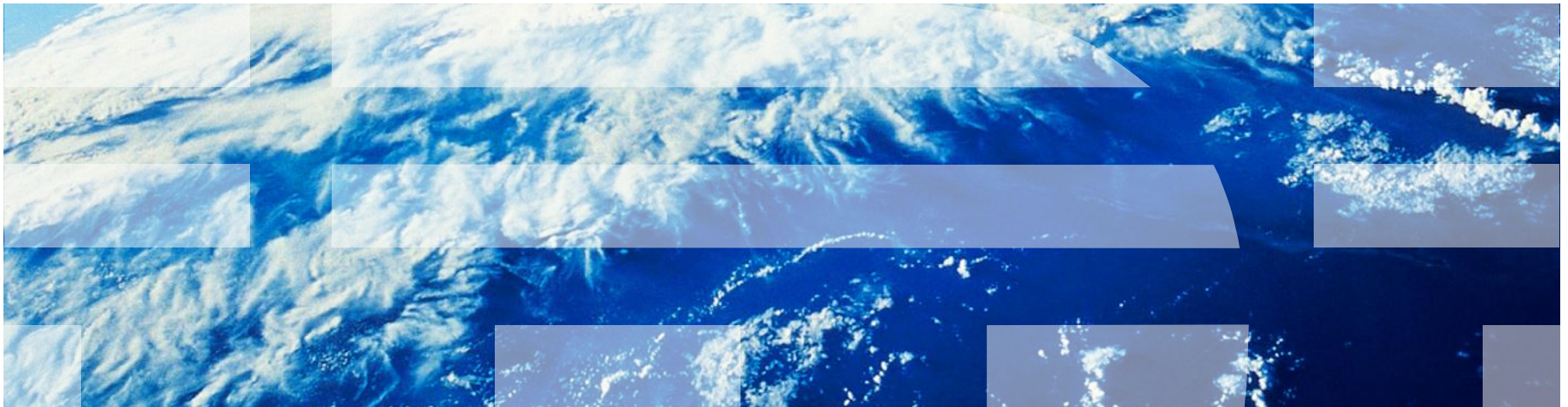# *IBM Worklight Foundation V6.2.0*
Getting Started

Adding Native Android UI Elements In Hybrid Applications

Trademarks

- IBM, the IBM logo, ibm.com, and Worklight are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

- Other company products or service names may be trademarks or service marks of others.

- This document may not be reproduced in whole or in part without the prior written permission of IBM.

About IBM®

- See http://www.ibm.com/ibm/us/en/

Agenda

- Overview

- Startup Flow

- Native SplashScreen Sample

- Send Action From JavaScript to Native

- Send Action From Native to JavaScript

- SendAction Sample

- Shared Session

Overview

- While you could write your Hybrid application by using solely web technologies, with IBM Worklight® Foundation® you can also mix and match native code with web code as needed.

- For example, you can choose to use some native UI controls, provide an animated native introduction screen, use native elements provided by iOS or Android, etc.

- To do so, you need to take control of part of the startup flow of your hybrid application.

- This tutorial assumes that you have a working knowledge of native Android development.

Agenda

- Overview

- Startup Flow

- Native SplashScreen Sample

- Send Action From JavaScript to Native

- Send Action From Native to JavaScript

- SendAction Sample

- Shared Session

Startup Flow

- When you create a new Hybrid application, Worklight creates and generates a main CordovaActivity class for you, handling various stages of the application startup flow. Open that file (/Android/native/src/..<Appname>.java) and study the default flow.
- Worklight starts by creating an instance of its main CordovaActivity class.
- The `ShowSplashScreen` method is called to display a simple splash screen while the Worklight framework is being initialized and resources are being loaded. ***Here, you can replace this line with any native introduction screen.***
- To initialize the Worklight framework and prepare web resources, the `initializeWebFramework` method is called.
- As soon as the web framework finished initializing and all resources are ready, the `onInitWebFrameworkComplete` method is called. You check the value of `WLInitWebFrameworkResult` and start the application.

Agenda

- Overview

- Startup Flow

- Native SplashScreen Sample

- Send Action From JavaScript to Native

- Send Action From Native to JavaScript

- SendAction Sample

- Shared Session

Native SplashScreen Sample (1 of 2)

1. Download the NativeUIInHybrid project, which includes a hybrid application called NativeSplashScreen.

2. Create a new Activity and call it InitiativeActivity.

- This Activity is used to show a simple TextView and a Button as our customized splash screen.

