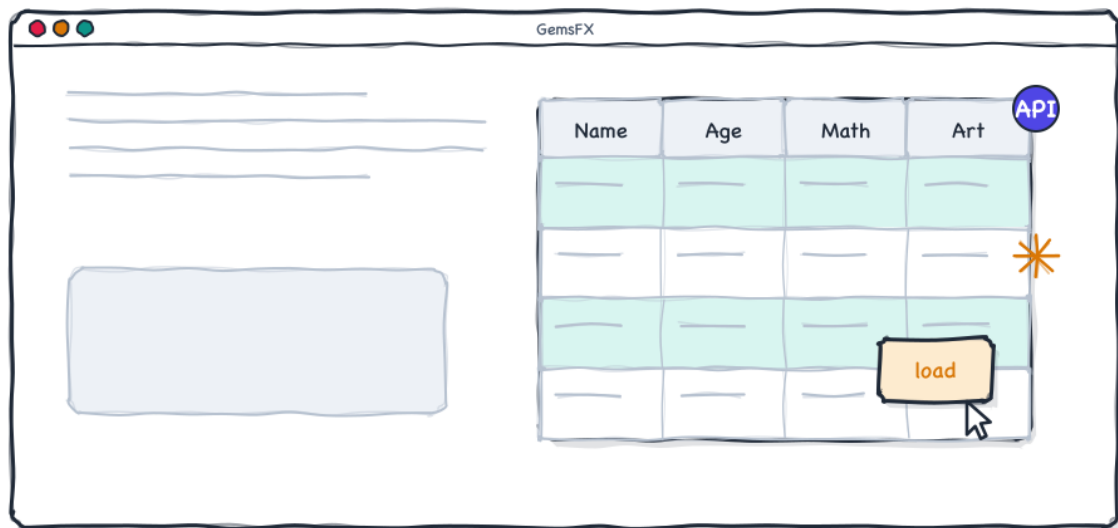


GridTableView

Developer Manual

GemsFX • `com.dlsc.gemsfx.gridtable` • JavaFX Custom Controls



Generated cartoon overview of GridTableView.

GridTableView renders rows and columns with GridPane rather than VirtualFlow, making it useful for compact tables with custom row headers, footers and loading overlays.

Table of Contents

1. Introduction	3
1.1 Key features	3
1.2 Maven dependency	3
2. Getting started	3
3. Anatomy	4
4. Control API	5
4.1 Rows and columns	5
4.2 Factories and callbacks	5
4.3 Loading	5
4.4 GridTableColumn and cell	5
5. Behaviour	6
5.1 Rebuilding the grid	6
5.2 Rows, headers and footers	6
5.3 Cell creation	7
6. Styling	7
6.1 Style classes	7
6.2 Pseudo classes	8
6.3 Styleable CSS properties	8
7. Localization	8
8. Accessibility	9
9. Recipes	9
9.1 Add row headers	9
9.2 Show loading overlay	9
9.3 Open on double click	9
9.4 Context menu per row	9
9.5 Custom cell	9
9.6 Checklist	10

10. See also 10

1. Introduction

GridTableView lives in `com.dlsc.gemsfx.gridtable`. Rows come from items, columns are `GridTableColumn` objects, and cells are `GridTableCell` instances created by each column.

1.1 Key features

- GridPane layout with column headers and row backgrounds.
- Optional row header and row footer factories per item.
- Placeholder shown when columns exist but items is empty.
- LoadingPane overlay with status, size, progress indicator and commit delay.
- Double-click open callback and context menu callback for rows.

1.2 Maven dependency

```
<dependency>
  <groupId>com.dlsc.gemsfx</groupId>
  <artifactId>gemsfx</artifactId>
  <version>4.4.1</version>
</dependency>
```

Use package `com.dlsc.gemsfx.gridtable`.

