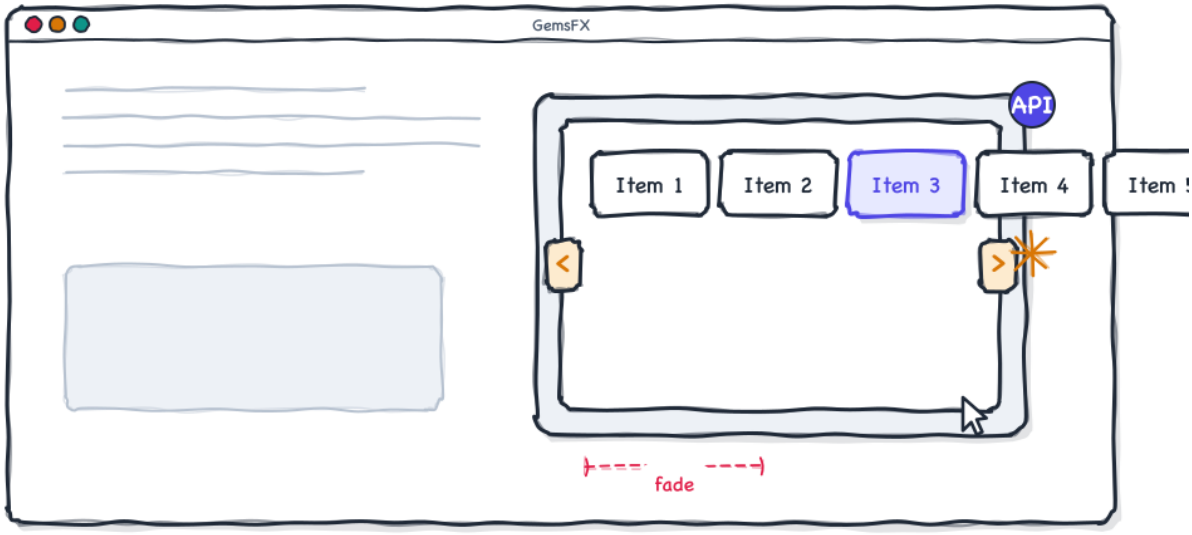
StripView

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



Generated cartoon overview of StripView.

StripView lays out a fixed sequence of items horizontally, fades the edges through MaskedView and shows arrow buttons when content overflows.

Table of Contents

1. Introduction

StripView is a compact horizontal selector. The skin creates one `StripCell` per item, places them in an HBox inside a MaskedView, and scrolls the container with mouse, keyboard or explicit scrollTo calls.

1.1 Key features

- Default preferred size is 400 x 50.
- Default cell factory creates StripCell labels.
- Selected item can auto-scroll into view and optionally stay centered.
- Scroll buttons fade in when content is hidden to the left or right.
- Keyboard selection supports LEFT, RIGHT, ENTER and TAB.

1.2 Maven dependency

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

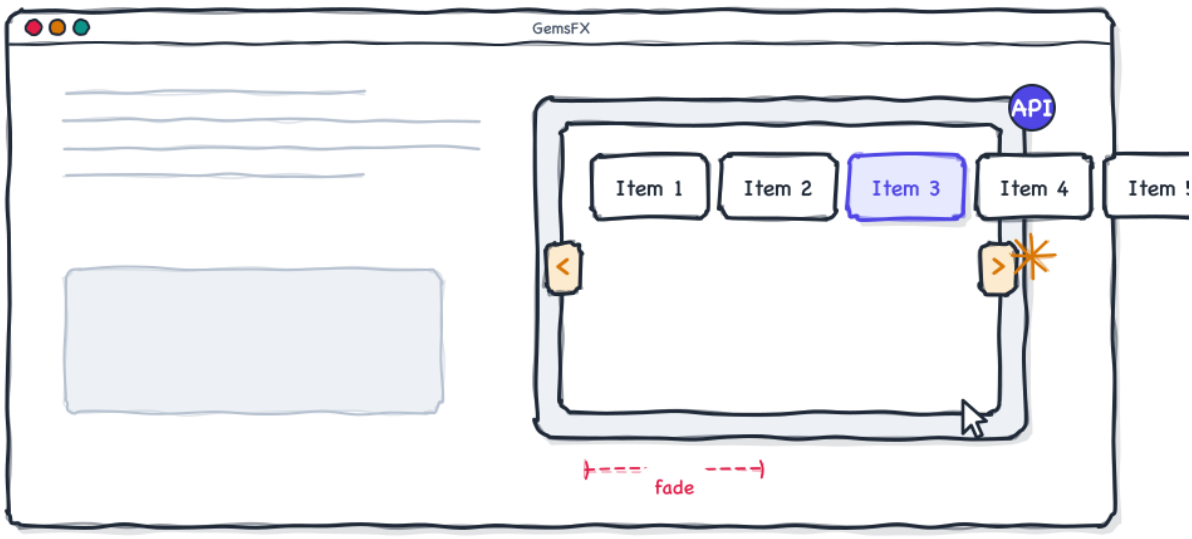
Use package `com.dlsc.gemsfx`.

