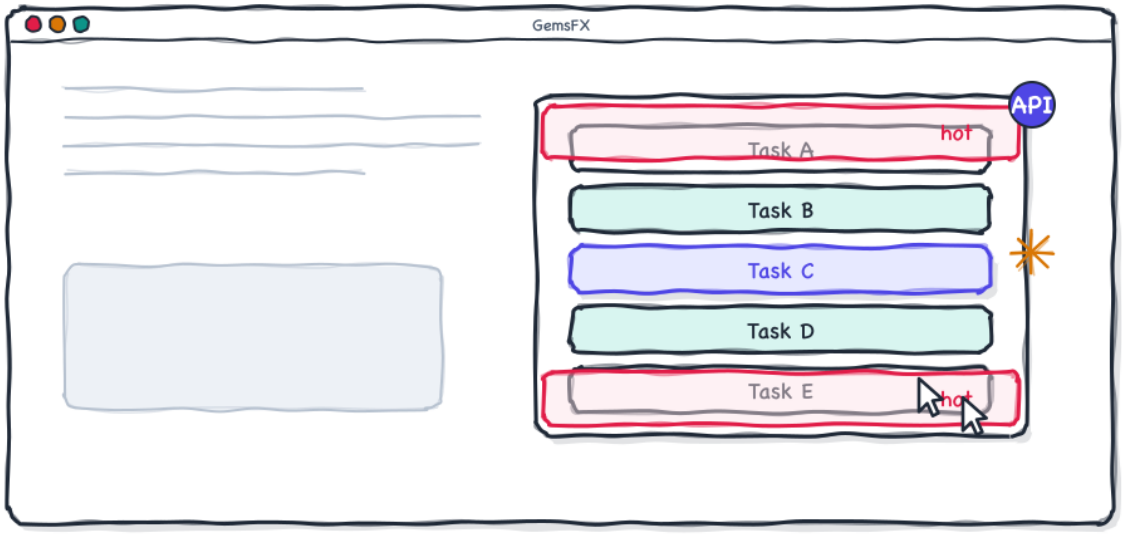
AutoscrollListView

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



Generated cartoon overview of AutoscrollListView.

AutoscrollListView extends JavaFX ListView and starts a daemon scroll loop when a drag operation reaches the top or bottom hot zone of the visible list area.

Table of Contents

1. Introduction

AutoscrollListView is a vertical ListView specialization for drag-and-drop UIs. It accepts transfer modes during drag-over, computes a vertical scroll delta near the clipped container edges, and scrolls the VirtualFlow until the drag leaves, drops or finishes.

1.1 Key features

- Constructors mirror ListView: empty or with an ObservableList of items.
- Uses a 20-pixel proximity hot zone at the top and bottom.
- Accepts TransferMode.ANY during drag-over.
- Starts a daemon ScrollThread with an initial 300 ms delay.
- Stops scrolling on DRAG_EXITED, DRAG_DROPPED and DRAG_DONE.

1.2 Maven dependency

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

Use package `com.dlsc.gemsfx`.