3. Add this code to the `onCreate` method:

```
setContentView(R.layout.activity_initiative);
startAppBtn = (Button) findViewById(R.id.StartApp);
```

4. Create the `onClickListener` method for the button:

```
startAppBtn.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
      setResult(Activity.RESULT_OK);
      finish();
}
});
```

Native SplashScreen Sample (2 of 2)

5. Add a TextView and a button to the layout file of this Activity:

```xml
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/InitiativeActivityText" />

<Button
        android:id="@+id/StartApp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="150dp"
        android:text="@string/InitiativeButtonText" />
```

6. Add an Intent to the MainActivity before the call to the `initializeWebFramework` method.

The intent loads the newly created Activity instead of opening the default Worklight splash screen.

```java
WL.createInstance(this);

Intent intent = new Intent(this, InitiativeActivity.class);
startActivity(intent);

WL.getInstance().initializeWebFramework(getApplicationContext(), this);
```

Agenda

- Overview

- Startup Flow

- Native SplashScreen Sample

- Send Action From JavaScript to Native

- Send Action From Native to JavaScript

- SendAction Sample

- Shared Session

Send Action From JavaScript to Native (1 of 2)

- With Worklight, you can send commands with parameters from the WebView (via JavaScript) to an Android native class (written in Java).

- Use this feature to trigger native code to be run in the background, to update the native UI, use native-only features, etc.

- From JavaScript, write `WL.App.sendActionToNative("doSomething", {customData: 12345} );`
where `doSomething` is an arbitrary action name to be used on the native side (see the next slide) and the second parameter is a JSON object that contains any data.

Send Action From JavaScript to Native (2 of 2)

- The native class to receive the action must implement the `WLActionReceiver` protocol.

```java
public class ActionReceiver implements WLActionReceiver{
}
```

- The `WLActionReceiver` protocol requires an `onActionReceived` method in which you check the action name and perform any native code that this action needs.

```java
public void onActionReceived(String action, JSONObject data){
if (action.equals("doSomething")){
    // Write you code here...
}
}
```

- For your action receiver to receive actions from the Worklight web view, you must register it. You can register it during the startup flow of the application to catch any actions early enough.

```java
WL.getInstance().addActionReceiver(new ActionReceiver(this));
```

Agenda

- Overview

- Startup Flow

- Native SplashScreen Sample

- Send Action From JavaScript to Native

- Send Action From Native to JavaScript

- SendAction Sample

- Shared Session

Send Action From Native to JavaScript (1 of 2)

- In Worklight applications, you can send commands with parameters from Android native Java code to web view JavaScript code.

- Use this feature to receive responses from a native method, notify the web view when a background code finished running, allow native UI to control the content of the web view, etc.

1. From Android native code, write

```
JSONObject data = new JSONObject();
data.put("someProperty", 12345);
WL.getInstance().sendActionToJS("doSomething", data);
```

 where `doSomething` is an arbitrary action name to be used on the JavaScript side (see the next slide) and the second parameter is a JSONObject that contains any data.

Send Action From Native to JavaScript (2 of 2)

2. Write a JavaScript function, with the name of your choice, which verifies the action name and implements any JavaScript code.

```
function actionReceiver(received){
    if (received.action == "doSomething" && received.data.someProperty ==
    "12345"){
      //perform required actions, e.g., update web user interface
    }
}
```

