

Adobe AIR Web Service Hello World test application

Posted At : 2 April 2009 10:55 | Posted By : Shaun McCran
Related Categories: Flex Remoting, Coldfusion, AIR, RIA, Flex

I've recently been looking at putting together some AIR applications. I've used FLEX for a few years now, and have only just come up with some *useful* AIR ideas, so I thought I'd build an application or two.

Usually I would use flash remoting, but I haven't spent too much time investigating how this works in AIR, so I've opted for the old school Web Service.

In the middle of my newest AIR application I stumbled upon an issue. No matter what I did I was receiving a 'HTTP Error' response from my Web Service. After stumbling around in the dark for a while tweaking code to no avail, I decided to write the most basic Web Service I could think of.

So here is 'Hello World', as a Web Service call from AIR.

Firstly create a call to your Web Service. In this case it was a local file. Point the wsdl variable at the fully qualified path to your service. I am using a coldFusion back end, so it is a CFC. This is also where you specify the fault handler and result handlers. You can add as many 'operation' methods here as you want, that way you address specific functions in your service individually.

```
d="getMessages" wsdl="http://192.168.XX.XXX/root/services/message.cfc?wsdl" showBu
:mx:operation name="sayHello" fault="faultHandler(event)" result="resultsHandler(e
</mx:WebService>
<mx:Button x="10" y="10" label="Click me" click="getData()"/>

</mx:WindowedApplication>
```

I've also added a button that will call a function to action the service call.

Next we will add the functions.

```
mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"

    <mx:Script>
        <![CDATA[
            import mx.rpc.events.ResultEvent;
            import mx.rpc.events.FaultEvent;
            import mx.controls.Alert;

            private function getData():void{
                getMessages.sayHello();
            }

            private function faultHandler(event:FaultEvent):void
            {
                Alert.show("Error: " + event.fault.faultString, "Application Error")
            }

            private function resultsHandler(event:ResultEvent):void
            {
                Alert.show(String(event.result))
            }

        ]]>
    </mx:Script>
```

A getter function, that will actually send the Service request, a fault handler that will simply Alert the user to a fault event, and a result handler that Alerts the user to whatever message is returned from the Web Service.

The CFC

The CFC service is a simply object to return a string. Just make sure that your 'Access' is set to remote.

```
<cfcomponent displayname="message">
```

```
= "sayHello" displayname="sayHello" hint="it says hello" access="remote" output="tr  
    <cfset var message = "Hello world">  
    <cfreturn message/>  
    </cffunction>  
    </cfcomponent>
```

So we end out with:

