

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

import com.vaadin.data.Item;
import com.vaadin.ui.CustomComponent;
import com.vaadin.ui.GridLayout;
import com.vaadin.ui.Label;
import com.vaadin.ui.TabSheet;
import fi.paatti.paattidatabasutils.dbsevice.PaattiResearchDBService;
import fi.paatti.research.paattiapplication.components.SearchComponent;

/**
 * The base class for the views of the application.
 */
 * @author Tapio Keränen, t.tapio.keranen@student.jyu.fi
 * @author Toni Salminen, toni.a.j.salminen@student.jyu.fi
 */
public abstract class ApplicationView extends CustomComponent {
    protected ApplicationTabSheet applicationTabSheet;
    private SearchComponent searchComponent;
    private TabSheet tabSheet;
    private Item selectedItem;
    private String tableColumnId;
    private Label selectedItemInfoLabel;

    /**
     * Constructor of the class.
     *
     * @param applicationTabSheet the tabsheet that contains the view.
     * @param tableName the name of the db table related to the view.
     * @param tableColumnId the name of the id column of the related table.
     */
    public ApplicationView(ApplicationTabSheet applicationTabSheet, String tableName, String tableColumnId) {
        this.applicationTabSheet = applicationTabSheet;
        this.tableColumnId = tableColumnId;

        searchComponent = new SearchComponent(this, tableName, tableColumnId);

        tabSheet = new TabSheet();
        tabSheet.setSizeFull();
        tabSheet.setEnabled(false);

        selectedItemInfoLabel = new Label();
        selectedItemInfoLabel.setCaption("");
    }
}
```

```
GridLayout grid = new GridLayout(4, 21);

grid.setSpacing(true);
grid.setMargin(true);
grid.setSizeFull();

grid.addComponent(searchComponent, 0, 0, 0, 20);
grid.addComponent(selectedItemInfoLabel, 1, 0, 3, 0);
grid.addComponent(tabSheet, 1, 1, 3, 20);

setCompositionRoot(grid);
}

/**
 * Adds a tab to the applicationView's tab layout.
 *
 * @param tab the CustomComponent to be added as a tab.
 * @param caption the caption of the tab.
 */
public void addTab(CustomComponent tab, String caption) {
    tabSheet.addTab(tab, caption);
}

/**
 * Sets the item for the applicationViewTab.
 *
 * @param item SQLContainer item.
 */
public void setContentts(Item item) {
    selectedItem = item;
    setTabsEnabled(true);
}

/**
 * Saves the view's information.
 */
public abstract void saveContentts();

/**
 * Enables or disables all the tabs that have been added to the view's tab
 * sheet.
 *
 * @param enabled the enabled status.
 */
public void setTabsEnabled(boolean enabled) {
    tabSheet.setEnabled(enabled);
}
```

```
if (!enabled) {
    for (int i = 0; i < tabSheet.getComponentCount(); i++) {
        ApplicationViewTab tab = (ApplicationViewTab) tabSheet.getTab(i).getComponent();

        tab.clearTab();
    }
}

/**
 * Refreshes the view's search component and selects the row that matches
 * the given itemId.
 *
 * @param itemId the itemId.
 */
public void refresh(Object itemId) {
    searchComponent.refresh(itemId);
}

/**
 * Activates the tab at given tabIndex and refreshes its content using the
 * given itemId.
 */
public void setActiveTab(int tabIndex, Object itemId) {
    applicationTabSheet.setActiveTab(tabIndex, itemId);
}

/**
 * Returns the ID of the currently selected item.
 *
 * @return itemId if selectedItem not null; otherwise null.
 */
public String getSelectedItemId() {
    return (selectedItem != null) ? selectedItem.getItemProperty(tableColumnId).toString() : null;
}

/**
 * Returns the database connection.
 *
 * @return PaattiResearchDBConnection connection.
 */
public PaattiResearchDBService getDBConnection() {
    return applicationTabSheet.getDBConnection();
}
```

```
/**
 * Sets the caption for the label which identifies the current selection.
 *
 * @param itemDescription the description of the selected item.
 */
public void setSelectedInfoLabel(String itemDescription) {
    selectedItemInfoLabel.setCaption(itemDescription);
}

/**
 * Creates the basic information string from the given item.
 *
 * @param item db item.
 * @return Created string.
 */
protected abstract String createInfoString(Item item);
}
```