

Invoking adapter procedures from native Windows Phone 8 applications

Overview

To create and configure a Windows Phone 8 native project, first follow the "Creating your first Native WP8 MobileFirst application (../hello-world/creating-first-native-windows-phone-8-mobilefirst-application/)" tutorial.

Initializing WLClient

```
1 | WLClient client = WLClient.getInstance();
```

1. To establish a connection to MobileFirst Server, use the `connect` method by specifying the `MyConnectResponseListener` class instance as a parameter.

```
1 | client.connect(new MyConnectResponseListener(this));
```

The `WLClient` instance tries to connect to the MobileFirst Server instance according to the properties of the `wlclient.properties` file.

After the connection is established, it invokes one of the methods of the `MyConnectResponseListener` class.

2. Specify that the `MyConnectResponseListener` class implements the `WLResponseListener` interface.

```
1 | public class MyConnectResponseListener : WLResponseListener
```

The `WLResponseListener` interface defines two methods:

- `public void onSuccess (WLResponse response) { }`
- `public void onFailure (WLFailResponse response) { }`

3. Use the previous methods to process connection success or connection failure.

Invoking an adapter procedure

After the connection is established with a MobileFirst Server instance, you can use the `WLClient` instance to invoke adapter procedures.

1. Create a `WLProcedureInvocationData` object with the adapter and procedure names.
2. Add the required parameters as an object array and set request options (for example: Invocation Context).
3. Get the existing `WLClient` instance and use it to invoke an adapter procedure.
4. Specify the `MyInvokeListener` class instance as a parameter.

```
1 | WLProcedureInvocationData invocationData = new WLProcedureInvocationData("RSSReader", "getStories");
2 | invocationData.setParameters(new Object[]{});
3 | String myContextObject = "InvokingAdapterProceduresWP8";
4 | WLRequestOptions options = new WLRequestOptions();
5 | options.setInvocationContext(myContextObject);
6 | WLClient.getInstance().invokeProcedure(invocationData, new MyInvokeListener(this), options);<br />
```

Receiving a procedure response

After the procedure invocation is completed, the `WLClient` instance calls one of the methods of the `MyInvokeListener` class.

As before, you must specify that the `MyInvokeListener` class implements the `WLResponseListener` interface.

```

1  using IBM.Worklight;
2  namespace InvokingAdapterProceduresWP8{
3  public class MyInvokeListener : WLResponseListener
4  {
5  {

```

The `onSuccess` and `onFailure` methods are invoked by the `WLClient`. The response object contains the response data. You can use its methods and properties to retrieve the required information.

```

1  public void onSuccess(WLResponse response)
2  {
3      WLProcedureInvocationResult invocationResponse = ((WLProcedureInvocationResult) response);
4      JObject items;
5      try
6      {
7          items = invocationResponse.getResponseJSON();
8          Deployment.Current.Dispatcher.BeginInvoke(() =>
9          {
10             myMainPage.AddTextToReceivedTextBlock("Response Success: " + items.ToString());
11             });
12         }
13     catch (JsonReaderException e)
14     {
15         Deployment.Current.Dispatcher.BeginInvoke(() =>
16         {
17             myMainPage.AddTextToReceivedTextBlock("JSONException : " + e.Message);
18             });
19         }
20     }
21 public void onFailure(WLFailResponse response)
22 {
23     Deployment.Current.Dispatcher.BeginInvoke(() =>
24     {
25         myMainPage.AddTextToReceivedTextBlock("Response failed: " + response.ToString());
26     });
27 }

```

Sample application

The sample contains two projects:

- The **InvokingAdapterProceduresNativeProject.zip** file contains a MobileFirst Native API to deploy to MobileFirst Server.
- The **InvokingAdapterProceduresWP8project.zip** file contains a native Windows Phone 8 application that uses a MobileFirst native API library to communicate with a MobileFirst Server instance.

Make sure to update the **wlclient.properties** file in **InvokingAdapterProceduresWP8** with the relevant server settings.

Click to download

(<http://public.dhe.ibm.com/software/products/en/MobileFirstPlatform/docs/v630/InvokingAdapterProceduresNativeProject.zip>)
the Studio project.

Click to download

(<http://public.dhe.ibm.com/software/products/en/MobileFirstPlatform/docs/v630/InvokingAdapterProceduresWP8Project.zip>)
the Native project.

