

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

import com.vaadin.data.Container.Filter;
import com.vaadin.data.Item;
import com.vaadin.data.util.filter.Compare;
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 schedules of the selected group.
 */
 * @author Toni Salminen, toni.a.j.salminen@student.jyu.fi
 */
public class GroupSchedulesInfoTab extends ApplicationViewTab {

    private Table schedulesTable;
    private Select scheduleSelect;
    private final String[] visibleColumns = new String[] { PaattiColumnNames.SCHEDULE_description };
    private final String[] visibleHeaders = new String[] { "Kuvaus" };
    private SQLContainer availableSchedulesContainer;

    /**
     * Constructor of the class.
     * @param view
     */
    public GroupSchedulesInfoTab(ApplicationView view) {
        super(view);
        createLayout();
    }

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

```
schedulesTable.setSelected(true);
schedulesTable.setRequired(true);
schedulesTable.setSizeFull();

Filter availableSchedulesFilter = new Compare.Equal(PaattiColumnNames.SCHEDULE_USERGROUP_userid, "-1");

availableSchedulesContainer = view.getConnection().getSQLContainerFromDBTableFiltered(PaattiTableNames.SCHEDULE,
    PaattiColumnNames.SCHEDULE_scheduledID, true, availableSchedulesFilter);
scheduleSelect = new Select(ElementNames.GROUP_VIEW_SELECT_SCHEDULE_LABEL);
scheduleSelect.setContainerDataSource(availableSchedulesContainer);
scheduleSelect.setItemCaptionPropertyId(PaattiColumnNames.SCHEDULE_description);

scheduleSelect.setNullSelectionAllowed(false);
scheduleSelect.setInvalidAllowed(false);
scheduleSelect.setRequired(true);
scheduleSelect.setSizeUndefined();
scheduleSelect.setWidth("100%");

Button addButton = new Button(ElementNames.GROUP_VIEW_ADD_SCHEDULE_BUTTON);
addButton.addListener(this);
addButton.setSizeUndefined();
addButton.setWidth("100%");

Button removeButton = new Button(ElementNames.GROUP_VIEW_REMOVE_SCHEDULE_BUTTON);
removeButton.addListener(this);
removeButton.setSizeUndefined();
removeButton.setWidth("100%");

Button scheduleViewButton = new Button(ElementNames.GROUP_VIEW_EXAMINE_SCHEDULE_BUTTON);
scheduleViewButton.addListener(this);
scheduleViewButton.setSizeUndefined();
scheduleViewButton.setWidth("100%");

HorizontalLayout addScheduleToGroupLayout = new HorizontalLayout();
addScheduleToGroupLayout.addComponent(scheduleSelect);
addScheduleToGroupLayout.addComponent(addButton);
addScheduleToGroupLayout.setComponentAlignment(scheduleSelect, Alignment.BOTTOM_LEFT);
addScheduleToGroupLayout.setComponentAlignment(addButton, Alignment.BOTTOM_RIGHT);
addScheduleToGroupLayout.setMargin(true);
addScheduleToGroupLayout.setSpacing(true);
addScheduleToGroupLayout.setSizeFull();
```

```
VerticalLayout layout = new VerticalLayout();

layout.setMargin(true);
layout.addComponent(schedulesTable);
layout.addComponent(addScheduleToGroupLayout);
layout.addComponent(removeButton);
layout.addComponent(scheduleViewButton);
layout.setComponentAlignment(removeButton, Alignment.BOTTOM_CENTER);
layout.setComponentAlignment(scheduleViewButton, Alignment.BOTTOM_CENTER);
layout.setExpandRatio(schedulesTable, 30);
layout.setExpandRatio(addScheduleToGroupLayout, 7);
layout.setExpandRatio(removeButton, 1);
layout.setExpandRatio(scheduleViewButton, 1);

setCompositionRoot(layout);

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

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

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

/**
 * {@inheritDoc}
 * <p/>
 * The handled button click events are:
 * <ul>
 * <li>add the selected schedule to the group,</li>
```

```
* <li>remove the selected schedule from the group, and</li>
* <li>examine the selected schedule.</li>
* </ul>
*/
@Override
public void buttonClick(ClickEvent event) {
    String buttonCaption = event.getButton().getCaption();
    String msg = null;
    PaattiResearchDBService db = view.getConnection();

    if (buttonCaption.equals(ElementNames.GROUP_VIEW_ADD_SCHEDULE_BUTTON)) {
        if (scheduleSelect.isValid()) {
            msg = db.addScheduleToGroup(view.getSelectedItemId(), scheduleSelect.getValue()
                ? ElementNames.UPDATE_OK
                : ElementNames.UPDATE_FAIL;
            view.refresh(view.getSelectedItemId());
        } else {
            msg = ElementNames.UPDATE_FAIL;
        }
    } else if (buttonCaption.equals(ElementNames.GROUP_VIEW_REMOVE_SCHEDULE_BUTTON)) {
        if (schedulesTable.isValid()) {
            msg = db.removeRowFromTable(schedulesTable.getValue(), PaattiTableNames.SCHEDULE,
                PaattiColumnNames.SCHEDULE_scheduleID)
                ? ElementNames.UPDATE_OK
                : ElementNames.UPDATE_FAIL;
            view.refresh(view.getSelectedItemId());
        } else {
            msg = ElementNames.UPDATE_FAIL;
        }
    } else if (buttonCaption.equals(ElementNames.GROUP_VIEW_EXAMINE_SCHEDULE_BUTTON)) {
        if (schedulesTable.isValid()) {
            view.setActiveTab(ApplicationTabSheet.SCHEDULE_TAB, schedulesTable.getValue());
        } else {
            msg = ElementNames.SOMETHING_REQUIRED;
        }
    }
    if (msg != null) {
        getWindow().showNotification(msg);
    }
}
```