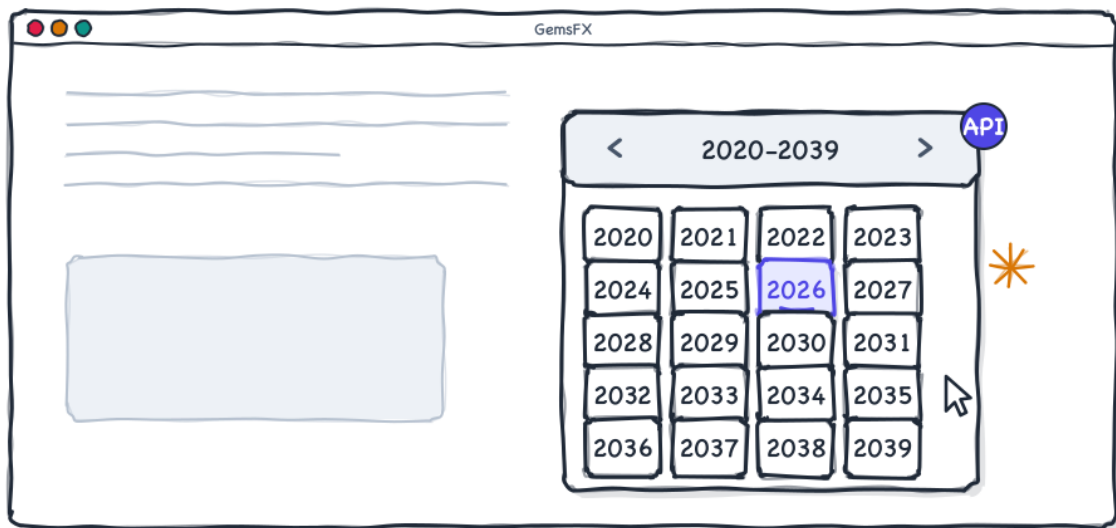


YearView

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



Generated cartoon overview of YearView.

YearView displays a configurable grid of years, marks the current and selected year, and pages through year ranges with arrow buttons.

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 Selection	4
4.2 Grid	4
5. Behaviour	4
5.1 Paged ranges	4
5.2 Range constraints	5
6. Styling	5
6.1 Style classes	5
6.2 Pseudo classes	6
6.3 Styleable CSS properties	6
7. Localization	6
8. Accessibility	7
9. Recipes	7
9.1 Show twelve years per page	7
9.2 Restrict to a product lifetime	7
9.3 Mirror the integer year	7
9.4 Clear the selected value	7
9.5 Checklist	7
10. See also	8

1. Introduction

YearView is part of the GemsFX module `com.dlsc.gemsfx`. This manual documents only behavior verified in the control, skin, CSS, resource bundles and demo app.

1.1 Key features

- Selects a `java.time.Year` value, initialized to `Year.now()`.
- Read-only year property exposes the selected year as an `int` or `-1` when value is null.
- Rows and columns define how many years are shown per page.
- Optional `earliestYear` and `latestYear` disable out-of-range years.
- The skin shows a range label such as 2020-2039.

1.2 Maven dependency

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

Use package `com.dlsc.gemsfx`.

