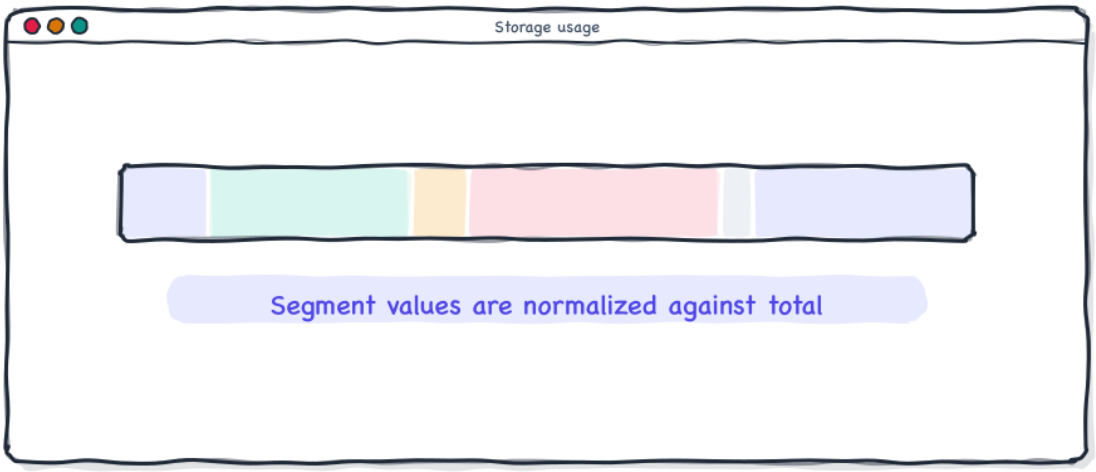
SegmentedBar

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



SegmentedBar lays out generic segments proportionally to their values.

SegmentedBar displays a list of valued segments as a proportional horizontal or vertical bar. Applications can use the default SegmentView or provide custom factories for typed segment subclasses.

Table of Contents

1. Introduction

SegmentedBar is generic over `T extends SegmentedBar.Segment`. Each segment has a non-negative value and optional text; the bar computes the total and lays out one node per segment.

1.1 Key features

- Generic segment type for domain-specific subclasses.
- Default SegmentView with label bound to segment.text.
- Custom segmentViewFactory for typed visuals and tooltips.
- Horizontal and vertical orientations.
- Minimum segment size so very small values remain visible.
- Position style classes for first, middle, last and only segments.

1.2 Maven dependency

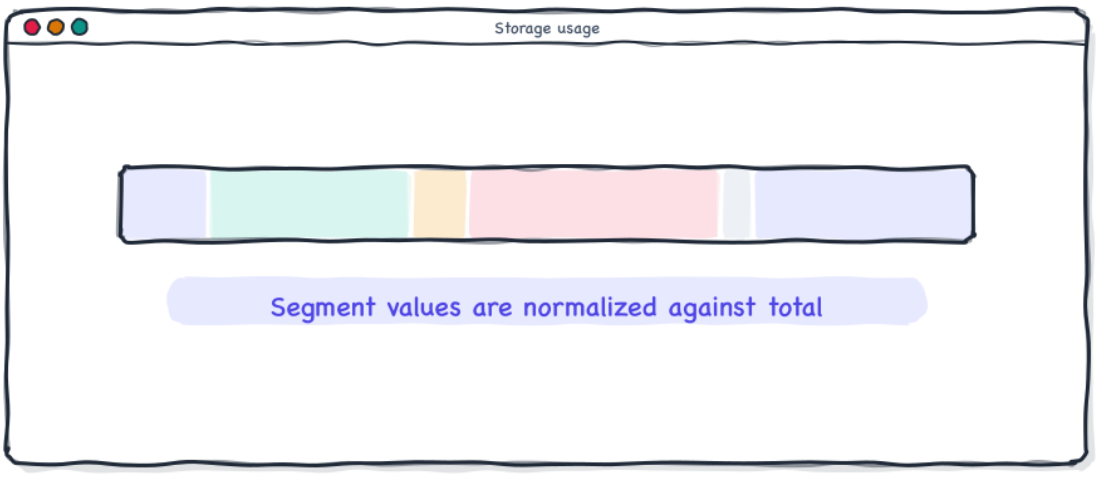
```
<dependency>
    <groupId>com.dlsc.gemsfx</groupId>
    <artifactId>gemsfx</artifactId>
    <version>4.4.1</version>
</dependency>
```

GemsFX controls live in module `com.dlsc.gemsfx`.