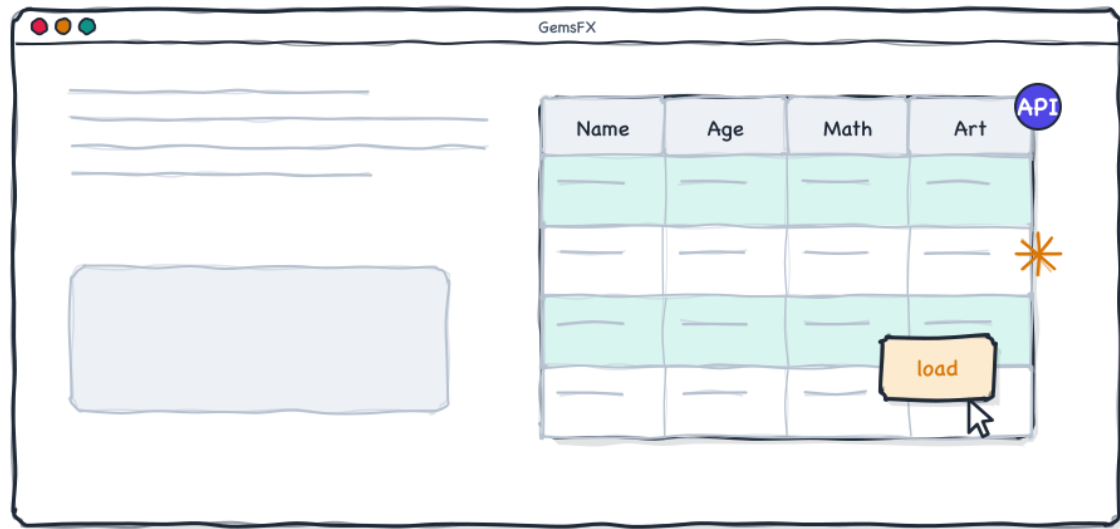
2. Getting started

The snippet below uses only APIs verified in the source and demo code.

```
GridTableView<Student> table = new GridTableView<>();
GridTableColumn<Student, String> name = new GridTableColumn<>("Name");
name.setCellValueFactory(new GridTablePropertyValueFactory<>("name"));
GridTableColumn<Student, Integer> age = new GridTableColumn<>("Age");
age.setCellValueFactory(new GridTablePropertyValueFactory<>("age"));

name.setPercentWidth(70);
age.setPercentWidth(30);
table.getColumns().setAll(name, age);
table.getItems().setAll(new Student("Tom", 12), new Student("Lucy", 11));
```

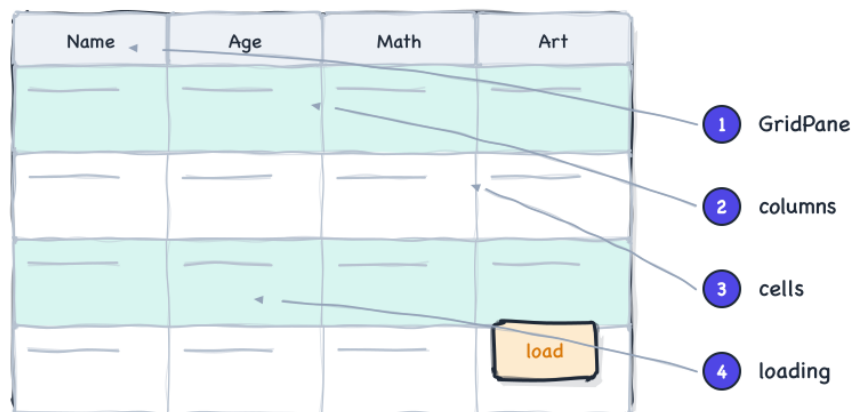
Minimal setup for GridTableView.



A first look at the control.

3. Anatomy

The diagram and table identify the nodes, model objects and style classes that matter when using or styling the control.



The main parts of the control.

Part	Type / style	Description
GridTableView	grid-table-view	Control root and stylesheet owner.
GridPane	grid-pane	Skin layout containing headers, body rows and LoadingPane.
GridTableColumn	ColumnConstraints	Header, width constraints, value factory and cell factory.
GridTableCell	grid-table-cell	Cell for one row/column value.

