

```
package fi.paatti.mobile.views.voluntaryeventlistview;

import com.vaadin.data.Item;
import com.vaadin.data.util.sqlcontainer.SQLContainer;
import com.vaadin.ui.Label;
import fi.paatti.mobile.applicationinfo.Resources;
import fi.paatti.mobile.mobileview.MobileViewLogic;
import fi.paatti.mobile.paattiapplication.MobileViewHandler;
import fi.paatti.mobile.views.eventlistview.EventListItem;
import fi.paatti.mobile.views.eventlistview.EventListView;
import fi.paatti.paattidatabasetils.names.PaattiColumnNames;
import fi.paatti.paattidatabasetils.names.PaattiTableNames;
import java.io.Serializable;

/**
 * A listview for all the voluntary events.
 */
@author Lauri Satokangas, lauri.n.satokangas@student.jyu.fi
*/
public class VoluntaryEventListView extends EventListView implements Serializable {

    private static final long serialVersionUID = 1L;
    private VoluntaryEventListViewLogic coluntaryEventListViewLogic;
    private Label noItemsLabel = new Label("Sinulla ei ole yhtään vapaaehtoista harjoitetta.");

    /**
     * The constructor.
     */
    @param viewHandler The applications view handler.
    */
    public VoluntaryEventListView(MobileViewHandler viewHandler) {
        super(viewHandler);
        createUi();
    }

    private void createUi() {
        updateFooterAndHeader();
        addComponent(noItemsLabel);
    }

    @Override
    public void updateFooterAndHeader() {
        getViewHandler().setHeaderText("Vapaaehtoiset harjoitukset");
    }
}
```

```
getViewHolder().setFooterButtonIcon(FOOTER_BUTTON_LEFT, Resources.getThemeResource(Resources.ICON_ARROW_LEFT_64));
getViewHolder().setFooterButtonEnabled(FOOTER_BUTTON_MIDDLE, false);
getViewHolder().setFooterButtonEnabled(FOOTER_BUTTON_RIGHT, false);
}

@Override
public void setContentViewModel(MobileViewLogic logic) {
    // super.setContentViewModel(VoluntaryEventListViewLogic logic);
    voluntaryEventListViewLogic = (VoluntaryEventListViewLogic) logic;

    // The list creation could be moved somewhere else.
    createListItemsFromDb(PaattiTableNames.EVENT, PaattiColumnNames.EVENT_eventID);
}

/**
 * Add list items from the database to the layout.
 * <p/>
 * Can be used to create the list from voluntary events.
 * <p/>
 * @param tableName Name of the table in the database.
 * @param tableIdColumnName The name of the id column in the table.
 */
@Override
public void createListItemsFromDb(String tableName, String tableIdColumnName) {
    // Remove old components before adding new.
    this.removeAllComponents();

    SQLContainer listItemsContainer = getViewHandler().getDbService().getUserVoluntaryEvents(
        getViewHandler().getUserID());

    Object itemId = listItemsContainer.getFirstItemId();
    Item eventItem;

    // Go through all the items in the container and create list items from the container items.
    while (itemId != null) {
        EventListItem eventListItem = null;

        eventItem = listItemsContainer.getItem(itemId);

        eventListItem = new EventListItem(eventItem, tableIdColumnName, getHeaderTextColumnName());
        // TODO: fix to work with
        if (eventItem.getItemProperty(PaattiColumnNames.USERGROUP_description) != null
            && eventItem.getItemProperty(PaattiColumnNames.USERGROUP_description).getValue() != null) {
            eventListItem.addRow(
                eventItem.getItemProperty(PaattiColumnNames.USERGROUP_description).getValue().toString());
        }
    }
}
```

```
    if (eventItem.getItemProperty(PaattiColumnNames.SCHEDULE_description) != null
        && eventItem.getItemProperty(PaattiColumnNames.SCHEDULE_description).getValue() != null) {
        eventListItem.addRow(eventItem.getItemProperty(PaattiColumnNames.SCHEDULE_description).getValue().toString());
    }

    eventListItem.addRow(coluntaryEventListViewLogic.getEstimatedTimeAsString(eventItem));
    eventListItem.setItemClickLogic(coluntaryEventListViewLogic);

    eventListItem.setStatus(EventListItem.STATUS_VOLUNTARY);
    addComponent(eventListItem);

    itemId = listItemsContainer.nextItemId(itemId);

    // If there are no events, show the label.
    noItemsLabel.setVisible(listItemsContainer.firstItemId() == null ? false : true);

}
}
```