2. Getting started

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

```
StripView<String> strip = new StripView<>();
strip.getItems().setAll("Item 1", "Item 2", "Item 3", "Item 4");
strip.setSelectedItem("Item 2");
strip.selectedItemProperty().addListener((obs, oldItem, newItem) -> {
    System.out.println("selected = " + newItem);
});
```

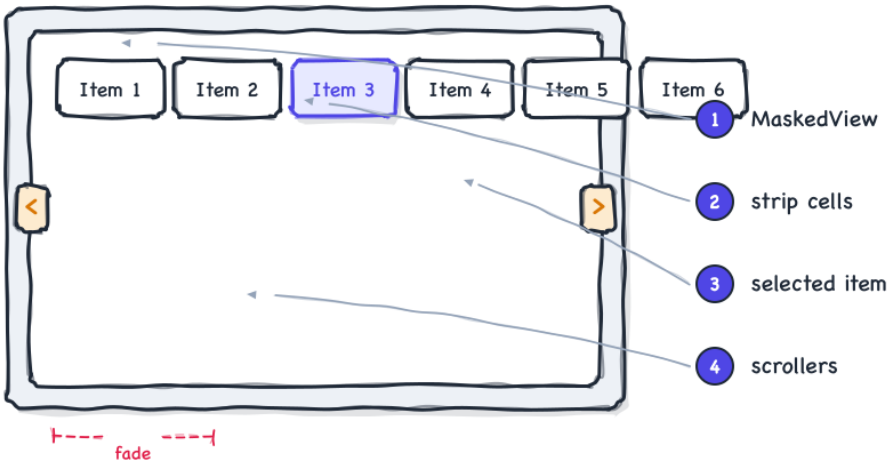
Minimal setup for `StripView`.



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
StripView	strip-view	Root control and stylesheet owner.
MaskedView	masked-view	Fades content at left/right edges.
HBox	container	Holds cells horizontally.
StripCell	strip-cell	Default selectable Label cell.
Scroller buttons	scroller left / right	Arrow regions shown when overflow exists.

4. Control API

4.1 Items and selection

Property	Type	Default	Description
items	ListProperty<T>	empty list	Model items shown in order.
selectedItem	ObjectProperty<T>	null	Currently selected item.
cellFactory	ObjectProperty<Callback<StripView<T>, StripCell<T>>>	new StripCell	Creates one cell per item.
autoScrolling	BooleanProperty	true	Automatically calls scrollTo(selectedItem) when selection changes.

4.2 Scrolling and layout

Property	Type	Default	Description
alwaysCenter	BooleanProperty	true	Selected item is centered if possible; styleable.
fadingSize	DoubleProperty	120	Fade width on both sides; styleable.
animateScrolling	BooleanProperty	true	Animates scroll and button fades; styleable.
animationDuration	ObjectProperty<Duration>	200 ms	Duration for scroll-to-item animation; styleable.
loopSelection	BooleanProperty	true	Keyboard selection wraps around; styleable.
scrollTo(T item)	method	property marker	Requests skin scrolling via the scroll.to property marker.