Part	Type / style	Description
LoadingPane	loading-pane	Overlay shown while loadingStatus is not OK.

4. Control API

4.1 Rows and columns

Property	Type	Default	Description
items	ListProperty<S>	empty list	Rows displayed in the table body.
columns	ListProperty<GridTableColumn<S, ?>>	empty list	Column definitions and GridPane column constraints.
minNumberOfRows	IntegerProperty	0	Minimum number of body rows; extra rows are empty.
placeholder	ObjectProperty<Node>	Label "No items"	Shown when columns exist but items is empty.

4.2 Factories and callbacks

Property	Type	Default	Description
rowHeaderFactory	ObjectProperty<Callback<S, Node>>	null	Creates an optional row header above the cells for an item.
rowFooterFactory	ObjectProperty<Callback<S, Node>>	null	Creates an optional row footer below the cells for an item.
onOpenItem	ObjectProperty<Consumer<S>>	null	Called on primary double-click when set.
onContextMenuForItemRequested	ObjectProperty<Callback<S, ContextMenu>>	null	Creates a context menu for a row item.

4.3 Loading

Property	Type	Default	Description
loadingStatus	ObjectProperty<LoadingPane.Status>	OK	Controls the LoadingPane overlay.
loadingStatusSize	ObjectProperty<LoadingPane.Size>	MEDIUM	Size of the loading status indicator.
progressIndicator	ObjectProperty<ProgressIndicator>	CircleProgressIndicator	Indicator used by the LoadingPane.
commitLoadStatusDelay	LongProperty	400	Delay in milliseconds before committed loading status is shown.

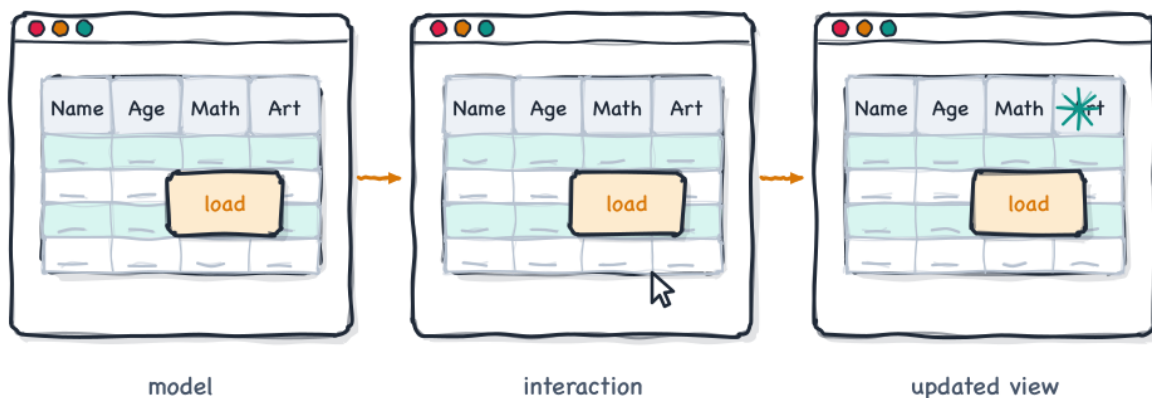
4.4 GridTableColumn and cell

Property	Type	Default	Description
text	StringProperty	"Header" default for no-arg column	Header text.
graphic	ObjectProperty<Node>	null	Header graphic.
contentDisplay	ObjectProperty<ContentDisplay>	LEFT / constructor-specific	Header graphic/text placement.
converter	ObjectProperty<StringConverter<T>>	SimpleStringConverter	Converts cell values to text.
cellValueFactory	ObjectProperty<Callback<S,T>>	null	Extracts the cell value from a row item.
cellFactory	ObjectProperty<Callback<GridTableView<S>, GridTableCell<S,T>>>	new GridTableCell	Creates cells.
transparent	StyleableBooleanProperty	false	GridTableCell CSS bridge for mouseTransparent.

