Flex in a Week, Flex 4.5
Video 2.06: Introducing RPC

Flex applications can load data in many ways, but in the Flex in a Week training, we will focus on using the remote procedure call, or RPC, components.

First, I will briefly introduce the three components and then I will compare the data packet sizes that are returned by each of them.

Lastly, you will learn about the crossdomain.xml file that is required to access services in different domains.

This video will just be a brief introduction and comparison of the three RPC components.

You will learn how to work with them in depth in the rest of this day of training as well as in Day 3.

RPC is based on a service-oriented architecture.

The RPC components interact with server environments via service calls to send and receive data.

The Flex application calls the services and waits for a response.

This model is asynchronous, so the application implements event handling to listen for the returned data.

The data is returned from the service and converted into an ActionScript object, which can then be used by the Flex application.

The three RPC components are:

HTTPService to request and send data via the HTTP GET or POST methods.

WebService to work with SOAP-compliant web services,

and RemoteObject to access native server-side methods directly using the binary Action Message Format, or AMF, services.

In the next few exercises, you will experiment with all three RPC components.

This is the exercise solution file for working with the HTTPService component.

Earlier, you used this component to retrieve static XML content for your application.

In this exercise, you will retrieve XML content from a dynamic ColdFusion page.

You could just as easily have referenced a PHP, JSP or ASP page.

HTTPService does not work with typed data, so any service page that can generate and return XML will work.

This training series is software-agnostic, since the Flex framework can interface with any server-side technologies that can work with HTTP service, web services or AMF.

The examples that I will use in the videos and that you will use in the associated exercises will use a ColdFusion server, but that is incidental.

You can use PHP, Java, ColdFusion, .NET or any other server technology.

The way that the returned data is used from these server technologies in Flex is the same.

If you want to explore more server-specific technologies for how they work with Flex, remember that you can do so by accessing the Adobe Developer Connection and then selecting one of the Integrate Flex links.

Back in Flash Builder, let's take a look at the type and file size for the returned data from this service call.

I am minimizing my Editor view so that I canaccess the Network Monitor view and enable this feature.

I am running the application to view it in the browser and then returning to Flash Builder to look at the audit trail.

As you learned in the last video, the Network Monitor gives you much useful information.

I am selecting the HTTPService call and then making sure the Response tab is selected.

You can see that the response size is 7264 bytes and that the response body is XML data.

This is the solution file for the WebService exercise.

I am running through the same steps to view the response data in the Network Monitor.

You can see that the data for a web service call is also XML, but the file size is much larger at 14,088 bytes.

The XML data for the HTTPService call is on the left and the XML data for the WebService call is on the right.

You can see that the SOAP-compliant WSDL document is much more verbose than the simple XML data returned for the HTTPService call.

This is the solution file for the RemoteObject exercise.

Again, I am running through the same steps to view the response data in the Network Monitor.

You can see that the data for RemoteObject call is in AMF format and is much smaller than either the HTTPService or WebService data packets.

The HTTPService component is very simple to use since the service call just needs to return generic XML data.

However, it cannot work with complex data types.

The WebService component is a great option if you are working with a SOAP-compliant service, but the XML files created by this method can be quite large.

Whenever possible, consider using the RemoteObject component, since it works with smaller binary packets and can directly access native objects on the server.

You will learn more about using these RPC components to retrieve data and handle faults in the in the next few videos.

You will learn how to send data to the server in Day 3 of training.

The last topic of this video is the crossdomain.xml file.

When you access RPC services with the RPC components, you are connecting to the service directly, which means that you might run into some security issues.

The RPC service must be in the same domain as the application that calls it.

If it is not, you must create a crossdomain.xml file that explicitly lists the domains that can access the services.

You must literally create an XML file named crossdomain.xml and place it on your server.

For instance, this example shows that applications in the macromedia.com and adobe.com domains can access the services on the server that contains this cross domain file.

For your next step, you can watch one, or all of the videos on “Retrieving and handling data” for HTTPService, WebService and RemoteObject.