JavaScript with iOS7

Blurring the line between JS and Obj-C

By Oscar Perez

April, 2014

In the past...

- JS Environment in UIWebView
- **JS to Obj-C:**
  Intercepting UIWebView URL Changes
- **Obj-C to JS:**
  stringByEvaluatingJavaScriptFromString:

In the past...

- Hybrid Frameworks: PhoneGap/Cordova

- Bridges/Bindings: Titanium, Xamarin, Ruby Motion

Introducing JavaScriptCore in iOS7

- JS bridge by Apple
- Makes JS an officially supported language in iOS

JavaScriptCore Details

- **JSContext:** Isolated JS execution environment.

  Holds your globals, objects, values, functions, etc.

JavaScriptCore Details

- **JSValue:** Bridges JS and Obj-C types.

undefined -> nil

null -> NSNull

number -> NSNumber
array -> NSArray
string -> NSString
object -> NSDictionary
function -> NSBlock

JavaScriptCore Details

- **JSExport:** Allows Obj-C objects to have methods invokable from JavaScript.

JavaScriptCore Example

```
JSContext *ctx = [[JSContext alloc] init];
JSValue *res = [ctx evaluateScript:@”\”hello\” + \”world\””];
NSLog(@”result: %@”, [res toString]); // “result: helloworld”
```

Getting UIWebView’s JSContext

- Create an NSObject category that implements:

```
- (void)webView:(id)opaqueWebView didCreateJavaScriptContext:(JSContext
*)jsContext forFrame:(id)opaqueFrame;
```

- Save the jsContext somewhere and track where it’s saved within the jsContext:

```
somewhere[contextKey] = jsContext;
jsContext[@"JSContextTrackerMapKey"] = contextKey;
```

Defining JS functions from Obj-C

- Retrieve the stored UIWebView JSContext
- Define your function:

```
- (void)webViewDidFinishLoad:(UIWebView *)targetWebView {
    self.webViewJSContext = [NSObject contextForWebView:
webView];
    self.webViewJSContext[@"console"][@"log"] = ^(JSValue*
value) {
        NSLog(@"%@", [value toString]);
    };
}
```

Shooting yourself in the foot

- Don’t do this:

```
JSValue *someFunction = self.webViewCtx[@”someObj”][@”
someFunc”];
```

```
[someFunction callWithArguments:@[1, @”myArg2”]];
```

- Do this:

```
[self.webView stringByEvaluatingJavaScriptFromString:@”someObj.
someFunc(1, \”myArg2\”);”];
```

Summary

- JavaScriptCore APIs in iOS 7 = Apple’s JS to Obj-C Bridge.
- Native functionality from JS and vice-versa without a third-party library.
- Be careful when dealing with UIWebView’s JSContext.

Questions