5. Behaviour

5.1 Rebuilding the grid

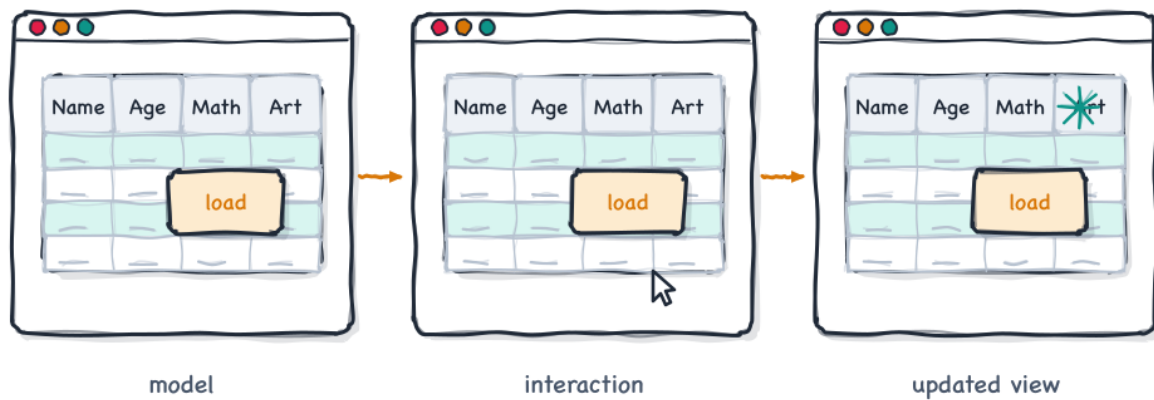
The skin clears and recreates the GridPane whenever items, columns, row header/footer factories or the refresh-items marker changes.



The main runtime behaviour.

5.2 Rows, headers and footers

Each item can produce a row header and footer. Row backgrounds span the cell row plus header/footer rows so striping remains continuous.



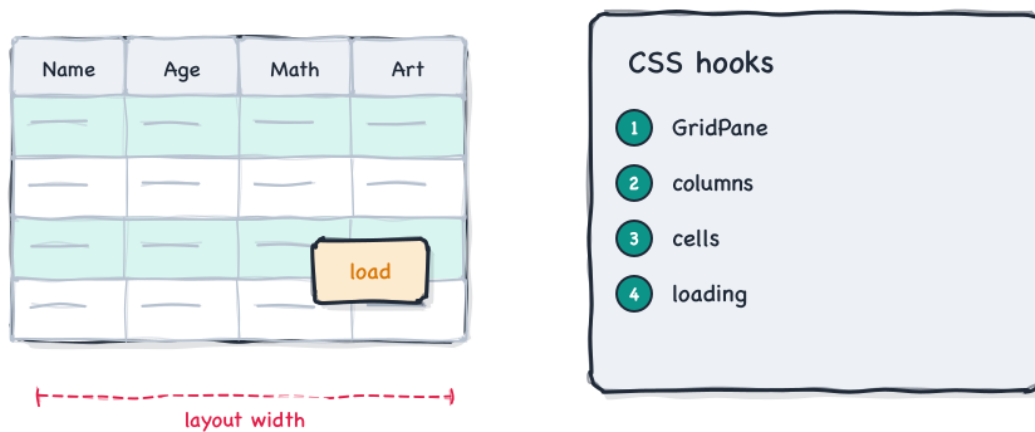
Data and interaction flow.

5.3 Cell creation

Each column creates a cell by reading the row item, applying its `cellValueFactory`, calling its `cellFactory`, setting `rowItem/index/column` and installing context-menu support.

6. Styling

The style hooks below were verified in the control, skin and CSS sources.



Style hooks and visual states.

6.1 Style classes

Style class	Where used
grid-table-view	Root control style class.
grid-pane	Skin GridPane.

