

Android - Using native pages

Overview

In this tutorial, integration of native and web "pages" in an Android application will be explained by using the `WL.NativePage.show()` API to open a native page from JavaScript. With this method, data can be sent from JavaScript to the opened native page, and specify a callback to call after the native page closes.

Connecting to the plugin from the JavaScript code

As a first step, `WL.NativePage.show()` needs to be implemented in order to open the native page:

```
1 function openNativePage(){
2     var params = {
3         nameParam : $('#nameInput').val()
4     };
5     WL.NativePage.show(nativePageClassName, backFromNativePage, params);
6 }
```

- `nativePageClassName`: The name of a native Android Activity to start.
- `backFromNativePage`: A callback function to call when the native page closes.
- `params`: optional custom parameters object to pass to the native code.

To handle the callback function:

```
1 function backFromNativePage(data){
2     alert("Received phone number is: " + data.phoneNumber);
3 }
```

- `backFromNativePage(data)`: After the native closes, it can pass data back to the web part of an application.

Creating a native page

In Android, the native page must be implemented as an Android Activity, or extend an existing Activity. **Step 1** As with any Activity, it must be declared in the `AndroidManifest.xml` file. For example:

```
1 <activity android:name=".HelloNative"/><activity>
```

Step 2 To retrieve custom data parameters that are passed from the web view, an `intent` should be used:

```
1 protected void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3
4     String name = getIntent().getStringExtra("nameParam");
```

Returning control to the web view

When the native page switches back to the web view, the `finish()` function is called for the Activity. To pass data back to the web view can be done by using an `Intent` object. For example: **Java**:

```
1 String phoneNumber = editText.getText().toString();
2 Intent phoneNumberInfo = new Intent();
3 phoneNumberInfo.putExtra("phoneNumber", phoneNumber);
4 setResult(RESULT_OK, phoneNumberInfo);
5 finish();
```

JavaScript:

```
1 function backFromNativePage(data){
2     alert("Received phone number is: " + data.phoneNumber);
3 }
```

Sample application

Click to download

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