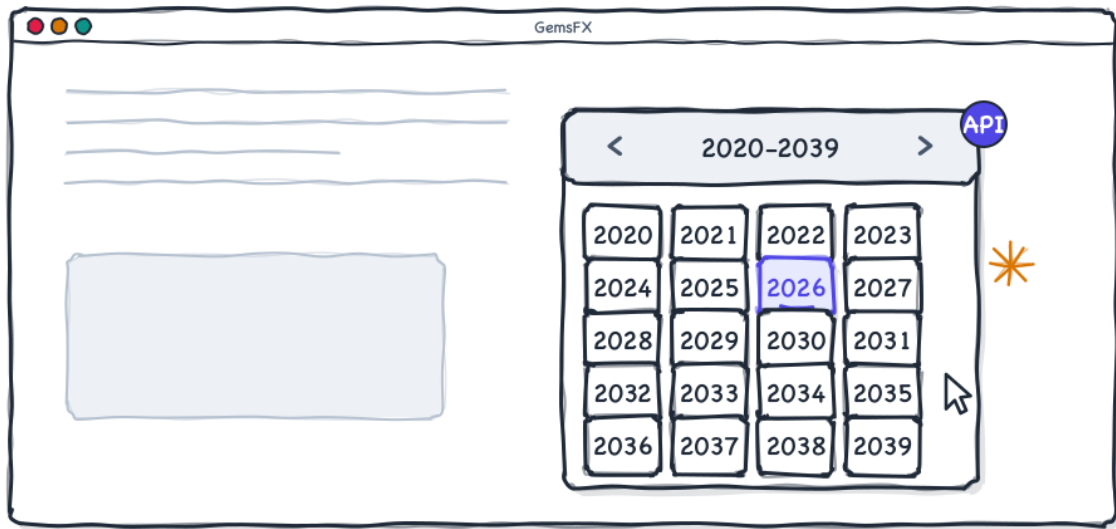
2. Getting started

Create the control, set the properties that define its valid values, and listen to the value or selection property that the control owns.

```
YearView view = new YearView();
view.setCols(4);
view.setRows(5);
view.setEarliestYear(Year.of(2000));
view.setLatestYear(Year.now().plusYears(5));

view.valueProperty().addListener((obs, oldYear, newYear) -> {
    System.out.println("selected year = " + newYear);
});
```

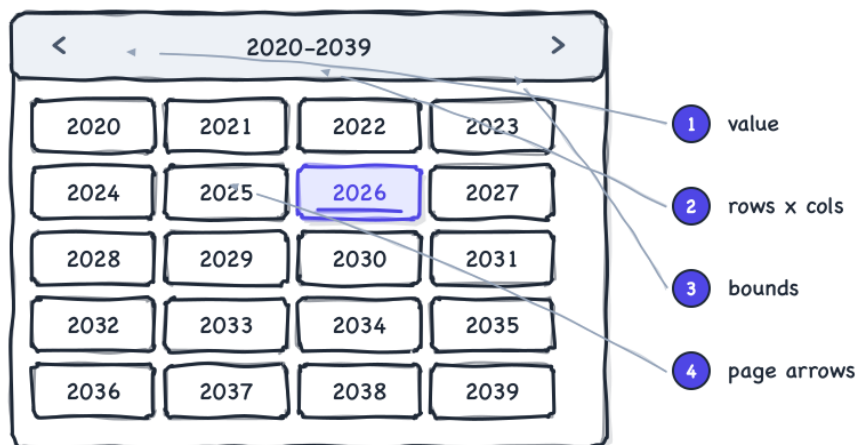
A compact setup for YearView.



A first look at the control in a simple application window.

3. Anatomy

The skin builds the visible nodes below the public control. The table lists the style classes and nodes that are useful when reading the source or writing CSS.



The main parts of the control.

Part	Style / node	Description
Root	year-view	Control root and stylesheet hook.
Header	header	Left button, year-range label and right button.
Range label	year-range-label	Shows first and last visible year on the current page.
Grid	grid-pane	GridPane with rows x cols equally sized cells.

Part	Style / node	Description
Year box	year-box	Selectable VBox for one year.
Selection indicator	selection-indicator	Small line shown when the box matches value/year.

4. Control API

4.1 Selection

Property	Type	Default	Description
value	ObjectProperty<Year>	Year.now()	Selected year.
year	ReadOnlyIntegerProperty	current year	Integer value derived from value, or -1 if value is null.
earliestYear	ObjectProperty<Year>	null	Earliest selectable year; earlier cells are disabled.
latestYear	ObjectProperty<Year>	null	Latest selectable year; later cells are disabled.

4.2 Grid

Property	Type	Default	Description
cols	IntegerProperty	4	Number of year columns. Must be at least 1. Styleable via -fx-cols.
rows	IntegerProperty	5	Number of year rows. Must be at least 1. Styleable via -fx-rows.

5. Behaviour

5.1 Paged ranges

The first visible year is calculated from the selected year or the current year, the visible cell count, and an internal page offset. Clicking arrows increments or decrements the offset.



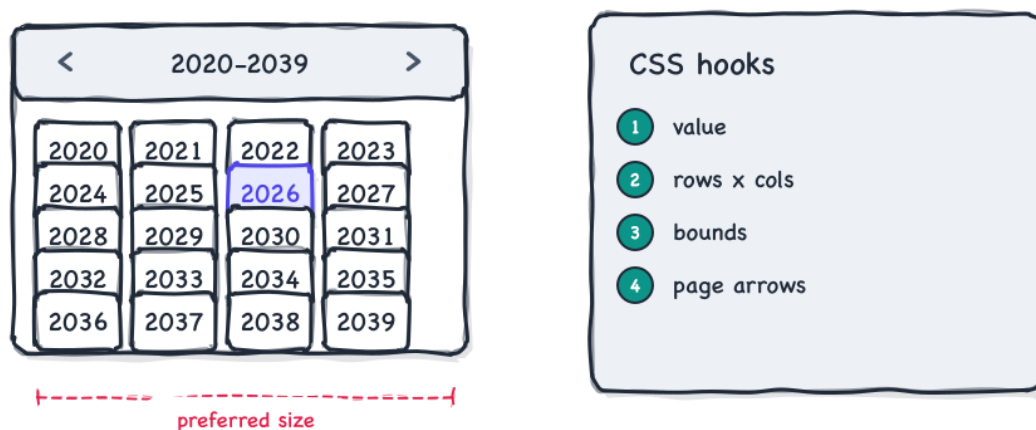
Interaction and value-flow behaviour.

