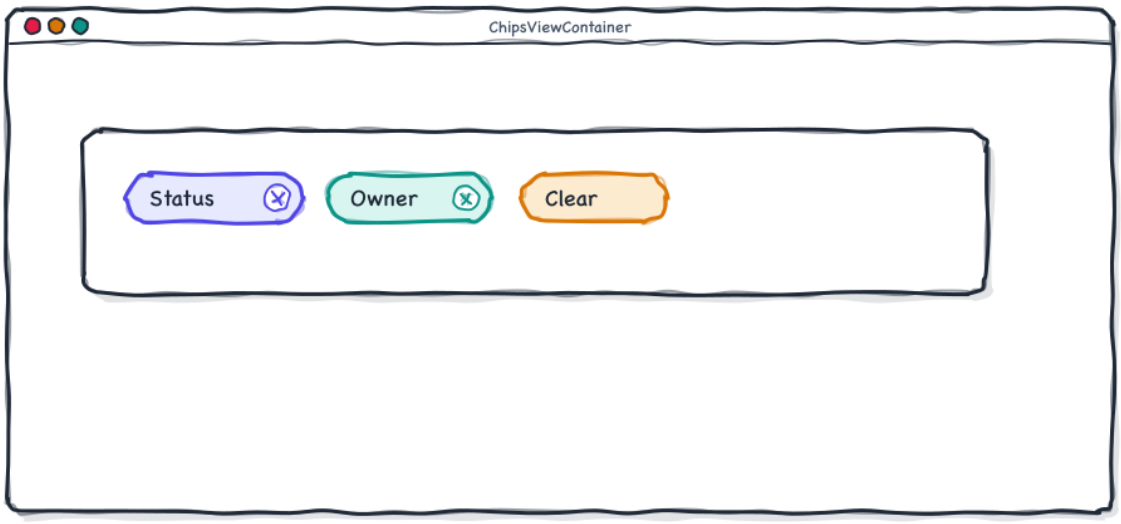
ChipsViewContainer

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



Generated cartoon overview of ChipsViewContainer.

A FlowPane that displays ChipView instances and, when configured, a clear Hyperlink. It manages its own visibility from the chip list.

Table of Contents

1. Introduction

ChipsViewContainer A FlowPane that displays ChipView instances and, when configured, a clear Hyperlink. It manages its own visibility from the chip list.

1.1 Key features

- When chips becomes empty the container is invisible and unmanaged.
- updateChips clears children and adds every ChipView from chips.
- API and styling details below are verified against the control, skin, CSS and resource bundles.

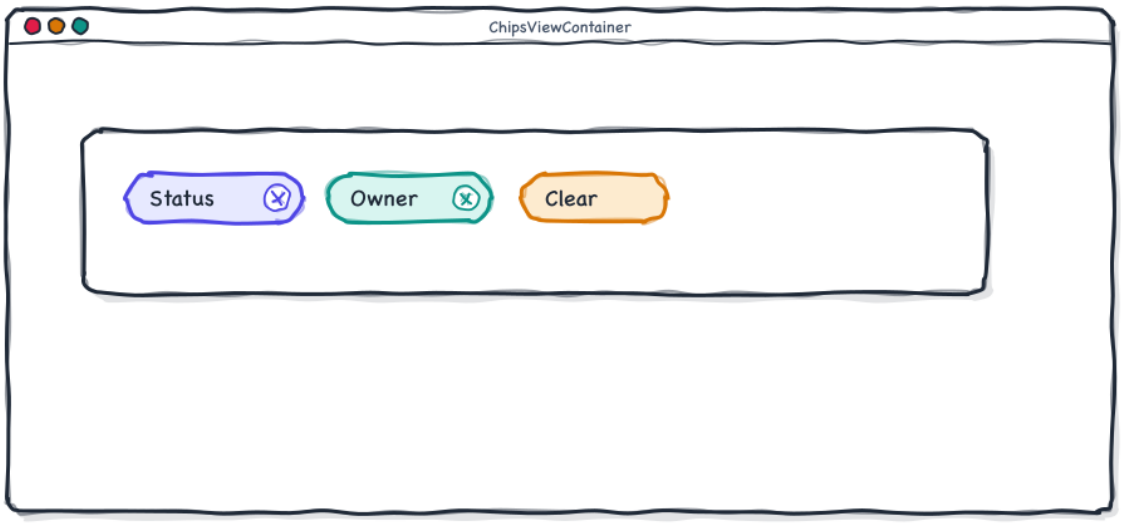
1.2 Maven dependency

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

2. Getting started

```
ChipsViewContainer container = new ChipsViewContainer();
container.getChips().setAll(statusChip, ownerChip);
container.setOnClear(() -> activeFilters.clear());
container.setClearText("Reset filters");
```