3. Register this JavaScript function to receive actions. Do it early enough in your JavaScript code to handle those actions as early as possible.

```
WL.App.addActionReceiver ("MyActionReceiverId", actionReceiver);
```
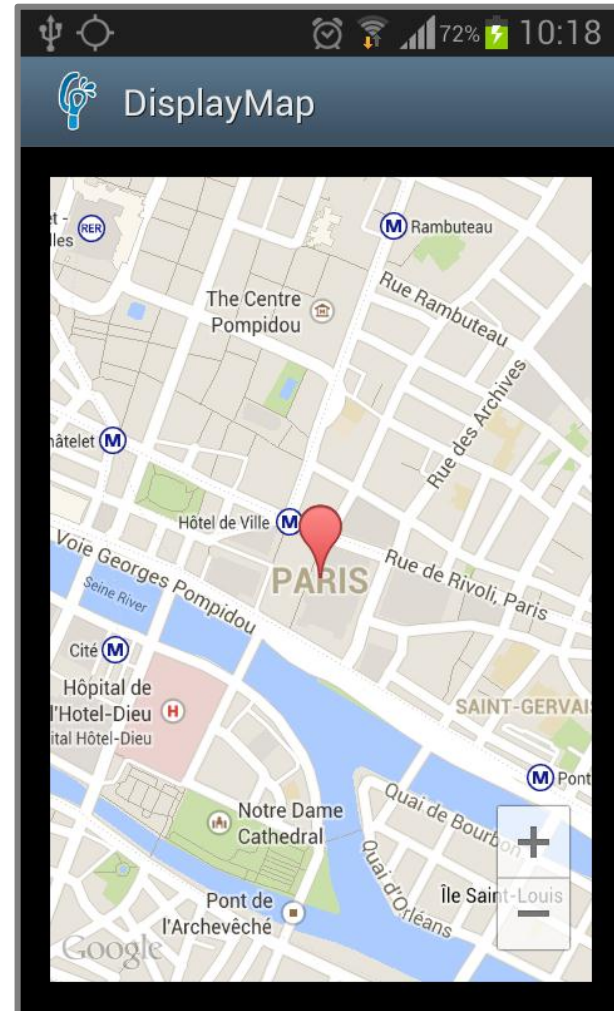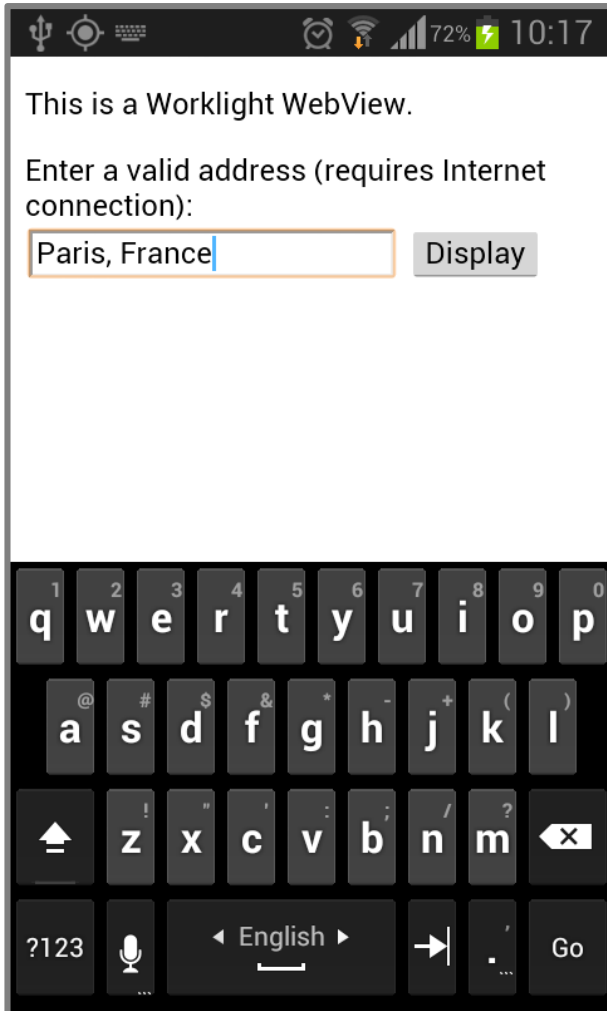
- The first parameter is an arbitrary name. You can use it later to remove an action receiver.

```
WL.App.removeActionReceiver("MyActionReceiverId");
```

Agenda

- Overview

- Startup Flow

- Native SplashScreen Sample

- Send Action From JavaScript to Native

- Send Action From Native to JavaScript

- SendAction Sample

- Shared Session

SendAction Sample - Preview

SendAction Sample - Overview

- Download the NativeUIInHybrid project, which includes a hybrid application called SendAction.

- This sample uses a MapView and GoogleMap services.
  For more information about MapView and GoogleMap services, see the Google Map Android page at
  https://developers.google.com/maps/documentation/android/

- This sample includes an additional FragementActivity (DisplayMap) to display the requested map.

SendAction Sample - HTML

- The HTML page the following elements:

  - A simple input field to enter an address.

  - A button to trigger validation.

  - An empty <p> line to show potential error messages.

```
<p>This is a Worklight WebView.</p>
<p>Enter a valid address (requires Internet connection):<br/>
      <input type="text" name="address" id="address"/>
   <input type="button" value="Display" id="displayBtn"/>
 </p>
 <p id="errorMsg" style="color:red;"></p>
```

SendAction Sample - JavaScript

- When the button is clicked, `sendActionToNative` is called to transmit the address to the native code.

```
$('#displayBtn').on('click', function(){
$('#errorMsg').empty();
    WL.App.sendActionToNative("displayAddress",
            { address: $('#address').val()}
    );
});
```

- The code also registers an action receiver to display potential error messages from the native code.

```
WL.App.addActionReceiver ("MyActionReceiverId", function
actionReceiver(received){
    if(received.action == 'displayError'){
        $('#errorMsg').html(received.data.errorReason);
    }
});
```

SendAction Sample – Action receiver (1 of 2)

- Add a new class that implements `WLActionReceiver` and call it `ActionReceiver`.