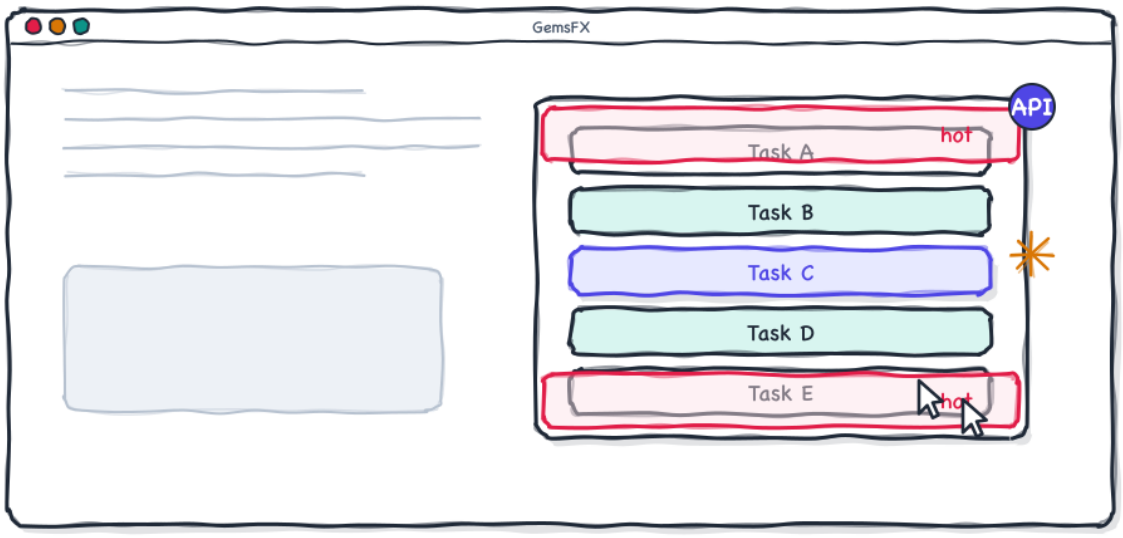
2. Getting started

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

```
AutoscrollListView<String> list = new AutoscrollListView<>();
list.getItems().setAll("One", "Two", "Three", "Four", "Five");

list.setCellFactory(view -> {
    ListCell<String> cell = new ListCell<>() {
        @Override protected void updateItem(String item, boolean empty) {
            super.updateItem(item, empty);
            setText(empty ? null : item);
        }
    };
    // install your drag source / drop target logic here
    return cell;
});
```

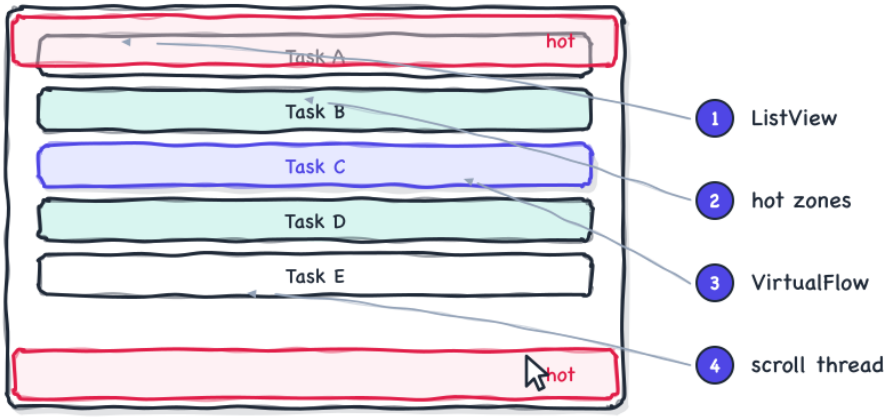
Minimal setup for `AutoscrollListView`.



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
AutoscrollListView	ListView subclass	Owns the items and standard ListView selection model.
DRAG_OVER filter	Event filter	Accepts transfer modes and computes scroll direction.
VirtualFlow	skin internals	Scrolled by pixel deltas from the background thread.
clipped-container	Region	Preferred hot-region for detecting top and bottom proximity.

Part	Type / style	Description
ScrollThread	daemon Thread	Posts Platform.runLater scrollPixels calls every 15 ms.

4. Control API

4.1 Constructors

Property	Type	Default	Description
AutoscrollListView()	constructor	empty observable list	Creates a list with FXCollections.observableArrayList().
AutoscrollListView(ObservableList<T>)	constructor	items from argument	Creates a list with the provided items.

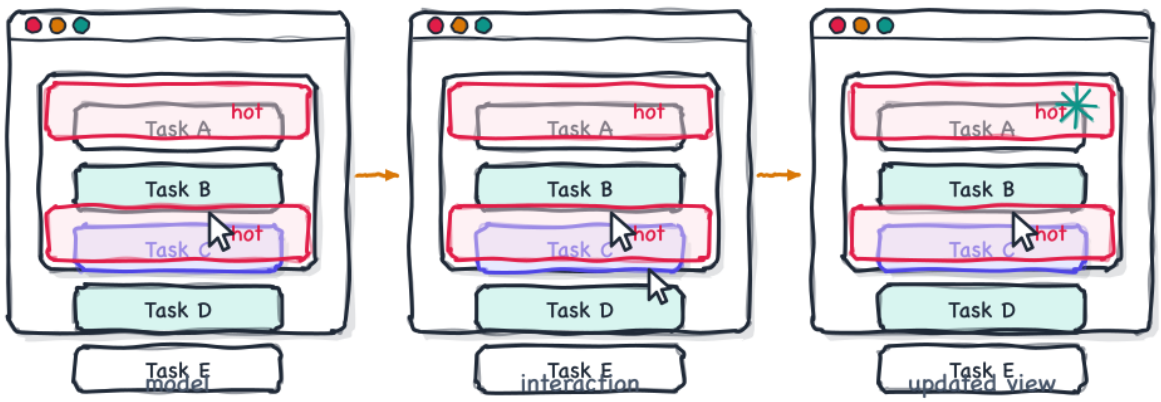
4.2 Implementation constants

Property	Type	Default	Description
proximity	double field	20	Distance in pixels from top or bottom that triggers autoscroll.
scrollThread	ScrollThread field	null	Created lazily while a drag is inside a hot zone and cleared on drag end.