2. Getting started

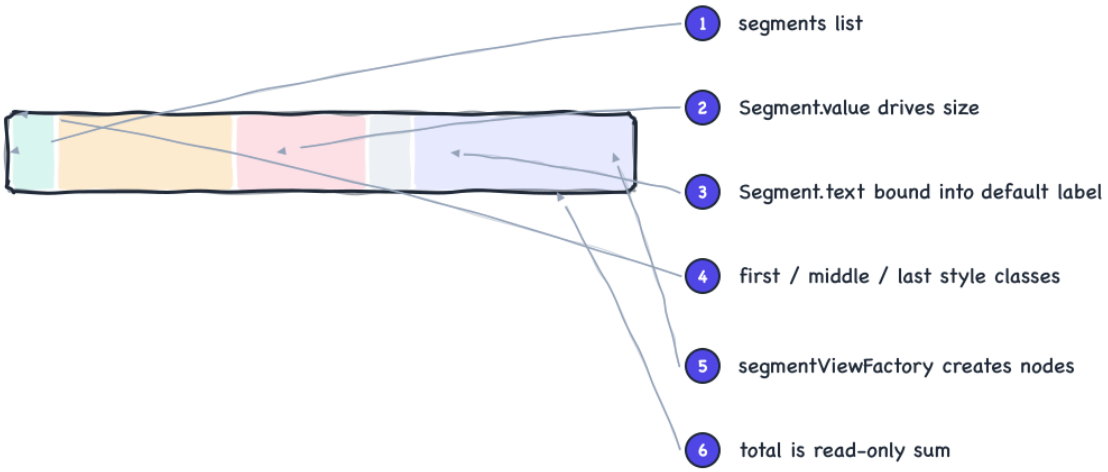
```
SegmentedBar<SegmentedBar.Segment> bar = new SegmentedBar<>();
bar.setOrientation(Orientation.HORIZONTAL);
bar.getSegments().addAll(
    new SegmentedBar.Segment(10, "10"),
    new SegmentedBar.Segment(40, "40"),
    new SegmentedBar.Segment(50, "50"));
```

A simple horizontal segmented bar.



A segmented storage bar with proportional segment widths.

3. Anatomy



The parts of SegmentedBar and its skin.

Style class / pseudo class	Description
.segmented-bar	Root control style class.
:horizontal / :vertical	Orientation pseudo classes.
.segment	Added by the skin to every segment node.
.segment-view	Default SegmentView style class.
.only-segment, .first-segment, .middle-segment, .last-segment	Position classes added by the skin.
.segment-view > .label	Default label inside SegmentView.

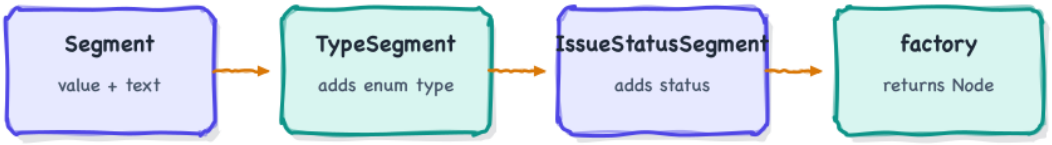
4. Control API

Property	Type	Default	Description
orientation	ObjectProperty<Orientation>	VERTICAL	HORIZONTAL or VERTICAL. Styleable as -fx-orientation; pseudo classes :horizontal and :vertical.
minSegmentSize	DoubleProperty	5	Minimum width or height for tiny segments. Styleable as -fx-min-segment-size.
segmentViewFactory	ObjectProperty<Callback<T, Node>>	SegmentView::new	Creates one node per segment.
segments	ListProperty<T>	empty list	Segments displayed by the bar.
total	ReadOnlyDoubleProperty	sum of segment values	Updated when segment values change.