- Before using this new class, call the `addActionReceiver` method to register `ActionReceiver` in the main Activity, after Worklight initialization process is complete.

```java
public void onInitWebFrameworkComplete(WLInitWebFrameworkResult result){
    if (result.getStatusCode() == WLInitWebFrameworkResult.SUCCESS) {
        super.loadUrl(WL.getInstance().getMainHtmlFilePath());
    } else {
        handleWebFrameworkInitFailure(result);
    }
    WL.getInstance().addActionReceiver(new ActionReceiver(this));
}
```

- The `ActionReceiver` class implements the `onActionReceived` method. Use this method to receive the address that was passed in the JavaScript code.

```java
public void onActionReceived(String action, JSONObject data){
if (action.equals("displayAddress")){
    try {
        mAddress = data.getString("address");
    } catch (JSONException e) {
        e.printStackTrace();
    }

    Intent intent = new Intent(parentActivity,DisplayMap.class);
    intent.putExtra("RECEIVED_ADDRESS", mAddress);
    parentActivity.startActivity(intent);
}
}
```

SendAction Sample – Display map

- After receiving the address parameter, create a new Fragment Activity and call it `DisplayMap`.

- This Activity uses the GoogleMap services and the Google Geocoder to display the requested address on a map.

- Send the address parameter that is just received in our `ActionReceiver` class to this Activity and use it to show the address on a MapView.

Agenda

- Overview

- Startup Flow

- Native SplashScreen Sample

- Send Action From JavaScript to Native

- Send Action From Native to JavaScript

- SendAction Sample

- Shared Session

Shared Session

- When using both JavaScript and native code in the same application, you might need to make HTTP requests to Worklight server (connect, procedure invocation, etc.).

- HTTP requests are explained in other tutorials (both for hybrid and native applications).

- IBM Worklight Foundation 6.2 and later keeps your session (cookies and HTTP headers) automatically synchronized between the JavaScript client and the native client.

Notices

- Permission for the use of these publications is granted subject to these terms and conditions.

- This information was developed for products and services offered in the U.S.A.

- IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

- IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

  – IBM Director of Licensing
  IBM Corporation
  North Castle Drive
  Armonk, NY 10504-1785
  U.S.A.

- For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

  – Intellectual Property Licensing
  Legal and Intellectual Property Law
  IBM Japan Ltd.
  1623-14, Shimotsuruma, Yamato-shi
  Kanagawa 242-8502 Japan

- **The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

- This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

- Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

- IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

- Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

  – IBM Corporation
  Dept F6, Bldg 1
  294 Route 100
  Somers NY 10589-3216
  USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Privacy Policy Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect session information (generated by the application server). These cookies contain no personally identifiable information and are required for session management. Additionally, persistent cookies may be randomly generated to recognize and manage anonymous users. These cookies also contain no personally identifiable information and are required.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent. For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacyIf the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent. For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/detailsIf the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent. For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the sections entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Support and comments

- For the entire IBM Worklight documentation set, training material and online forums where you can post questions, see the IBM website at:
  - http://www.ibm.com/mobile-docs
- **Support**
  - Software Subscription and Support (also referred to as Software Maintenance) is included with licenses purchased through Passport Advantage and Passport Advantage Express. For additional information about the International Passport Advantage Agreement and the IBM International Passport Advantage Express Agreement, visit the Passport Advantage website at:
    - http://www.ibm.com/software/passportadvantage
  - If you have a Software Subscription and Support in effect, IBM provides you assistance for your routine, short duration installation and usage (how-to) questions, and code-related questions. For additional details, consult your IBM Software Support Handbook at:
    - http://www.ibm.com/support/handbook
- **Comments**
  - We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.
  - For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.
  - When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state.
  - Thank you for your support.
  - Submit your comments in the IBM Worklight Developer Edition support community at:
    - https://www.ibm.com/developerworks/mobile/worklight/connect.html
  - If you would like a response from IBM, please provide the following information:
    - Name
    - Address
    - Company or Organization
    - Phone No.
    - Email address

Thank You