Resource Request from Native Android Applications

RENAMING

Overview

MobileFirst applications can access resources using the `WLResourceRequest` REST API.
The REST API works with all adapters and external resources LINK TO using-mobilefirst-server-authenticate-external-resources.
This tutorial explains how to use the `WLResourceRequest` API with an HTTP adapter.

To create and configure an Android native project, first follow the Adding the MobileFirst Platform Foundation SDK to Android Applications (../../adding-the-mfpf-sdk/adding-the-mfpf-sdk-to-android-applications) tutorial.

Initializing WLClient

WLCLIENT.CONNECT

1. Create an instance of the `WLClient` class.
   The `WLClient` instance requires a reference to the activity in which it is running.

   ```
   WLClient client = WLClient.createInstance(context);
   ```

2. To establish a connection to the MobileFirst Server instance, use the `connect` method by specifying the `MyConnectListener` class instance as a parameter.
   The `WLClient` instance tries to connect to the MobileFirst Server according to the properties of the `mfpclient.properties` file.
   After the connection is established, it invokes one of the methods of the `MyConnectListener` class.

3. Specify that the `MyConnectListener` class implements the `WLResponseListener` interface.

   ```
   public class MyConnectListener implements WLResponseListener  {
   }
   ```

   The `WLResponseListener` interface defines two methods:

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

   Use these methods to process connection success or connection failure.

Calling an adapter procedure

The `WLResourceRequest` class handles resource requests to MobileFirst adapters or external resources.

1. Define the URI of the resource:

   ```
   URI adapterPath = new URI("/adapters/RSSReader/getFeed");
   ```

- For JavaScript adapters, use `/adapters/{AdapterName}/{procedureName}`
- For Java adapters, use `/adapters/{AdapterName}/{path}`
- To access resources outside of the project, use the full URL

2. Create a `WLResourceRequest` object and choose the HTTP Method (GET, POST, etc):

```
WLResourceRequest request = new WLResourceRequest(adapterPath,WLResourceRe
quest.GET);
```

3. Add the required parameters:

   - In JavaScript adapters, which use ordered nameless parameters, pass an array of parameters with the name `params`:

```
request.setQueryParameter("params","['MobileFirst_Platform']");
```

   - In Java adapters or external resources, use the `setQueryParameter` method for each parameter:

```
request.setQueryParameter("param1","value1");
request.setQueryParameter("param2","value2");
```

4. Call the procedure by using the `.send()` method.
   Specify a `MyInvokeListener` class instance:

```
request.send(new MyInvokeListener());
```

> See the user documentation to learn more about `WLResourceRequest` and other signatures for the `send` method, which are not covered in this tutorial.

Receiving a procedure response

When the procedure invocation is completed, the framework calls one of the methods of the `MyInvokeListener` class.

1. Specify that the `MyInvokeListener` class implements the `WLResponseListener` interface:

```
public class MyInvokeListener implements WLResponseListener {
}
```

2. Implement the `onSuccess` and `onFailure` methods.
   If the procedure invocation is successful, the `onSuccess` method is called. Otherwise, the `onFailure` method is called. Use these methods to get the data that is retrieved from the adapter.
   The `response` object contains the response data and you can use its methods and properties to retrieve the required information.

```
public void onSuccess(WLResponse response) {
      String responseText = response.getResponseText();
      AndroidNativeApp.updateTextView("Adapter Procedure Invoked Successfu
ly\n" + responseText);
}

public void onFailure(WLFailResponse response) {
     String responseText = response.getResponseText();
     AndroidNativeApp.updateTextView("Failed to Invoke Adapter Procedure\n
" + responseText);
}
```

Sample application

Click to download (https://github.com/MobileFirst-Platform-Developer-Center/InvokingAdapterProcedures) the MobileFirst project.

Click to download (https://github.com/MobileFirst-Platform-Developer-Center/InvokingAdapterProceduresAndroid) the Native project.

- The `InvokingAdapterProcedures` project contains a MobileFirst Native API to deploy to your MobileFirst Server instance.
- The `InvokingAdapterProceduresAndroid` project contains a native Android application that uses a MobileFirst native API library to communicate with MobileFirst Server.
- Make sure to update the `mfpclient.properties` file in the native Android project with the required server settings.

SCREENSHOT