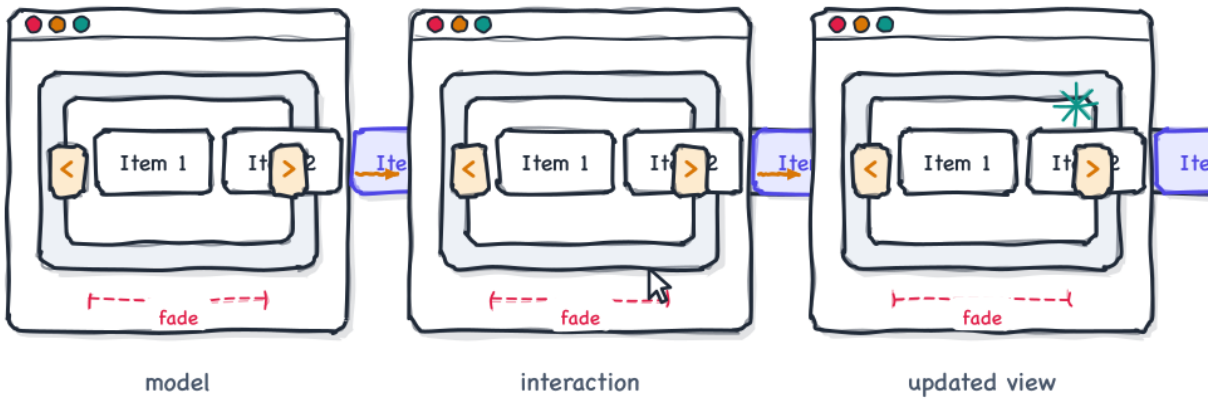
4.3 StripCell

Property	Type	Default	Description
stripView	ObjectProperty<StripView<T>>	set by skin	Back-reference to owning strip.
item	ObjectProperty<T>	set by skin	Cell item.
selected	BooleanProperty	false	Updates selected pseudo class and accessible selected attribute.

5. Behaviour

5.1 Selection and scrolling

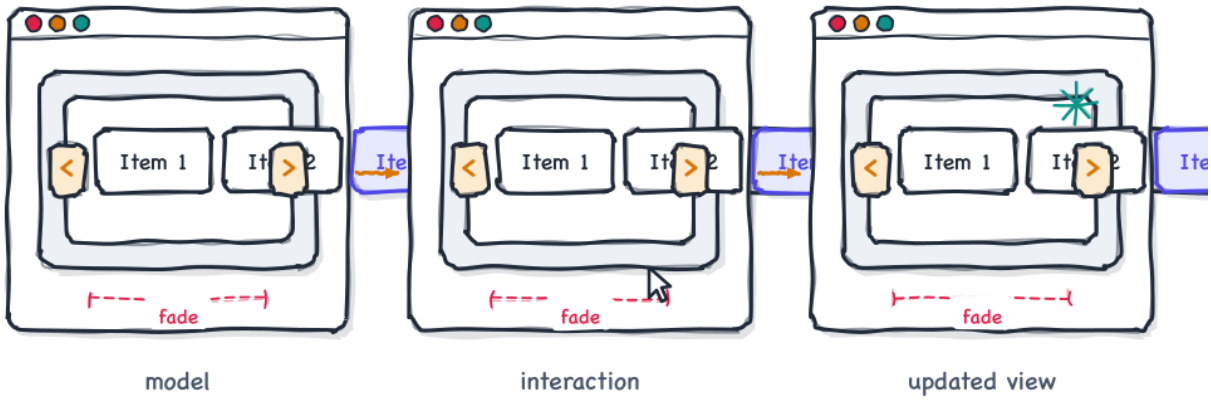
Clicking a cell selects it, requests focus and scrolls it into view. If alwaysCenter is true, scrollTo centers the item where possible.



The main runtime behaviour.

5.2 Keyboard and mouse navigation

RIGHT, ENTER and TAB select the next item; LEFT selects the previous item. With loopSelection true, navigation wraps at both ends. Mouse wheel adjusts the horizontal translate value.



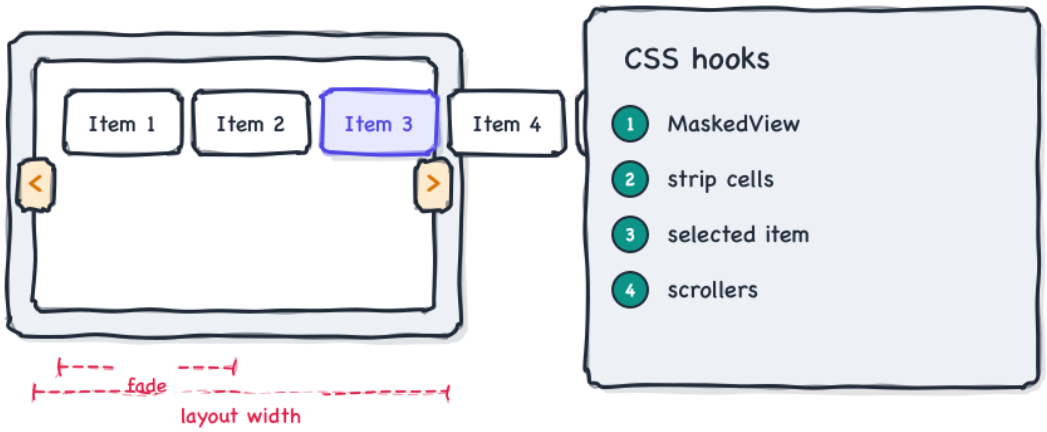
Data and interaction flow.