Style class	Where used
column-header column-header-background	Header nodes and background.
row-background odd even first middle last only	Row background regions.
grid-table-cell first middle last row-first row-middle row-last col-index-N row-index-N	Cells created by columns.
loading-pane error-pane error-label progress-indicator-wrapper	LoadingPane overlay classes.

6.2 Pseudo classes

Pseudo class	Meaning
focused	Root focus styling.
hover	Applied to row background while a cell is hovered.
odd / even	GridTableCell pseudo classes based on row index.

6.3 Styleable CSS properties

Property	Type	Default	Description
-fx-mouse-transparent	Boolean	false	Styleable property on GridTableCell only.

```
.grid-table-view > .grid-pane > .grid-table-cell {
    -fx-padding: 8px;
}
.grid-table-view > .grid-pane > .grid-table-cell:even {
    -fx-background-color: -fx-control-inner-background-alt;
}
```

Example CSS.

7. Localization

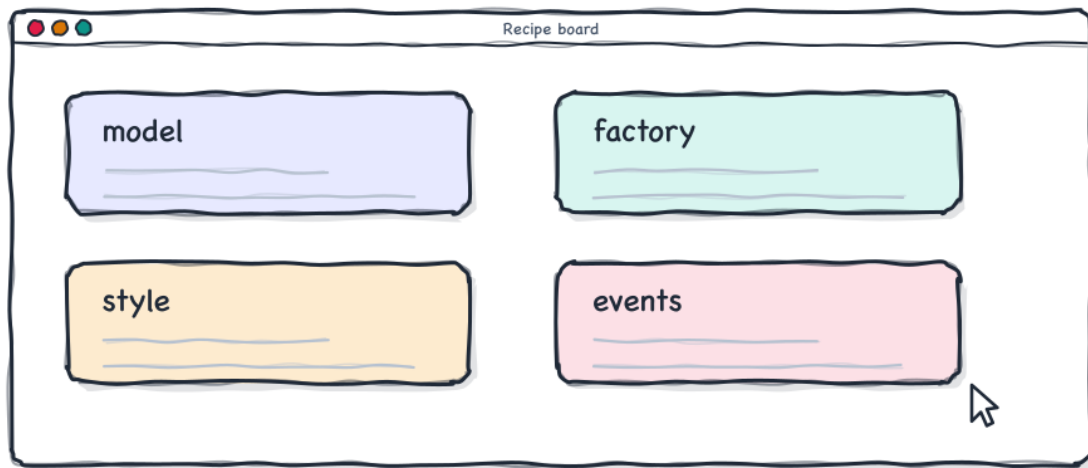
The following keys are read via ResourceBundleManager.

Key	English default
placeholder.no-items	No items

8. Accessibility

GridTableView sets AccessibleRole.TABLE_VIEW and requests focus on primary single-click.

9. Recipes



Common configuration recipes.

9.1 Add row headers

```
table.setRowHeaderFactory(student -> new Label("Header for " + student.getName()));
```

9.2 Show loading overlay

```
table.setLoadingStatus>LoadingPane.Status.LOADING);  
table.setCommitLoadStatusDelay(400);
```

9.3 Open on double click

```
table.setOnOpenItem(student -> openEditor(student));
```

9.4 Context menu per row

```
table.setOnContextMenuForItemRequested(student -> new ContextMenu(new MenuItem("Edit")));
```

9.5 Custom cell

```
ageColumn.setCellFactory(view -> new GridTableCell<>());
```

9.6 Checklist

- 1 Create columns before expecting placeholder/body rows.
- 2 Use `minNumberOfRows` for fixed-height compact tables.
- 3 Call `refresh()` after changes that do not update observable lists.
- 4 `GridTablePropertyValueFactory` reads fields by reflection, not JavaFX properties.

10. See also

Demo app: `GridTableViewApp`. Run it with:

```
mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.GridTableViewApp
```

- Related GemsFX controls: `AdvancedTableView`, `MultiColumnListView`, `TableView`.
- API documentation: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>