iOS - Using native pages

Overview

In this tutorial, integration of native and web "pages" in an iOS 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:

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

- `nativePageClassName`: The name of a native iOS UIViewController instance 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:

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

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

Creating a native page

To manage to native page, the generated iOS projects need to be opened in Xcode and afterwards:

Step 1

Add a new Cocoa Touch Class file, make sure it is a subclass of `UIViewController`, and save it in the `Classes` folder of the Xcode project.

Important:
If working with existing .m and .h files, the files must be referenced while in Xcode.
Placing the .m and .h files only in the `iphone\native\Classes>` folder in Eclipse is not sufficient, as these files will not be referenced in the Xcode project unless added in Xcode.

Step 2

To retrieve custom data parameters that are passed from the web view, the `setDataFromWebView:(NSDictionary*)data` method should be used:

```
-(void)setDataFromWebView:(NSDictionary*)data{
    self.nameParam = (NSString*)[data valueForKey:@"nameParam"]
;
}
```

Returning control to the web view

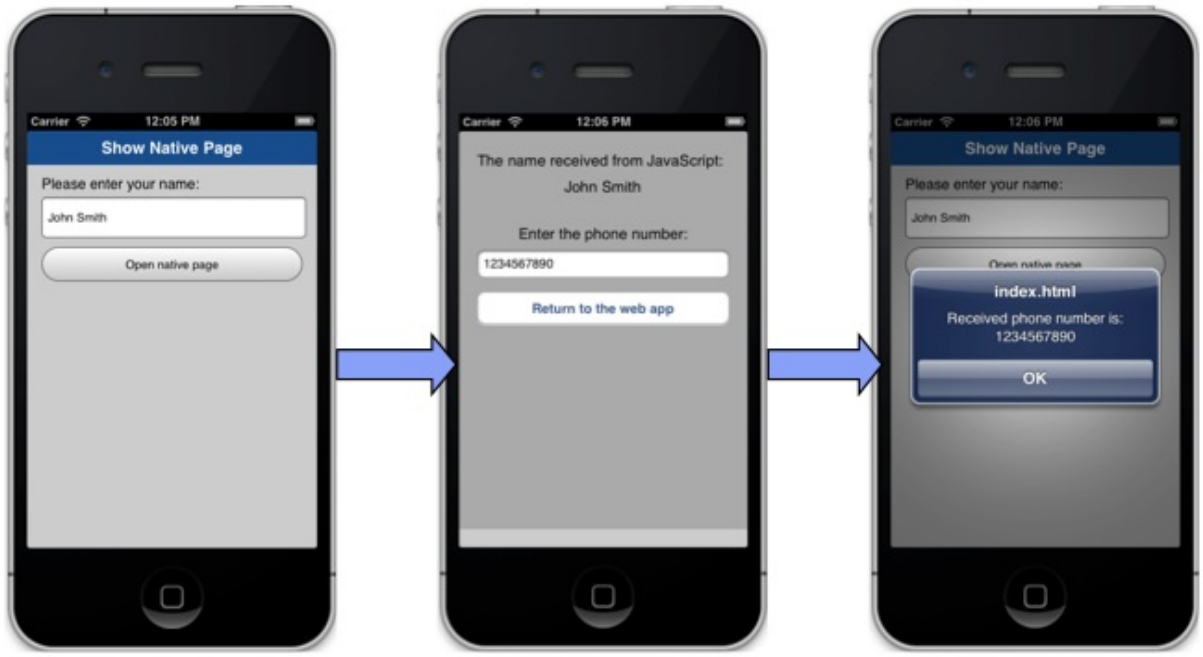
When the native page switches back to the web view, a `[NativePage showWebView:]` method should be called. To pass data back to the web view can be done by using the `NSDictionary` object. For example:

Objective-C:

```
-(IBAction)returnClicked:(id)sender{
    NSString *phone = [phoneNumber text];
    NSDictionary *returnedData = [NSDictionary dictionaryWithObject:phone forKey:@"phoneNumber"]
;
    [NativePage showWebView:returnedData];
}
```

JavaScript:

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

Sample application

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