Now we've been writing a lot of code recently.

Let's quickly review what we're actually trying to build. The final app that we're looking to build is going to look like this.

The part I want to draw your attention to is the fact that we have a loading screen where we're fetching the location and the weather data, and we have the location screen where we're displaying the weather information to the user. In this lesson we're going to improve that transition between the loading screen and the location screen by adding a little spinner. In terms of our code,here's what we've implemented so far. In the last lesson we created our networking.dart helper class here where we pass over a URL when we initialize this class and then we use that URL to make a http get request.

Then we check to see if the response was favorable namely if it was 200 and the request was received successfully. And we take the body of the response and use jsonDecode to decode that data and pass it back out into wherever it's needed.

So notice how this network helper is really flexible.

It doesn't just have to work on weather data, it can do anything that is networking related. And we've split off that functionality into its own custom location so that we can use it for other things in the future, if we need more networking functionality.

Now in our getLocationData method, we're getting the current location of the user and then we're using our networkHelper to fetch the weather data for that location. And then we end up with a weather data object that we can use somewhere else.

So the somewhere else that we need it is our location screen.

We need to put that data into these text widget so that we can display it on the screen for the user.

Now in order to pass that data over to the location screen, we're going to have to use something that we learnt previously, which is using the navigator to push a route onto the screen.

So the context is going to be kept as the context.

The route is going to be created using the MaterialPageRoute, and the builder is going to expect a callback.

So something that has an input of context and it's going to return the screen that we want to show which is the location screen.

And of course it doesn't know about the location screen unless we import the location\_screen.dart file into here. And now we have the location screen route setup. So now if we run our app again, so hot restart, then it should start out in this black screen. And once it's gotten the location and it's gotten the weather data, it should push over the next screen which is the location screen.

Now that takes a little while right?

And the user's kind of staring at a blank screen for a while.

So that might make them think that the app has crashed or is not working.

So let's add a loading indicator onto the loading screen so that they actually know that something is happening in the background and we're waiting to get some data or get some location. The package that we're going to use is called Flutter SpinKit.

And what it does is it gives us access to a whole bunch of different animated loading indicators that we can use in our app.

And the one that I particularly fancy is this double bound here, which shows that something's happening,wait for it and will make the screen transition.

So let's go ahead and add this as a dependency to our app.

So let's go into our pubspec.yaml again and add that below our http. And then I'm gonna remove the quotes as per usual to keep my formatting consistent and I'm gonna run packages.get to fetch that package.

Now once that's done, I can imported into my loading screen, so right at the top here. And we're going to use it inside our build method.

So the way that we use it is to tap into the particular widget that we want, be it's SpinKit rotating circle or SpinKit double bounds or SpinKit wave, whatever it is that we need.

And then we can set a color, we can set a size.

So that's what I'm going to do inside my scaffold. So my scaffold is simply going to have a body which is going to be a center widget.

So I can center my SpinKit animator and the center widget has a child property which I'll set to the SpinKitDoubleBounce. And I can use this widget just like any other Flutter widget.

I simply add some properties,so I'm gonna give it color property, gonna set that to maybe just white. And I'll also give it a size property which I'll set to a 100pixels to make it relatively large.

So now let's rerun our app from the start,so hot restart and it should start out from the blank screen.

It's gonna use the loading indicator for a little while before it fetches the location and the weather and it pushes onto the screen.

So our app now looks a lot more user friendly. But we still have a problem because we have our weather data in our loading screen, but where we actually need it is our location screen.

So how can we parse it over? Now parsing data from the loading screen to the location screen and displaying that data is what we'll cover in the next lesson.

So I'll see you there.

