opening button.
   * <p/>
   * @param icon The icon to be set.
   */
  public void setOpenButtonIcon(ThemeResource icon) {
    openButton.setIcon(icon);

  }
}
```