5.3 Overflow buttons and masking

The skin clamps translateX so content cannot scroll past its bounds. The left/right scroller opacity follows whether hidden content exists on that side.

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
strip-view	Root control style class.
masked-view	Nested MaskedView.
container	HBox content container.
strip-cell	Default cell style class.
scroller left right	Scroll arrow regions.

6.2 Pseudo classes

Pseudo class	Meaning
selected	Applied to StripCell when its item is selected.

6.3 Styleable CSS properties

Property	Type	Default	Description
-fx-always-center	Boolean	true	Center selected item when scrolling.
-fx-animate-scrolling	Boolean	true	Animate scroll transitions and button fades.
-fx-animation-duration	Duration	200 ms	Scroll-to-item duration.
-fx-fading-size	Number	120	Edge fade size.
-fx-loop-selection	Boolean	true	Keyboard navigation wraps around.

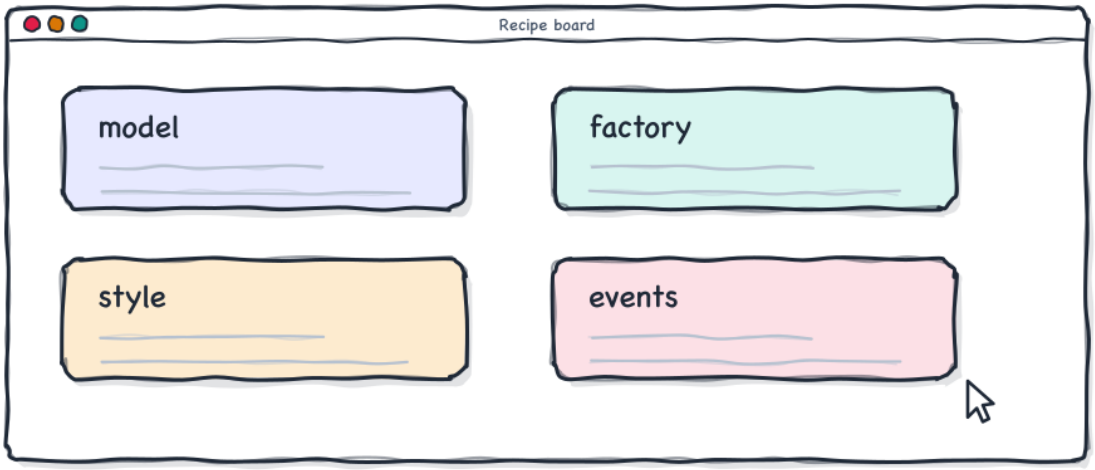
```
.strip-view {
    -fx-fading-size: 160;
    -fx-loop-selection: false;
}
.strip-view .strip-cell:selected {
    -fx-background-color: -fx-accent;
}
```

Example CSS.

7. Accessibility

StripView sets AccessibleRole.LIST_VIEW. StripCell notifies AccessibleAttribute.SELECTED when its selected property changes.

8. Recipes



Common configuration recipes.

8.1 Use custom cells

```
strip.setCellFactory(view -> new StripView.StripCell<>() {
    @Override protected void updateItem(String item) {
        super.setText(item == null ? "" : item.toUpperCase());
    }
});
```

8.2 Scroll without selecting

```
strip.scrollTo("Item 12");
```

8.3 Disable wrapping

```
strip.setLoopSelection(false);
```

8.4 Tune the fade

```
strip.setFadingSize(200);
```

8.5 Turn off animation

```
strip.setAnimateScrolling(false);
```

8.6 Checklist

1. Use selectedItemProperty as the selection model.
2. Call scrollTo only for items that exist in getItems().

3. Set cellFactory before populating when custom visuals are needed.
4. Large fadingSize values reduce the fully visible center area.

9. See also

Demo app: `StripViewApp`. Run it with:

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

- Related GemsFX controls: MaskedView, AutoscrollListView, Carousel-like controls.
- API documentation: https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/