

Native B4A and B4J skeleton projects for ABMaterial WebApps using the WebView

Setting some properties in ABMApplication allows ABMaterial to generate a Skeleton project for B4A and B4J using a webview. I would love to make it for B4i too, but my knowledge of creating iOS libraries is limited. If someone feels up to the challenge, please contact me by mail (alain.bailleul@telenet.be). The library itself is pretty simple, and I have some ideas on how we could implement the raise event from the WebView -> B4i.

Notes:

The webview does not 100% support everything used in ABMaterial!

So far, it does a pretty good job, but it is for example not capable to load the ABMPivot component. Probably it has problems with the iFrame in it. ABMaterial will also **NOT** be changed in favour for WebView. Normal web browsers like Chrome, Firefox, Safari and mobile web browsers stay **TOP PRIORITY**

These apps are by no way a replacement for the real B4A/B4J native apps!

Real B4A/B4J apps are a lot more powerful and have a much better user experience than the apps generated by ABMaterial. There is also a good chance such app will be refused by the App Stores so only use this if really needed. In our company, it is for example useful to write a small tool which takes a picture and uploads it to the ftp using the ABMaterial interface.

What to do in your ABMaterial WebApp:

Select the library **ABMControllerB4J** (see zip for the library, copy in your libraries).

In ABMApplication, you have to set some Native properties and call **ABM.NativeGenerateApps()**

```
ABM.Native.Initialize("com.abmaterial.proj2102", "proj2102", "Manual Registration",  
"http://prd.one-two.com:51042/proj2102")
```

```
ABM.Native.VersionCode = "1"  
ABM.Native.VersionName = ""  
ABM.Native.AndroidIcon72x72PngPath = "Z:/icon.png"  
ABM.Native.DesktopIcon256x256PngPath = "Z:/temp/icon.png"  
ABM.Native.NoConnectionCancel = "Cancel"  
ABM.Native.NoConnectionRetry = "Retry"  
ABM.Native.NoConnectionMessage = "This application needs an internet connection!"  
ABM.Native.QuitMessage = "Do you want to quit this application?"  
ABM.Native.QuitYes = "Yes"  
ABM.Native.QuitNo = "No"
```

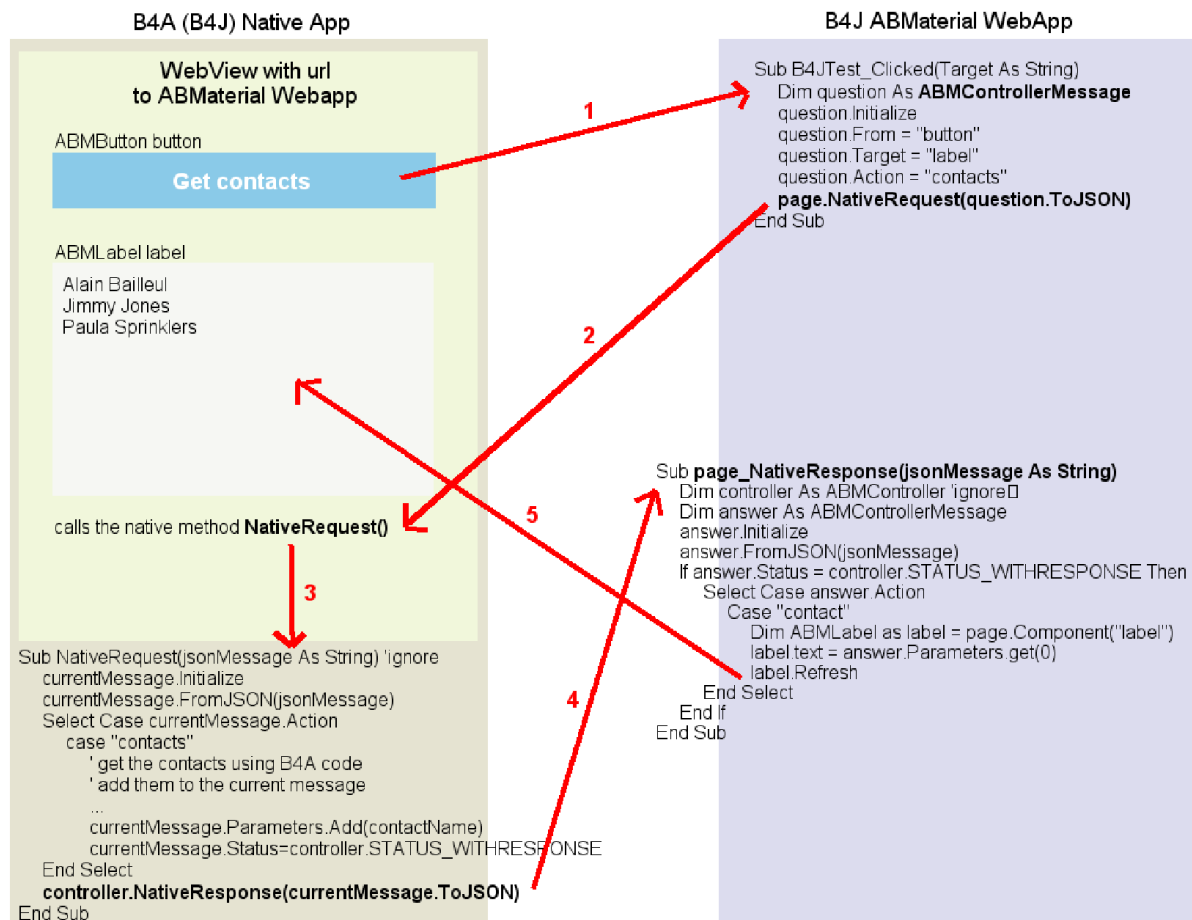
```
' Do Not FORGET To DISABLE THIS LINE If YOU WANT To MAKE CHANGES To THE APPS!  
ABM.NativeGenerateApps(False)
```

In the page where you want to respond to an answer from the native app, add the **page_NativeResponse** event.

When you now run your ABMaterial WebApp, a folder named **NativeApps** will be created next to the Files and Objects folder of your ABMaterial B4J app. It contains two new projects: one for B4A and one for B4J.

NOTE: For the B4A app you will need to copy the **ABMControllerB4A** library to your B4A libraries folder! (see zip for the library)

How it works:



1. So when you press e.g. a button in the WebView, the normal event is raised to your WebApp on the server. Here you can do some checks, and finally build a **ABMControllerMessage**.

The ABMControllerMessage has just a couple of properties:

From: a String where you can set the origin of the message

Target: a String where you can set the target of the message

Action: a String where you can define the 'method' to call. You can see this as your internal list of methods.

Status: a status field, use the ABMController.STATUS_ constants to check if your message is ok

Parameters: a List of Strings where you can add parameters of your Method, or return values to the query.

2. Send the message back to the WebView, it will be automatically forwarded to the Native B4A/B4J method **NativeRequest()**. Here you can act according to your 'Action' you defined in the message.
3. This is where the real power is! You can do everything here a native B4A/B4J app does, use all its libraries, access all device things like camera, gps, etc...
4. Finally, use the message to define a response. You can clear the parameters, add new ones etc. and send it back to the server.
5. The server will receive the response in **page_NativeResponse()**. Again, you can use the 'Action' to handle the response and e.g. do some normal ABMaterial things like changing a labels contents.