

Flex in a Week, Flex 4.5

Video 1.13: Adding scrollbars

In this video, you will learn about viewports and how they relate to the implementation of the Spark Scroller component to add scrollbars to your application.

You can think of a viewport as a window through which you can view content that is otherwise concealed.

Any class that implements the `IViewport` interface can act as a viewport.

The `Group` container, and its subclasses `VGroup` and `HGroup`, can function as viewports.

The left side of this schematic illustrates some employee content and a possible viewport, in gray, for that content.

The left side of this schematic shows the implemented viewport, with a vertical scrollbar.

You may have noticed that the Spark containers are not automatically scrollable.

You must use the Spark Scroller component to display horizontal and/or vertical scrollbars.

The component will determine which scrollbars are necessary based on the size of the viewport and the content being viewed.

You add the Scroller component to your containers by wrapping them in the tag.

Be careful that you put the size constraints on the Scroller itself, not the viewport container.

It will work, if you put the constraints on the viewport container, but it is a recommended best practice to leave them on the Scroller.

If you are working with skinnable containers, you must define their scrollbars in their skins.

You will learn how to do that in Day 5.

Here is the application that I showed you in the last video.

I have some header text and then a number of employees listed in a tile layout.

I want make the list display smaller so I can add a scrollbar so, back in the code, I am adding the height and width properties to the `BorderContainer` with values of 260 and 500 pixels, respectively.

When I save the file and run the application, you can see that the border changed but it did not constrain the display of the employees.

I am adding a Scroller tag block around the BorderContainer instance.

When I save the file, you will see a compiler error.

Remember that the BorderContainer class is a skinnable container, which means that you should add scrollbars in its skin.

You will learn about skinning in later videos.

For now, I will add a Group container inside the BorderContainer class and then move the Scroller tag block around it.

I am also moving the width and height properties from the BorderContainer to the Group container.

Note that it's considered a better practice to put the width and height properties on the Scroller rather than the Group container.

However, in this case, doing so does not give me the desired results.

Now let me do a little code cleanup.

This series follows a coding methodology that involves indenting nested tag blocks so I am highlighting the entire Group container and pressing the Tab key on my keyboard to indent it.

I'm also fixing the rest of the code.

When I save the file and run the application, you can see that the Group container is scrollable but the border is around everything, including the scrollbar.

Back in the BorderContainer class, I am adding a width property and setting it to 500.

Now when I save the file and run the application, you can see that the border hugs the Group container and is inside the scrollbar.

You can also define a Scroller component's viewport using the viewport property of the Scroller tag.

In this code, I have defined an id property for the Group block and bound it to the viewport property of the Scroller tag.

Back in my code, I am removing the Scroller block around the Group container and then highlighting the code and typing Shift+Tab to un-indent it.

Next, I am giving the Group container an id property with a value of employeeGroup.

Now I am adding a Scroller instance between the closing Group and BorderContainer tags and adding the viewport property, binding it to the value of the Group container's id property, employeeGroup.

When I save the file and run the application, you can see it looks the same as before.

For your next step, work through the exercise titled “Experimenting with container layouts”.