Flex in a Week, Flex 4.5

Video 3.04_remote: Making a RemoteObject request with parameters

In the first and second Days of training, you learned how to request data from the server to use in your application.

In this video, you will learn how to send data to the server using the Flex RemoteObject component.

You will also handle the confirmation message from the server that returns the record id for the data that was inserted into the database.

In this Flex in a Week training, I am assuming that you are watching the videos in order, which means that you have already watched the introduction to the HTTPService component video on this third Day of training.

If you have not, you might want to consider watching the first four-and-a-half minutes of that video.

In that section, I orient you to the topics you have learned that prepare you for sending this data to the server.

If you watched the WebService component video, also on this third Day of training, then you will find many similarities with RemoteObject.

In this video I will use the RemoteObject component to send the form data as a parameter to the server.

When I submit the form, the confirmation message displays the id for the record that was inserted into the database.

In the Declarations block of the starter file, I already have one RemoteObject component instance.

I am creating a second RemoteObject instance to send the data back to the server.

I am giving it an id property with a value of vehicleService.

I am adding the destination property with a value of ColdFusion.

You learned on Day 2, that the destination value is configured in a configuration file on the remoting server.

For a ColdFusion server, the value is ColdFusion.

Remember that, although we are using a ColdFusion server for this training series, you can use any backend technology – like Java, JSP, .NET or PHP – that has an AMF-compatible server.

I am adding a source property with a value of f45iaw100.remoteData.vehicleRequestData.

This dot notation path points to the ColdFusion CFC that will accept the data.

I am adding an endpoint property with a value of this URL (http://adobetes.com/flex2gateway/).

This property is not necessary if your Flex application is on the same server as the remoting service.

Since this Flex application is local and pointing to a remote service, we need to define this endpoint.

Lastly, I am adding the result event and using Flash Builder to generate the result handler.

Now, I am opening the VehicleRequest.as file from the valueObjects package.

Above the VehicleRequest() class definition function, I am typing [Rem to invoke the content assist tool and pressing Enter to insert the RemoteClass metadata tag into my code.

I am placing my cursor inside the closing bracket of the RemoteClass tag and typing an opening parentheses. I am pressing CTRL+Space and the alias property is added to my code.

I am changing the value of the alias property to f45iaw100.remoteData.VehicleRequest.

This will map the VehicleRequest value object to this Java object on the ColdFusion server.

I am saving the file.

The two ways to pass arguments to the server-side method calls through RemoteObject are: parameter binding and explicit parameters.

Parameter binding allows developers to pre-define the source data for arguments that will be passed to server-side methods.

This technique is only available if you use the RemoteObject MXML tag.

You nest an arguments MXML tag within the method MXML tags of the RemoteObject tag block.

All arguments are placed in ordinal position, not by name.

To call a method and pass the defined values, use the RemoteObject class' send() method.

When using explicit parameters, you add the arguments to the method call in the same order that they are defined in the server-side method.

Note that the argument is defined as an object name/value pair.

This is the way that I will send the data to the server for the Employee Portal: Vehicle Request Form.

I have debugged the application so that you can see the VehicleRequestData value object in the event object that is passed from the custom component.

I'm stopping the debugging session and returning to the Flash Perspective.

In the event handler for the custom component event, I am deleting the Alert.show() method and adding the vehicleService.addVehicleRequest() method to send the data to the server.

This method is expecting the data object and I am passing it the event.vehicleRequestData.

I'm saving the file.

I am enabling the Network Monitor, maximizing the view and then running the application.

You can see that the first RemoteService call retrieves the data for display in the DropDownList control.

I am filling out everything in the form, except the Mobile Phone field and clicking the Submit button.

I receive a runtime error.

I am returning to Flash Builder. You can see that a second service request was made, but there was an error because the mobile phone number is required.

I am minimizing the Network Monitor view and returning to my code.

Now, I am going to add a fault handler to show an Alert message when the Mobile Phone field is left blank.

To the vehicleService RemoteObject object, I am adding a fault event and using the content assist tool (CTRL+Space) to generate the fault handler.

I am locating the generated fault handler in the Script block and deleting the generated code stub.

Within the function, I am adding the Alert.show() method to display the event.fault.faultString as the text in the Alert pop-up.

I am giving the Alert message a title of Fault Information for addVehicleRequest.

I am saving the file and running the application.

Now when I fill out everything except the Mobile Phone field in the form and submit it, I see an alert message.

I want to display an alert message with the record id when the data is returned from the server.

So, in the result handler, I am typing if and pressing CTRL+Space twice to show code templates in the content assist tool. I am selecting the template for the if statement.

I am setting the conditional statement to evalutate the event.result.id value.

Within the if statement, I am adding an Alert.show() method that states "The request was submitted. The record id is " and then concatenating the event.result.id.

I am saving the file, running the application and then filling out the form.

When I submit the form you can see that the record id for the inserted data is successfully returned.

For your next step, work through the exercise titled “Passing data to the server with the RemoteObject class”.