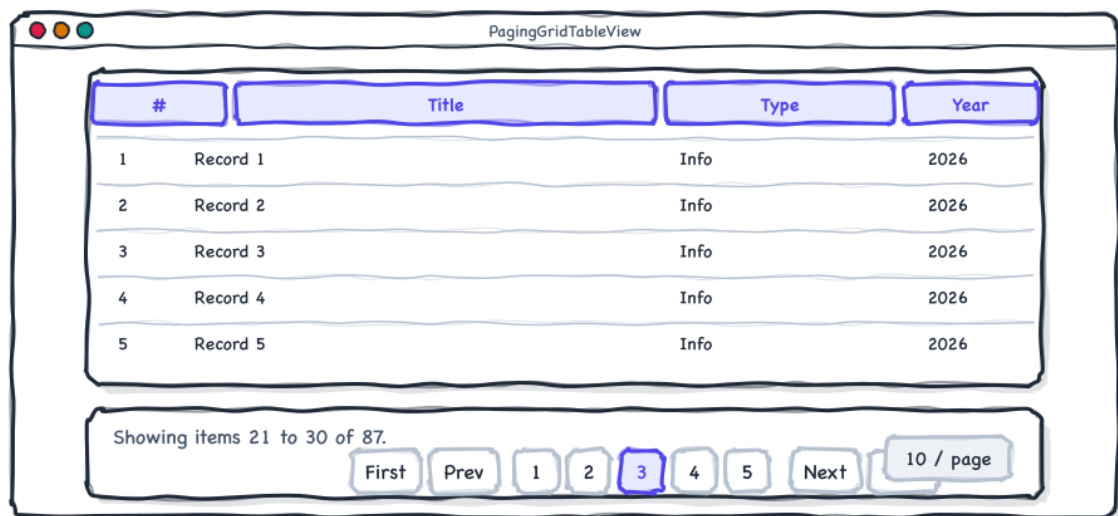


PagingGridTableView

Developer Manual

GemsFX • com.dlsc.gemsfx.paging • JavaFX Custom Controls



A GridTableView receives one loaded page and a synchronized PagingControls bar.

PagingGridTableView applies the shared item-paging loader contract to a GridTableView. It exposes a columns list, current page items, loading status, paging controls and open-item hooks for table-like page navigation.

Table of Contents

1. Introduction	2
1.1 Key features	2
1.2 Maven dependency	2
2. Getting started	2
3. Anatomy	3
4. Control API	4
4.1 Shared item paging properties	4
4.2 Grid-table properties	4
5. Loader contract	5
6. Columns and rows	5
7. Refresh and opening items	6
8. Styling	6
9. Localization	7
10. Accessibility	7
11. Recipes	7
11.1 Fill the last page	7
11.2 Move controls to the top	7
11.3 Open row details	7
11.4 Reload after filter changes	7
11.5 Checklist	7
12. See also	8

1. Introduction

PagingGridTableView is the table counterpart of **PagingListView**. It embeds a **GridTableView**, binds its rows to `itemsOnCurrentPage`, and forwards columns, loading status, placeholder and open-item callbacks to the inner grid table.

1.1 Key features

- Shared asynchronous paging loader.
- Observable columns list bound into the inner **GridTableView**.
- Optional filling of the last page by binding the minimum row count to `pageSize`.
- Embedded **PagingControls** at top or bottom.
- `reload()` for data loads and `refresh()` for row refreshes.

1.2 Maven dependency

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

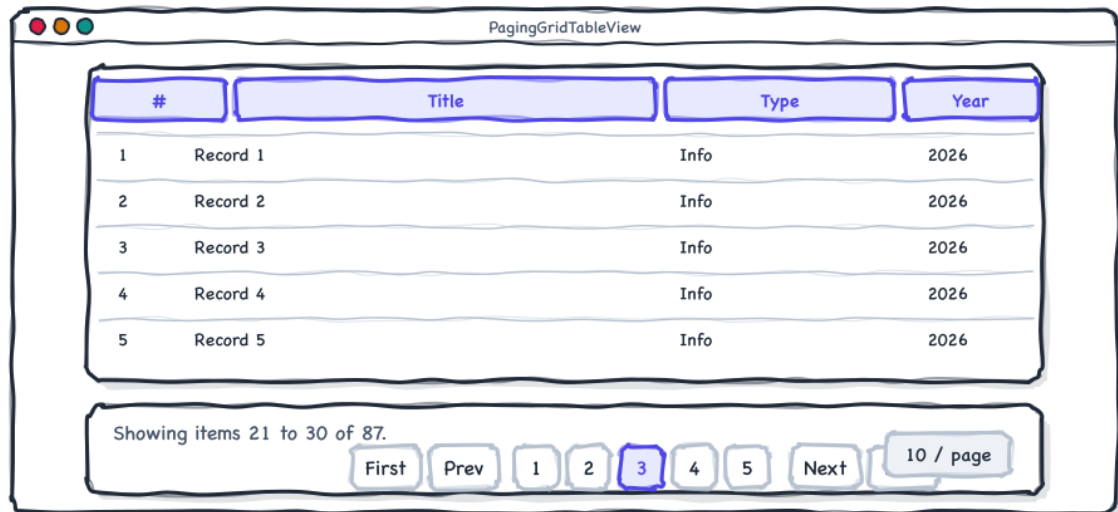
The class is in `com.dlsc.gemsfx.paging`.

