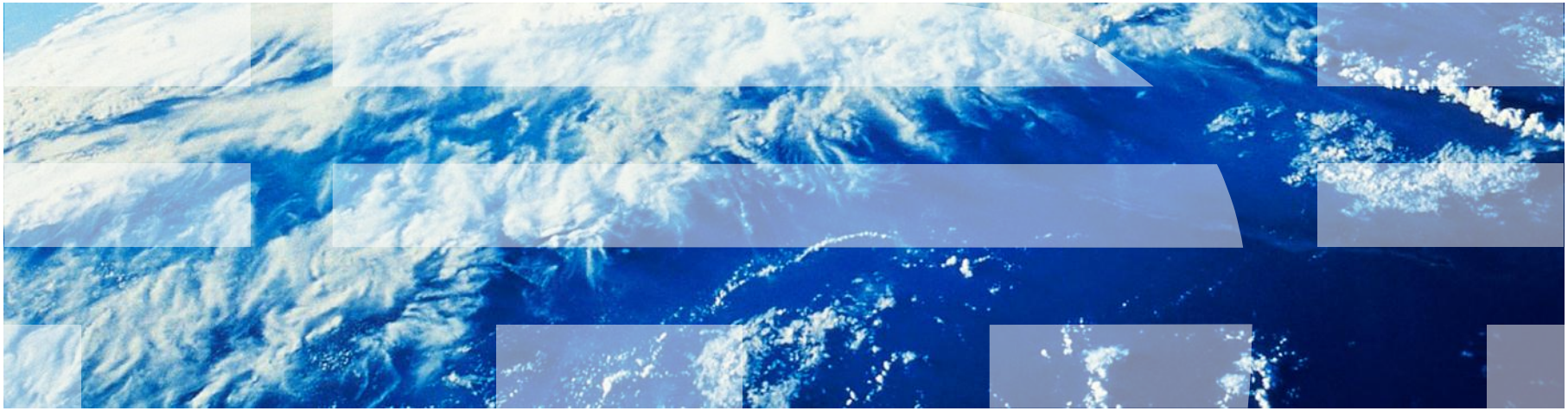
IBM Worklight Foundation V6.2.0 Getting Started

Common UI controls



20 June 2014

Trademarks

- IBM, the IBM logo, ibm.com, and Worklight are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Worklight is a trademark or registered trademark of Worklight, an IBM Company. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.
- Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- 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

- Common controls
  - What is a common control?
  - Busy indicator
  - Simple dialog
  - Tab bar
  - Options menu
  - Splash screen
- Sample

What is a common control?

- Some controls are common to most environments, such as modal pop-up windows, loading screens, and tab bars.

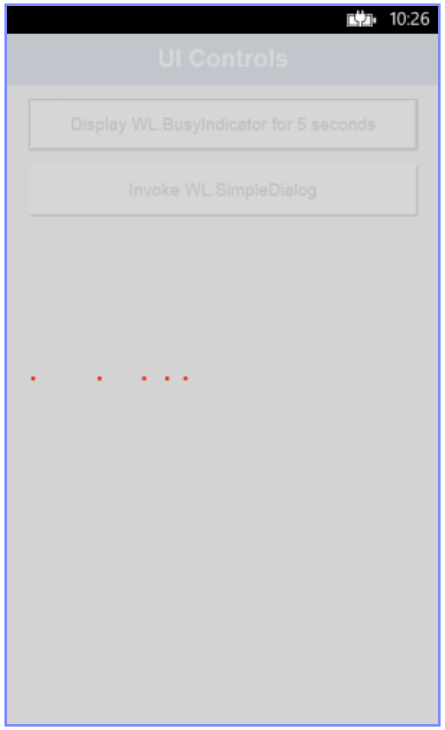
- With IBM Worklight® Foundation, you can use a JavaScript™ API to invoke these controls regardless of the environment. This API automatically renders these controls in a native way for each mobile platform.

Agenda

- Common controls
  - What is a common control
  - Busy indicator
  - Simple dialog
  - Tab bar
  - Options menu
  - Splash screen
- Sample

WL.BusyIndicator

- `WL.BusyIndicator` implements a common API to display a modal activity indicator.
- It uses native implementation on the following platforms: Android, iOS, BlackBerry 10, and Windows Phone 8.



WL.BusyIndicator – continued

- It must be initialized before use.

```
busyIndicator = new WL.BusyIndicator( null, {text : 'Loading...'});
```

- The parent element ID for `WL.BusyIndicator` is ignored in iOS, Android, Windows Phone, and BlackBerry environments.
- Available options are:
  - *text* – set the modal text.
  - *color* – set the text color.
  - *fullScreen* – should modal message be displayed full screen (iOS only).
  - For more information about the options, see the Worklight user documentation.

WL.BusyIndicator – continued

- `WL.BusyIndicator` provides the following API:
  - Initialization.

```
busyIndicator = new WL.BusyIndicator( null, {text : 'Loading...'});
```

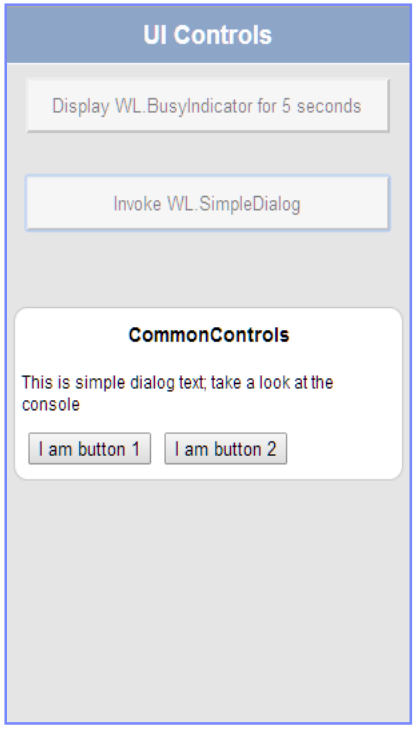
  - void `myBusyIndicator.show()` – displays busy indicator.
  - void `myBusyIndicator.hide()` – hides busy indicator.
  - boolean `myBusyIndicator.isVisible()` – returns whether the busy indicator is visible.

