

Flex in a Week, Flex 4.5

Video 3.04_web: Making a WebService 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 WebService 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 it, I review how data has been retrieved from the server to populate this Employee Portal: Vehicle Request Form.

I also review how the controller is handling the data between the model and the view.

Lastly, I discussed packaging the value object, which contains all the data that the user fills out in this form, in an event object, which is now ready for us to pass back to the server.

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

When I submit this form the record id for the record that was inserted into the database is returned and I handle that in an alert message.

This is the starter file for the main application.

In the Declarations block, I already have one WebService component instance.

I am typing Web and pressing CTRL+Space to show code templates in the content assist tool.

I am pressing Enter to insert the WebService code template.

I will use the second WebService instance to send the data back to the server.

I am modifying the code template so that the WebService object is in single tag syntax and not block syntax.

I am changing the id property value to vehicleService and changing the wsdl property value to this URL (<http://www.adobetes.com/f45iaw100/remoteData/vehicleRequestData.cfc?wsdl>).

Remember that, although we are using a ColdFusion server to generate this WSDL document, you can use any backend technology – like Java, JSP, .NET or PHP – that supports SOAP-based web services.

Now, I am removing the value within the result event and using the Flash Builder's code assist tool to generate the result handler.

Lastly, I am deleting the useProxy, showBusyCursor, and fault properties from the WebService instance.

The two ways to pass arguments to the server operations through web services 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 operations.

This technique is only available if you use the WebService MXML tags.

You nest a request MXML tag within the operation MXML tags of the WebService tag block.

To call an operation and pass the defined values, use the WebService class' send() method.

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

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

In the custom event handler for this custom component, I am placing a breakpoint on the closing curly brace.

I am debugging the application by first filling out the form and submitting it.

This will trigger the breakpoint so that you can see the VehicleRequestData value object in the event object.

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

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

The data object that I am passing is 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 WebService 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.

Back in Flash Builder, you can see that a second service request is made, but there was an error.

In the Response tab I am switching to Raw view.

The first line tells me that I have experienced the 500 Internal Server Error.

As I scroll down, I see a message that says "Field 'VEHICLEREQUESTS.MOBILEPHONE' cannot be a zero-length string."

The mobile phone number is a required field.

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 WebService 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.

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 with the faultString.

Of course, in the real world you likely would not have printed the fault string in the alert message.

You can place any message that you want to in the alert box.

Lastly, 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 condition statement to evaluate 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 the server sends back the record id for the record that was submitted into the database.

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