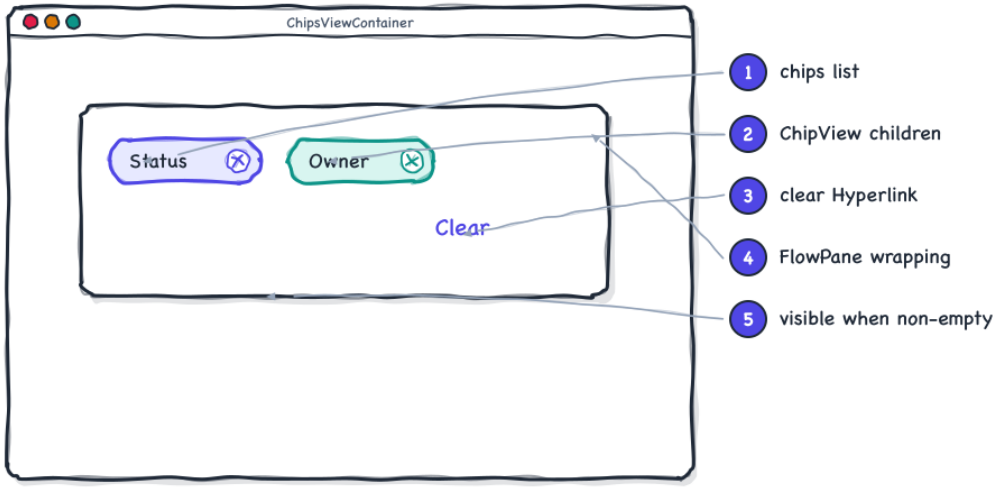
A compact setup for ChipsViewContainer.



ChipsViewContainer in a typical application context.

3. Anatomy

The diagram identifies the nodes and state holders created by the implementation.



The parts of ChipsViewContainer.

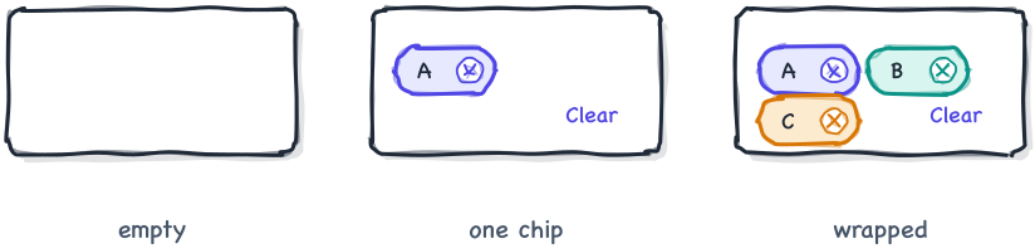
Topic	Verified source detail
Stylesheet	`com/dlsc/gemsfx/chips-view-container.css`
Root style class	`.chips-view-container`
Graphics	Generated SVGs; no screenshots.

4. Control API

4.1 Properties and callbacks

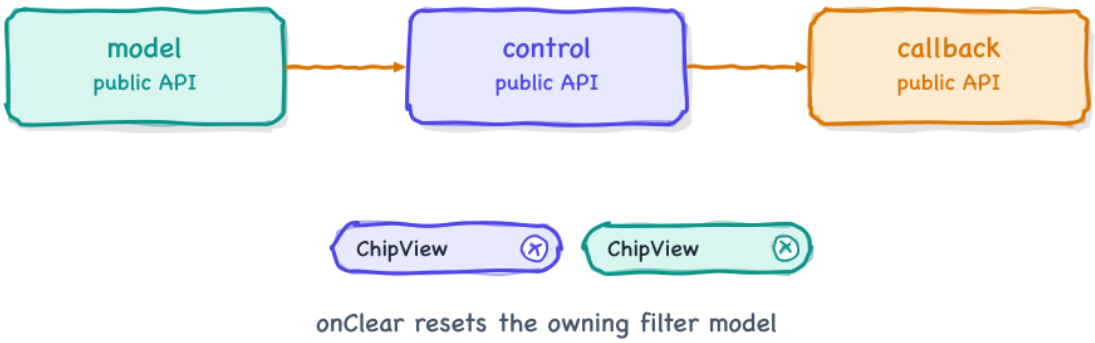
Property	Type	Default	Description
chips	ListProperty<ChipView<?>>	empty list	Chip controls shown in the FlowPane.
onClear	ObjectProperty<Runnable>	null	Invoked by the clear Hyperlink.
clearText	StringProperty	localized "Clear"	Text of the clear Hyperlink.
visible	BooleanProperty	bound to !chips.empty	Container visibility.
managed	BooleanProperty	bound to visible	Container managed state.