4.1 Segment model

Property	Type	Default	Description
Segment.value	DoubleProperty	constructor value	Must be greater than or equal to zero; negative values throw.
Segment.text	StringProperty	null	Optional label text used by the default SegmentView.

5. Generics and factories



Use subclasses for domain-specific segment data.

The demo defines TypeSegment and IssueStatusSegment subclasses. The bar itself only requires value/text, but a custom factory can inspect extra fields and return any Node.

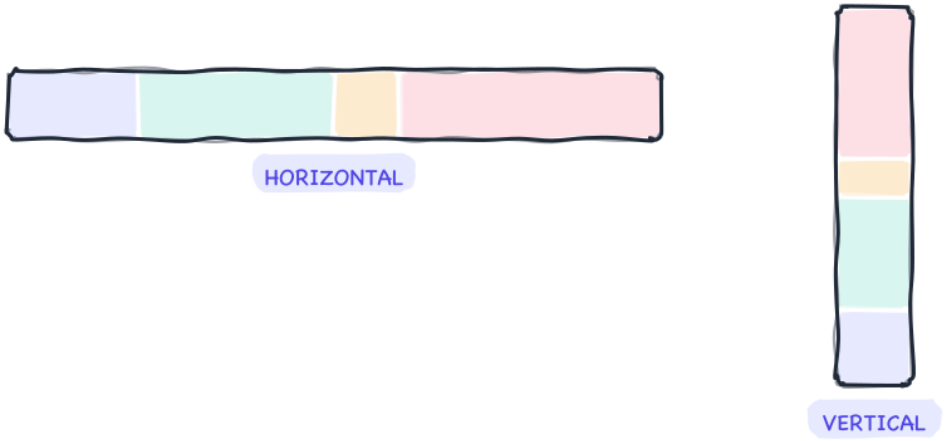
```
SegmentedBar<TypeSegment> bar = new SegmentedBar<>();
bar.setSegmentViewFactory(segment -> new TypeSegmentView(segment));
bar.getSegments().add(new TypeSegment(14, MediaType.PHOTOS));
```

Typed segment factory pattern.

6. Layout algorithm

The skin divides available width or height by segment.value / total. When a segment would be smaller than minSegmentSize, it receives the minimum and that space is taken from the remaining content size.

minSegmentSize keeps tiny values visible.



Horizontal and vertical orientations.

7. Default SegmentView

The default SegmentView is a StackPane with style class segment-view and one Label. The label text binds to segment.text and is hidden during layout when there is not enough space to show it fully.

8. Styling

Style class / pseudo class	Description
.segmented-bar	Root control style class.
:horizontal / :vertical	Orientation pseudo classes.
.segment	Added by the skin to every segment node.
.segment-view	Default SegmentView style class.
.only-segment, .first-segment, .middle-segment, .last-segment	Position classes added by the skin.

Style class / pseudo class	Description
.segment-view > .label	Default label inside SegmentView.

CSS property	Type	Default
-fx-orientation	Orientation	VERTICAL
-fx-min-segment-size	size	5

```
.segmented-bar {
    -fx-orientation: horizontal;
    -fx-min-segment-size: 10px;
}
.segmented-bar > .segment {
    -fx-background-color: steelblue;
}
```

Example CSS for SegmentedBar.

9. Localization and accessibility

Key	English default
accessible.role-description	segmented bar

The constructor sets AccessibleRole.NODE with the localized role description *segmented bar*.

10. Recipes

10.1 Custom colors by type

```
bar.setSegmentViewFactory(segment -> {
    StackPane view = new StackPane(new Label(segment.getText()));
    view.setStyle("-fx-background-color: " + colorFor(segment.getType()));
    return view;
});
```

10.2 Bind values

```
segment.valueProperty().bind(task.progressProperty().multiply(100));
```

10.3 Switch orientation

```
bar.setOrientation(Orientation.VERTICAL);
```

10.4 Checklist

1. Never create segments with negative values.
2. Use subclasses when the view needs more than value/text.
3. Set minSegmentSize for skewed data.
4. Use totalProperty for labels or summaries.

11. See also

- Demo app: `SegmentedBarApp`. Run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.SegmentedBarApp`.
- Related controls: ProgressIndicator-style controls for non-segmented progress.
- API docs: https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/