5. Behaviour

5.1 Hot-zone scrolling

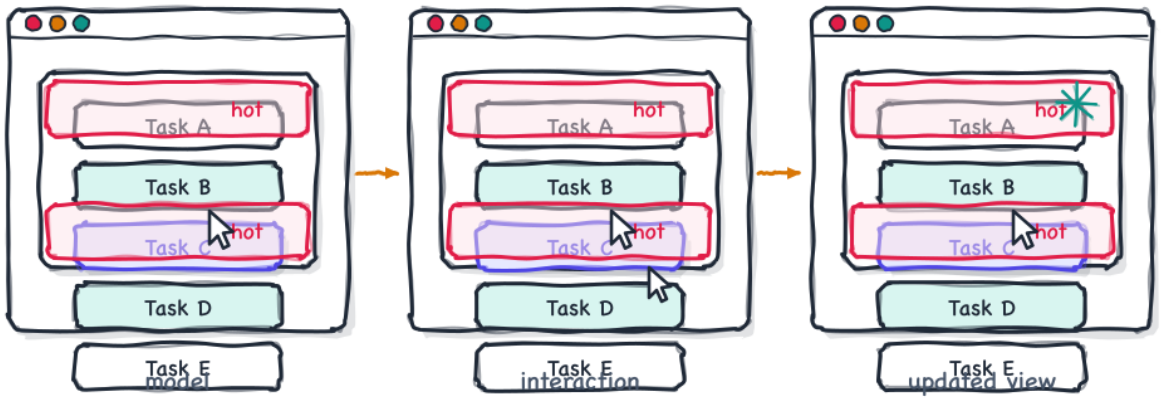
During DRAG_OVER the control finds the clipped container of the VirtualFlow. If the cursor is within 20 pixels of the top, it scrolls up; if it is within 20 pixels of the bottom, it scrolls down.



The main runtime behaviour.

5.2 Thread lifecycle

The ScrollThread sleeps 300 ms before its first scroll, then posts scrollPixels(yOffset) about every 15 ms. It is stopped by drag exit, drop and done events.



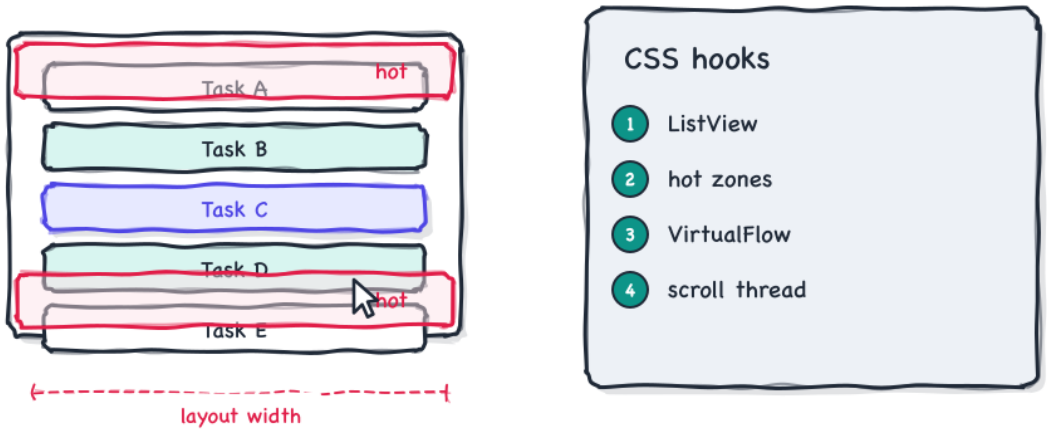
Data and interaction flow.

5.3 Vertical-only implementation

The source explicitly documents vertical orientation only. There is no horizontal autoscroll path.

