

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

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.paattidatabaseutils.names.PaattiColumnNames;
import fi.paatti.paattidatabaseutils.names.PaattiTableNames;
import fi.paatti.research.applicationinfo.Resources;
import fi.paatti.research.paattiapplication.ElementNames;
import fi.paatti.research.paattiapplication.views.ApplicationView;
import fi.paatti.research.paattiapplication.views.ApplicationViewTab;
import fi.paatti.research.utils.StringHelper;

/**
 * The tab that displays the basic information of the selected event.
 */
* @author Tapio Keränen, t.tapio.keranen@student.jyu.fi
*/
public class EventInfoTab extends ApplicationViewTab {
    private static final long serialVersionUID = 1L;

    private EventTool eventTool;
    private Embedded image;
    private TextField eventTitleField;
    private TextArea eventDescriptionArea;
    private Select eventTypeSelect;
    private Button saveButton;
    private Button removeButton;
    private SQLContainer eventTypeContainer;
    private TextField eventEstimatedTimeField;

    /**
     * EventInfoTab constructor.
     *
     * @param view the parent view.
     * @param tool the event tool.
     */
    public EventInfoTab(ApplicationView view, EventTool tool) {
        super(view);
        createLayout();

        eventTool = tool;
    }
}
```

```
/**
 * {@inheritDoc}
 */
@Override
protected void createLayout() {
    image = new Embedded();
    image.setSource(Resources.getThemeResource(Resources.IMG_BOOK));
    image.setWidth("125px");

    eventTitleField = new TextField(ElementNames.EVENT_VIEW_NAME_FIELD_LABEL);
    eventTitleField.setWidth("100%");
    eventTitleField.setInputPrompt(ElementNames.EVENT_VIEW_NAME_FIELD_PROMPT);
    eventTitleField.setRequired(true);

    eventDescriptionArea = new TextArea(ElementNames.EVENT_VIEW_DESCRIPTION_AREA_LABEL);
    eventDescriptionArea.setWidth("100%");
    eventDescriptionArea.setInputPrompt(ElementNames.EVENT_VIEW_DESCRIPTION_AREA_PROMPT);
    eventDescriptionArea.setRequired(true);

    eventTypeSelect = new Select(ElementNames.EVENT_VIEW_TYPE_SELECT_LABEL);
    eventTypeSelect.setWidth("100%");
    eventTypeSelect.setRequired(true);
    eventTypeSelect.setNullSelectionAllowed(false);
    eventTypeContainer = view.getConnection().getSQLContainerFromDBTable(PaattiTableNames.EVENTTYPE,
        PaattiColumnNames.EVENTTYPE_eventTypeID, true);
    eventTypeSelect.setContainerDataSource(eventTypeContainer);
    eventTypeSelect.setItemCaptionPropertyId(PaattiColumnNames.EVENTTYPE_description);

    eventEstimatedTimeField = new TextField(ElementNames.EVENT_VIEW_ESTIMATEDTIME_FIELD_LABEL);
    eventEstimatedTimeField.setRequired(true);

    saveButton = new Button(ElementNames.EVENT_VIEW_SAVE_BUTTON_LABEL);
    saveButton.addListener(this);

    removeButton = new Button(ElementNames.EVENT_VIEW_REMOVE_BUTTON_LABEL);
    removeButton.addListener(this);

    HorizontalLayout typeAndTimeLayout = new HorizontalLayout();

    typeAndTimeLayout.setSpacing(true);
    typeAndTimeLayout.addComponent(eventTypeSelect);
    typeAndTimeLayout.addComponent(eventEstimatedTimeField);

    VerticalLayout layout = new VerticalLayout();
```

```
layout.setMargin(true);
layout.setSpacing(true);
layout.addComponent(image);
layout.addComponent(eventTitleField);
layout.addComponent(eventDescriptionArea);
layout.addComponent(typeAndTimeLayout);
layout.addComponent(saveButton);
layout.addComponent(removeButton);

setCompositionRoot(layout);
}
/**
 * {@inheritDoc}
 *
 * @param event event db Item.
 */
@Override
protected void fillTab(Item event) {
    eventTitleField.setValue(event.getItemProperty(PaattiColumnNames.EVENT_title).getValue());
    eventDescriptionArea.setValue(event.getItemProperty(PaattiColumnNames.EVENT_description).getValue());
    eventEstimatedTimeField.setValue(event.getItemProperty(PaattiColumnNames.EVENT_estimatedTime).getValue());
    for (Object itemId : eventTypeSelect.getItemIds()) {
        if (itemId.toString().equals(
            event.getItemProperty(PaattiColumnNames.EVENT_EVENTTYPE_eventTypeID).toString())) {
            eventTypeSelect.select(itemId);
        }
    }
    removeButton.setEnabled(true);
}
/**
 * {@inheritDoc}
 */
@Override
protected void clearTab() {
    eventTitleField.setValue("");
    eventDescriptionArea.setValue("");
    eventEstimatedTimeField.setValue("");
    eventTypeSelect.setValue(null);
    removeButton.setEnabled(false);
}
/**
 * {@inheritDoc}
 * <p/>
```

```
* The handled button click events are:
* <ul>
* <li>save the selected event and</li>
* <li>remove the selected event.</li>
* </ul>
*/
@Override
public void buttonClick(ClickEvent event) {
    String buttonCaption = event.getButton().getCaption();
    String msg = null;

    if (buttonCaption.equals(ElementNames.EVENT_VIEW_SAVE_BUTTON_LABEL)) {
        if (eventTitleField.isValid() && eventDescriptionArea.isValid() && eventTypeSelect.isValid()
            && eventEstimatedTimeField.isValid()) {
            if (eventTool.isUnfinished()) {
                msg = eventTool.getWarningMessage();
            } else {
                String title = eventTitleField.getValue().toString().trim();
                String description = eventDescriptionArea.getValue().toString().trim();
                Object eventId = eventTypeSelect.getValue();

                // TODO: Check the user input!
                Integer estimatedTime = StringHelper.parseStringToInteger(eventEstimatedTimeField.getValue().toString());

                estimatedTime = (estimatedTime == null) ? 0 : estimatedTime; // if the input was not numbers, set the time to zero.

                Object rowId = eventTool.saveEventTree(view.getSelectedItemId(), title, description, eventType,
                    estimatedTime);

                msg = (rowId != null) ? ElementNames.UPDATE_OK : ElementNames.UPDATE_FAIL;
                view.refresh(rowId);
            }
        } else {
            msg = ElementNames.SOMETHING_REQUIRED;
        }
    } else if (buttonCaption.equals(ElementNames.EVENT_VIEW_REMOVE_BUTTON_LABEL)) {
        if (view.getConnection().removeRowFromTable(view.getSelectedItemId(), PaattiTableNames.EVENT,
            PaattiColumnNames.EVENT_eventID)) {
            msg = ElementNames.UPDATE_OK;
            view.refresh(null);
            view.setTabsEnabled(false);
        } else {
            msg = ElementNames.UPDATE_FAIL;
        }
    }
}
```

```
if (msg != null) {  
    getWindow().showNotification(msg);  
}  
}  
}
```