Flex in a Week, Flex 4.5

Video 3.08: Using the Flash Builder data wizard

At the end of Day 2, you used the Flash Builder Data/Service view to generate a data service and a value object class.

However, you only used the value object class.

In this video, I will, again, use Flash Builder to generate data services.

Then I will show you how to use the visual features of the tool to create a master/detail form based on that data service.

This is the master/detail form that you will learn to create in this video.

You will use the generated data service calls to populate this DropDownList control with employee last names.

As you select an employee from the list, that employee's information will be displayed in this detailed view.

In my starter project, I am creating a data service by selecting the Data/Services view.

I am clicking the Connect to Data/Services link which opens the dialog window to select the service type.

I am selecting the HTTP service type and clicking Next.

In the dialog window, I am entering getEmployees for the operation.

I am entering the url to the page on the server that will respond with the data.

I'm setting the service name to EmployeeService.

Notice that when I entered the service name, the service package and data type package were automatically entered into the fields.

The service name represents the identity of the service within Flash Builder.

The service package is where the Flash Builder generated service files will live.

The data type package is where the generated value object files will live.

We discussed this in the video at the end of Day 2.

I am clicking Finish.
In the Data/Services view I can see my service -EmployeeService - and the operation associated with it.

Now I need to configure the return type because it is currently set to return a generic object.

I also showed you how to do this at the end of Day 2, so this will be a quick review.

I am right-clicking on the getEmployee operation and select Configure Return Type.

I will auto-detect the type so I am clicking Next.

I am keeping the defaults on this page as well, so I am clicking Next again.

I am changing the root of the XML properties to the repeating node and clicking Finish.

You can see that the return type has changes to Employee[].

Remember that the square brackets let you know that the service will return an array of objects.

Now that that is taken care of, I'll build the interface.

This is my main application file.

I am switching to Design mode and, from the Components view, I am dragging a DropDownList control and placing it under the Employee Portal; Employee Information label and using the blue guidelines to align it to the left in the Design area.

From the Data/Services view, I am dragging the getEmployees() operation and dropping it on the DropDownList control.

When the Bind To Data dialog box appears, I am leaving the New service call radio button selected and selecting the lastName field to display in the control.

I am clicking OK to close the dialog.

In the Design area, I am right-clicking on the DropDownList control and selecting Generate Details Form.

Within the Generate Form dialog box, I am making sure that the Master-Detail option is selected and am deselecting the Make form editable check box.

I am clicking Next and then unchecking the lastName property to remove it from the form display.

When I click Finish, the form appears in the Design area.

I am moving it to the right of the DropDownList control using my arrow keys.

When I save the file and run the application, you see the form displayed without any employee data.

I am selecting an employee from the DropDownList control to populate the form.

For your next step, work through the exercise titled “Creating a master/detail interface with a wizard”.