5. Behaviour



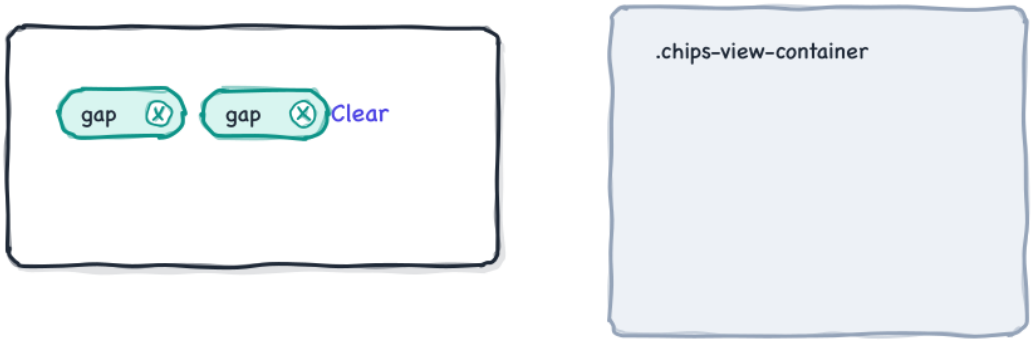
Important runtime states of ChipsViewContainer.

- When chips becomes empty the container is invisible and unmanaged.
- updateChips clears children and adds every ChipView from chips.
- A Hyperlink is appended only when there are chip children.
- The Hyperlink text is bound to clearText and its action calls onClear.run().



How application data flows through ChipsViewContainer.

6. Styling



Style hooks for ChipsViewContainer.

6.1 Style classes and pseudo classes

Selector / pseudo class	Purpose
`.chips-view-container`	Verified in source, skin or CSS.

6.2 Styleable CSS properties

This control declares no additional styleable CSS properties beyond inherited JavaFX properties.

```
.chips-view-container {
    /* start with the documented root selector */
}
```

7. Localization

Key	English text
`action.clear`	Clear

8. Accessibility

No explicit AccessibleRole and no AccessibilityUtil usage in the source.

9. Recipes

9.1 Programmatic configuration

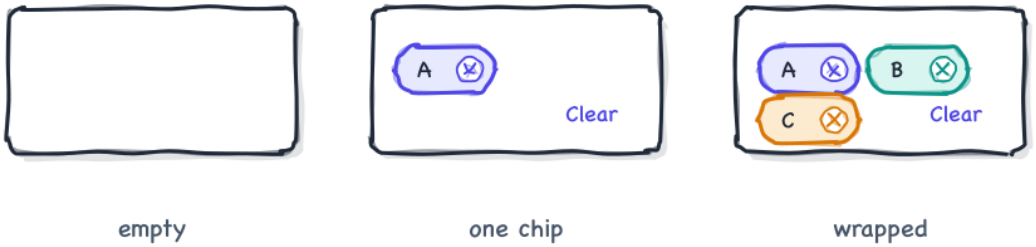
```
ChipsViewContainer container = new ChipsViewContainer();
container.getChips().setAll(statusChip, ownerChip);
container.setOnClear(() -> activeFilters.clear());
container.setClearText("Reset filters");
```

9.2 Checklist

1. Use the public properties listed in the API chapter.
2. Style through documented selectors and CSS properties only.
3. Keep application model updates in callbacks such as onClose, onClear or onItemSelected.
4. Do not depend on private skin nodes except for documented style selectors.

10. Layout and visibility details

ChipsViewContainer is deliberately small: layout is inherited from FlowPane, and all dynamic behaviour comes from the chips list listener and the visible / managed bindings.



Empty, one-chip and wrapped states.

State	Result
chips empty	The container is invisible and unmanaged, so parent layouts reclaim the space.
chips non-empty and onClear null	Only ChipView children are shown; no active clear link is managed.
chips non-empty and onClear set	A Hyperlink with clearText is appended after the chips.
chips list replaced	The listener on the ListProperty calls updateChips and rebuilds the children.

```
container.visibleProperty().bind(container.chipsProperty().emptyProperty().not());
container.managedProperty().bind(container.visibleProperty());
```

The visibility rule used by the implementation.

11. Integration notes

11.1 Use public API boundaries

The implementation exposes enough properties for normal integration. Keep application state in observable lists, converters and callbacks instead of querying skin children.

11.2 Validation checklist

Concern	Recommendation
Model updates	Perform them in the documented callback or observable list.
Styling	Start at the root style class and keep selectors scoped.
Localization	Override the documented resource bundle keys only when they exist.
Accessibility	Preserve the role set by the control and add application-specific accessible text when needed.

```
// Integration pattern
// 1. Configure model properties.
// 2. Configure callbacks.
// 3. Style through documented selectors.
// 4. Leave skin internals private.
```

12. See also

- Demo application: `com.dlsc.gemsfx.demo.SimpleFilterViewApp` (run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.SimpleFilterViewApp`)
- API documentation: https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/