2. Getting started

```
PagingGridTableView<Person> table = new PagingGridTableView<>();
table.getColumns().setAll(firstNameColumn, lastNameColumn);

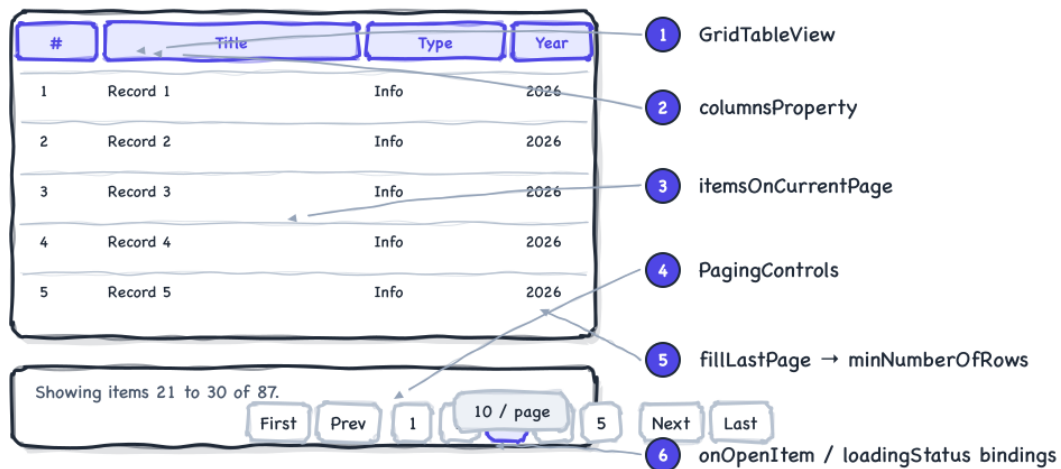
table.setLoader(request -> {
    int offset = request.getPage() * request.getPageSize();
    return new PagingLoadResponse<>(repository.find(offset, request.getPageSize()),
                                    repository.count());
});
```

A minimal paged grid table.



PagingGridTableView shows a grid table and a paging bar.

3. Anatomy



The parts of PagingGridTableView.

Part	Style class	Description
.paging-grid-table-view	Root style class.	
.content	Container for grid table and paging controls.	
GridTableView classes	The inner grid table keeps its own table-specific style classes.	

4. Control API

4.1 Shared item paging properties

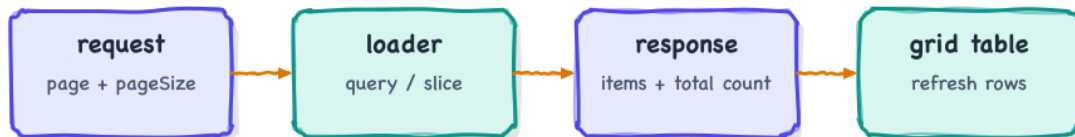
Property	Type	Default	Description
loader	ObjectProperty<Callback<PagingLoadRequest, PagingLoadResponse<T>>>	LoadingService default callback	Callback that receives zero-based page and pageSize and returns items plus the total count.
itemsOnCurrentPage	ReadOnlyListProperty<T>	empty list	Items returned by the latest successful load and shown by the inner view.
loadingStatus	ReadOnlyObjectProperty<LoadingStatus>	OK	OK, LOADING or ERROR based on the internal LoadingService state.
loadDelayInMillis	LongProperty	50	Delay before the control starts a load. Negative values are rejected.
commitLoadStatusDelay	LongProperty	400	Delay before the loading status is committed to the UI.
fillLastPage	BooleanProperty	false	Pads the last page layout where supported. Styleable as -fx-fill-last-page.
pagingControlsLocation	ObjectProperty<Side>	Side.BOTTOM	Places controls at TOP or BOTTOM. LEFT and RIGHT are not supported and reset to BOTTOM.
showPagingControls	BooleanProperty	true	Shows or hides embedded PagingControls. Styleable as -fx-show-paging-controls.
usingScrollPane	BooleanProperty	false	Wraps the content in a ScrollPane when true. Styleable as -fx-use-scroll-pane.
placeholder	ObjectProperty<Node>	Label("No items")	Node shown when the current page has no content.
onOpenItem	ObjectProperty<Consumer<T>>	null	Callback invoked by the inner item view when an item is opened.
selectionModel	ObjectProperty<MultipleSelectionModel<T>>	control-specific model	Selection model of the inner list or grid table.

