

```
package fi.paatti.research.paattiapplication.views.groupview;

import com.vaadin.data.Item;
import com.vaadin.data.util.sqlcontainer.SQLContainer;
import com.vaadin.ui.Button.ClickEvent;
import com.vaadin.ui.*;
import fi.paatti.paattidatabasutils.dbsevice.PaattiResearchDBService;
import fi.paatti.paattidatabasutils.names.PaattiColumnNames;
import fi.paatti.paattidatabasutils.names.PaattiTableNames;
import fi.paatti.research.paattiapplication.ElementNames;
import fi.paatti.research.paattiapplication.views.ApplicationTabSheet;
import fi.paatti.research.paattiapplication.views.ApplicationView;
import fi.paatti.research.paattiapplication.views.ApplicationViewTab;

/**
 * The tab shows the users of the selected group.
 */
 * @author Toni Salminen, toni.a.j.salminen@student.jyu.fi
 */
public class GroupUsersInfoTab extends ApplicationViewTab {

    private Table usersTable;
    private Select userSelect;
    private final String[] visibleColumns = new String[] {
        PaattiColumnNames.USER_name, PaattiColumnNames.USER_description;
    };
    private final String[] visibleHeaders = new String[] { "Nimi", "Kuvaus" };
    private SQLContainer usersContainer;

    /**
     * Constructor of the class.
     */
    * @param view the parent view that contains this tab.
    */
    public GroupUsersInfoTab(ApplicationView view) {
        super(view);
        createLayout();
    }

    /**
     * {@inheritDoc}
     */
    @Override
    protected void createLayout() {
        usersTable = new Table();
    }
}
```

```
usersTable.setSelected(true);
usersTable.setImmediate(true);
usersTable.setRequired(true);
usersTable.setSizeFull();

usersContainer = view.getConnection().getSQLContainerFromDBTable(PaattiTableNames.USER,
    PaattiColumnNames.USER_userid, true);
userSelect = new Select(ElementNames.GROUP_VIEW_SELECT_USER, usersContainer);
userSelect.setItemCaptionPropertyId(PaattiColumnNames.USER_name);
userSelect.setNullSelectionAllowed(false);
userSelect.setInvalidAllowed(false);
userSelect.setRequired(true);
userSelect.setSizeUndefined();
userSelect.setWidth("100%");

Button addUserToGroupButton = new Button(ElementNames.GROUP_VIEW_ADD_USER_BUTTON);

addUserToGroupButton.addListener(this);
addUserToGroupButton.setSizeUndefined();
addUserToGroupButton.setWidth("100%");

Button removeUserFromGroupButton = new Button(ElementNames.GROUP_VIEW_REMOVE_USER_BUTTON);

removeUserFromGroupButton.addListener(this);
removeUserFromGroupButton.setSizeUndefined();
removeUserFromGroupButton.setWidth("100%");

Button examineUserButton = new Button(ElementNames.GROUP_VIEW_EXAMINE_USER_BUTTON);

examineUserButton.addListener(this);
examineUserButton.setSizeUndefined();
examineUserButton.setWidth("100%");

HorizontalLayout addUserToGroupLayout = new HorizontalLayout();

addUserToGroupLayout.addComponent(userSelect);
addUserToGroupLayout.addComponent(addUserToGroupButton);
addUserToGroupLayout.setComponentAlignment(userSelect, Alignment.BOTTOM_LEFT);
addUserToGroupLayout.setComponentAlignment(addUserToGroupButton, Alignment.BOTTOM_RIGHT);
addUserToGroupLayout.setMargin(true);
addUserToGroupLayout.setSpacing(true);
addUserToGroupLayout.setSizeFull();

VerticalLayout layout = new VerticalLayout();

layout.setMargin(true);
```

```
layout.addComponent(usersTable);
layout.addComponent(addUserToGroupLayout);
layout.addComponent(removeUserFromGroupButton);
layout.addComponent(examineUserButton);
layout.setComponentAlignment(removeUserFromGroupButton, Alignment.BOTTOM_CENTER);
layout.setComponentAlignment(examineUserButton, Alignment.BOTTOM_CENTER);
layout.setExpandRatio(usersTable, 30);
layout.setExpandRatio(addUserToGroupLayout, 7);
layout.setExpandRatio(removeUserFromGroupButton, 1);
layout.setExpandRatio(examineUserButton, 1);

setCompositionRoot(layout);
}

/**
 * {@inheritDoc}
 */
@Override
protected void fillTab(Item group) {
    SQLContainer groupUsers = view.getConnection().getGroupUsers(
        group.getItemProperty(PaattiColumnNames.USERGROUP_userid));

    usersTable.setContainerDataSource(groupUsers);
    usersTable.setVisibleColumns(visibleColumns);
    usersTable.setColumnHeaders(visibleHeaders);
    setEnabled(true);
}

/**
 * {@inheritDoc}
 */
@Override
protected void clearTab() {
    usersTable.setContainerDataSource(null);
    setEnabled(false);
}

/**
 * {@inheritDoc}
 * <p/>
 * The handled button click events are:
 * <ul>
 * <li>add the selected user to the group,</li>
 * <li>remove the selected user from the group, and</li>
 * <li>examine the selected user.</li>
 * </ul>

```

```
*/
@Override
public void buttonClick(ClickEvent event) {
    String buttonCaption = event.getButton().getCaption();
    String msg = null;
    PaattiResearchDBService db = view.getDBConnection();

    if (buttonCaption.equals(ElementNames.GROUP_VIEW_ADD_USER_BUTTON)) {
        if (userSelect.isValid()) {
            msg = db.addUserToGroup(view.getSelectedItemId(), userSelect.getValue()
                ? ElementNames.UPDATE_OK
                : ElementNames.UPDATE_FAIL;
            view.refresh(view.getSelectedItemId());
        } else {
            msg = ElementNames.SOMETHING_REQUIRED;
        }
    } else if (buttonCaption.equals(ElementNames.GROUP_VIEW_REMOVE_USER_BUTTON)) {
        if (usersTable.isValid()) {
            msg = db.removeUserFromGroup(view.getSelectedItemId(), usersTable.getValue()
                ? ElementNames.UPDATE_OK
                : ElementNames.UPDATE_FAIL;
            view.refresh(view.getSelectedItemId());
        } else {
            msg = ElementNames.UPDATE_FAIL;
        }
    } else if (buttonCaption.equals(ElementNames.GROUP_VIEW_EXAMINE_USER_BUTTON)) {
        if (usersTable.isValid()) {
            view.setActiveTab(ApplicationTabSheet.USER_TAB, usersTable.getValue());
        } else {
            msg = ElementNames.SOMETHING_REQUIRED;
        }
    }
    if (msg != null) {
        getWindow().showNotification(msg);
    }
}
```