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
none	No dedicated GemsFX stylesheet or style classes were found.

6.2 Pseudo classes

Pseudo class	Meaning
none	No pseudo classes are set by this control.

6.3 Styleable CSS properties

Property	Type	Default	Description
none			No styleable CSS properties are declared by this control.

```
/* AutoscrollListView inherits normal ListView styling. */
.autoscroll-list-view .list-cell {
    -fx-padding: 0.5em;
}
```

Example CSS.

7. Implementation notes

7.1 VirtualFlow lookup

The implementation uses `lookup("VirtualFlow")` and then scans the flow children for the `clipped-container` style class. This is why the control is tied to the standard JavaFX ListView skin structure.

7.2 Empty-list fallback

If the clipped container has no width, the code falls back to the list view itself. If that fallback still has no width, autoscrolling is stopped for the current drag event.

7.3 Scroll delta

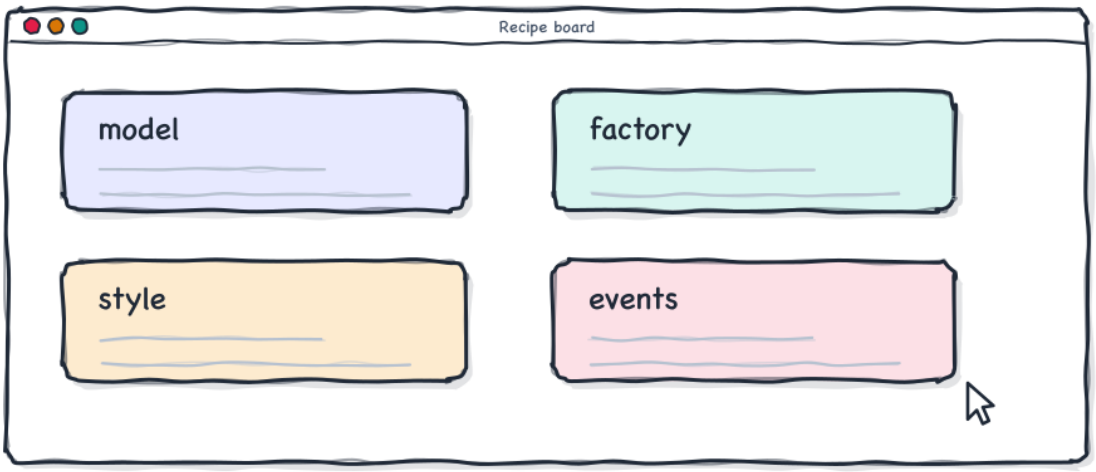
```
double delta = evt.getSceneY() - hotRegion.localToScene(0, 0).getY();
if (delta < proximity) {
    yOffset = -(proximity - delta);
}

delta = hotRegion.localToScene(0, 0).getY() + hotRegion.getHeight() - evt.getSceneY();
if (delta < proximity) {
    yOffset = proximity - delta;
}
```

The signed delta is larger the deeper the cursor is inside a hot zone.

> **Tip.** Because the scroll loop is an implementation detail, applications should not try to start or stop it directly. Use normal drag events; the control stops the daemon thread when the drag exits, drops or finishes.

8. Recipes



Common configuration recipes.

8.1 Use as a kanban column list

```
AutoscrollListView<Task> list = new AutoscrollListView<>();
list.setCellFactory(view -> new TaskCell());
```

8.2 Provide items in the constructor

```
ObservableList<String> items = FXCollections.observableArrayList("A", "B", "C");
AutoscrollListView<String> list = new AutoscrollListView<>(items);
```

8.3 Stop work on drag completion

```
list.addEventHandler(DragEvent.DRAG_DONE, evt -> {
    // AutoscrollListView stops its scroll thread automatically.
});
```

8.4 Combine with MultiColumnListView

```
multiColumnListView.setListViewFactory(view -> new AutoscrollListView<>());
```

8.5 Checklist

1. Use it for vertical ListView drag-and-drop scenarios.
2. Install your own drag source/drop target logic; this class only handles edge scrolling.
3. Do not rely on a public proximity property; it is a fixed field.
4. No dedicated CSS, bundle or accessibility setup exists.

9. See also

No dedicated demo app was found in `gemsfx-demo` for this control.

- Related GemsFX controls: MultiColumnListView, ListView.
- API documentation: https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/