4.2 Grid-table properties

Property	Type	Default	Description
columns	ObservableList<GridTableColumn<T, ?>>	empty list	Columns bound into the internal GridTableView.
selectionModel	ObjectProperty<MultipleSelectionModel<T>>	inner GridTableView model	Selection model used by the inner grid table.

5. Loader contract

The data-loading contract is identical to PagingListView. A request contains page and pageSize; a response returns the page items and the total count. The service updates loadingStatus and itemsOnCurrentPage after a successful callback.

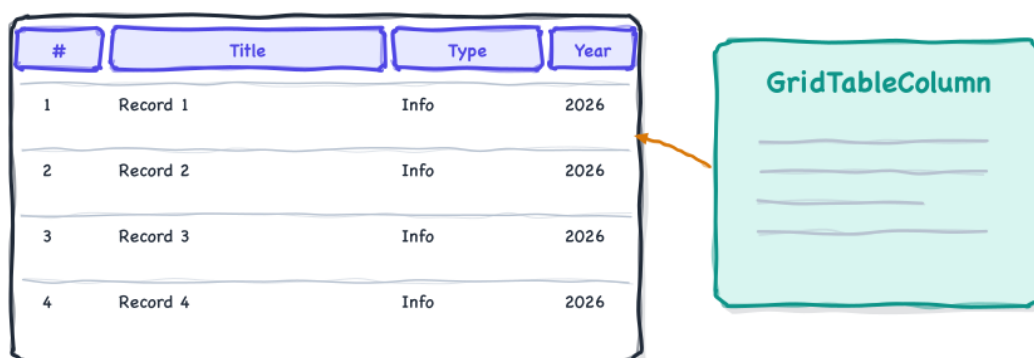


Request and response flow for a grid table page.

Note. The loader callback should not return null items. Use `PagingLoadResponse.emptyResponse()` for empty results.

6. Columns and rows

The control exposes the inner table columns through `getColumns()`. Set them once, then update the loader independently.



Columns are bound into the internal GridTableView.

#	Title	Type	Year
1	Record 1	Info	2026
2	Record 2	Info	2026
3	Record 3	Info	2026

`fillLastPage = false`

#	Title	Type	Year
1	Record 1	Info	2026
2	Record 2	Info	2026
3	Record 3	Info	2026
4	Record 4	Info	2026
5	Record 5	Info	2026

`fillLastPage = true`

fillLastPage controls whether the last page keeps the same row footprint.

7. Refresh and opening items

`refresh()` delegates to the inner `GridTableView` `refresh`. `reload()` restarts the `LoadingService`. `onOpenItem` is forwarded to the grid table for item activation.

8. Styling

Style class	Description
<code>.paging-grid-table-view</code>	Root style class.
<code>.content</code>	Container for grid table and paging controls.
GridTableView classes	The inner grid table keeps its own table-specific style classes.

CSS property	Type	Default
<code>-fx-fill-last-page</code>	boolean	false
<code>-fx-paging-controls-location</code>	Side	BOTTOM
<code>-fx-show-paging-controls</code>	boolean	true
<code>-fx-use-scroll-pane</code>	boolean	false
<code>-fx-show-page-size-selector</code>	boolean	true
<code>-fx-same-width-page-buttons</code>	boolean	false
<code>-fx-show-previous-next-page-button</code>	boolean	true
<code>-fx-message-label-strategy</code>	enum	ALWAYS_SHOW
<code>-fx-first-last-page-display-mode</code>	enum	SHOW_PAGE_BUTTONS
<code>-fx-max-page-indicators-count</code>	integer	5
<code>-fx-page-alignment</code>	HPos	RIGHT

```
.paging-grid-table-view {  
  -fx-fill-last-page: true;  
  -fx-paging-controls-location: BOTTOM;  
}  
.paging-grid-table-view > .stack-pane > .content {  
  -fx-spacing: 8px;  
}
```

Example CSS for PagingGridTableView.

9. Localization

The embedded PagingControls use the paging-control bundle. The item-paging placeholder uses the English default *No items*.

10. Accessibility

PagingGridTableView sets `AccessibleRole.TABLE_VIEW`. The inner grid table provides row and column semantics while the embedded pager remains keyboard-operable.

11. Recipes

11.1 Fill the last page

```
table.setFillLastPage(true);
```

11.2 Move controls to the top

```
table.setPagingControlsLocation(Side.TOP);
```

11.3 Open row details

```
table.setOnOpenItem(person -> detailsPane.setPerson(person));
```

11.4 Reload after filter changes

```
filter.textProperty().addListener(obs -> table.reload());
```

11.5 Checklist

- 1 Return `totalItemCount` for the unpaginated query.
- 2 Keep columns independent from the loader.
- 3 Use `fillLastPage` only when a stable table height is desired.
- 4 Use `refresh()` for visible row repainting, not for data queries.

12. See also

- Demo app: `PagingGridTableViewApp`. Run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.PagingGridTableViewApp`.
- Related controls: `PagingControls`, `PagingListView` and `GridTableView`.
- API docs: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>