5.2 Range constraints

Year boxes bind their disabled state to `earliestYear` and `latestYear`. Clicking a disabled box is prevented by JavaFX disabled-node behavior.

6. Styling

The stylesheet is loaded by the control or its base class. The rows below list style hooks verified in the CSS and skin source.



Style classes and pseudo-class states.

6.1 Style classes

Style class	Where used
year-view	Root control style class.
popover	Optional root class for popup-specific padding and border.

Style class	Where used
header	Navigation row.
year-range-label	Header label showing the visible range.
arrow-button left-button right-button	Navigation buttons.
arrow left-arrow right-arrow	Arrow glyph regions.
grid-pane	GridPane containing year boxes.
year-box	Selectable year cell.
year-label	Label for the year number.
selection-indicator	Indicator below selected year.

6.2 Pseudo classes

Pseudo class / marker	Meaning
focused	Border switches to -fx-accent.
selected	Style class, not pseudo class, added to selected year-box.
current	Style class, not pseudo class, added to current year-box.

6.3 Styleable CSS properties

Property	Type	Default	Description
-fx-cols	Integer	4	Number of columns; positive integers only.
-fx-rows	Integer	5	Number of rows; positive integers only.

```
.year-view {
    -fx-cols: 3;
    -fx-rows: 4;
}
.year-view > .grid-pane > .year-box > .selection-indicator {
    -fx-background-color: -fx-accent;
}
```

Example CSS.

7. Localization

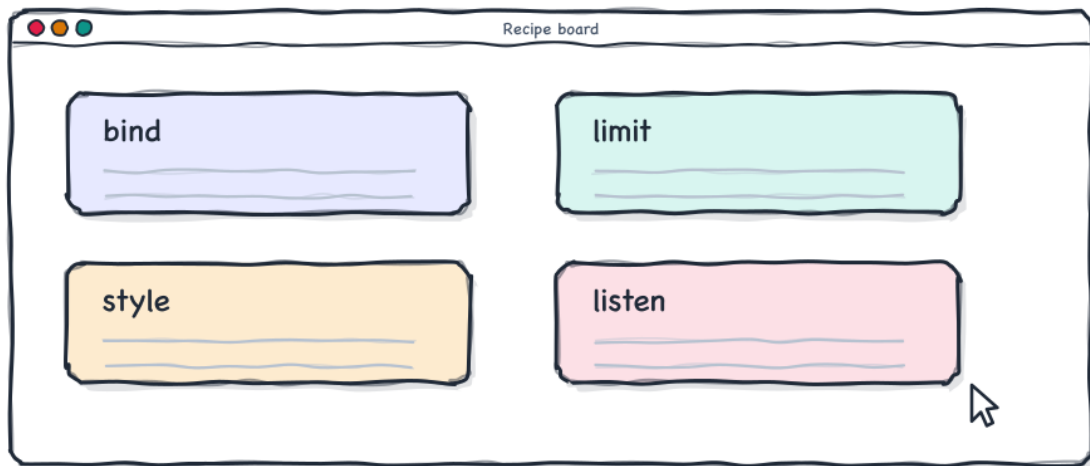
This control uses `ResourceBundleManager` for the following keys.

Key	English default
accessible.role-description	year selector

8. Accessibility

The constructor sets `AccessibleRole.DATE_PICKER` through `AccessibilityUtil.setRole` and uses the localized role description "year selector".

9. Recipes



Common configuration recipes.

9.1 Show twelve years per page

```
YearView view = new YearView();  
view.setCols(3);  
view.setRows(4);
```

9.2 Restrict to a product lifetime

```
view.setEarliestYear(Year.of(2020));  
view.setLatestYear(Year.of(2030));
```

9.3 Mirror the integer year

```
Label label = new Label();  
label.textProperty().bind(Bindings.convert(view.yearProperty()));
```

9.4 Clear the selected value

```
view.setValue(null); // yearProperty becomes -1
```

9.5 Checklist

- 1 Keep cols and rows ≥ 1 .

- 2 Use `earliestYear/latestYear` for selectable limits.
- 3 Remember that `selected/current` are style classes on cells.
- 4 Use the `value` property for Year objects and `yearProperty` for int display.

10. See also

Demo app: `YearViewApp`. Run it with:

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

- Related GemsFX controls: `YearPicker`, `YearMonthView`, `YearMonthPicker`.
- API documentation: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>