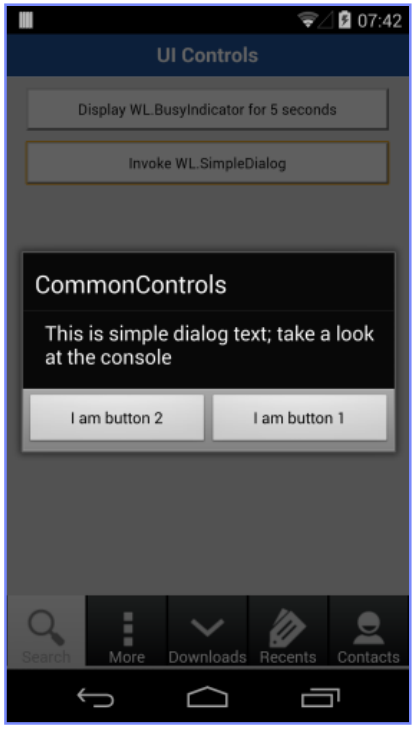
Agenda

- Common controls
  - What is a common control
  - Busy indicator
  - Simple dialog
  - Tab bar
  - Options menu
  - Splash screen
- Sample

WL.SimpleDialog

- The `WL.SimpleDialog` implements a common API for showing a modal dialog window with buttons.
- Uses native implementation on the following platforms: Android, iOS, Windows Phone 8, and BlackBerry 10.
- Adobe Air, BlackBerry 6/7, Desktop webpage, and Mobile Web use a JS-based implementation.





WL.SimpleDialog – continued

- The invocation syntax is:

```
WL.SimpleDialog.show(title, text, buttons, options);
```

- Parameters are `title`, `text`, and `buttons` as an array of button objects.
- The dialog is closed when any of the buttons is pressed.

WL.SimpleDialog – continued

- Each button object has two properties:
  - *text* – the text that displayed on the button.
  - *handler* – the function to invoke if the button is pressed.

```
var dialogTitle = "CommonControls";
var dialogText = "This is simple dialog text; take a look at the console";

WL.SimpleDialog.show(dialogTitle, dialogText, [
    {
        text : 'I am button 1',
        handler : simpleDialogButton1Click
    }, {
        text : 'I am button 2',
        handler : simpleDialogButton2Click
    }
]);
```

WL.SimpleDialog – continued

The following limitations exist:

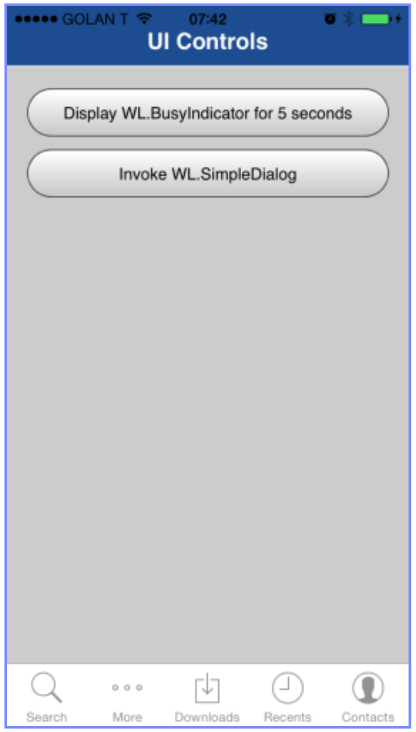
- When you use `WL.SimpleDialog` in **Windows Phone 8**, you can use at most four buttons in each instance of `WL.SimpleDialog`.
- When you use `WL.SimpleDialog` in **Android**, you can use at most three buttons in each instance of `WL.SimpleDialog`.

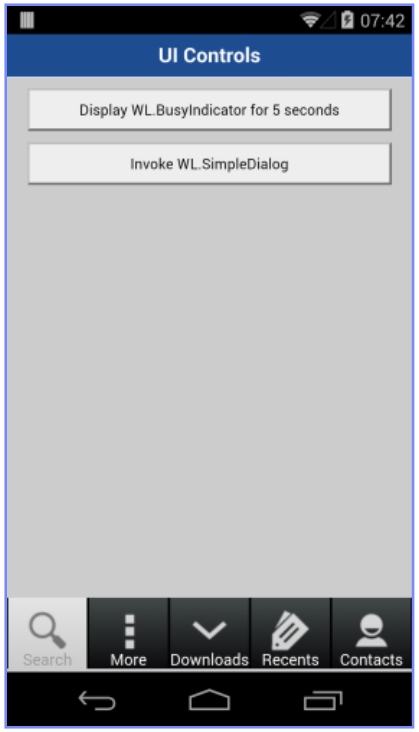
Agenda

- Common controls
  - What is a common control
  - Busy indicator
  - Simple dialog
  - Tab bar
  - Options menu
  - Splash screen
- Sample

WL.TabBar

- Supported environments: Android and iOS.
- Application navigation with a tab bar component.



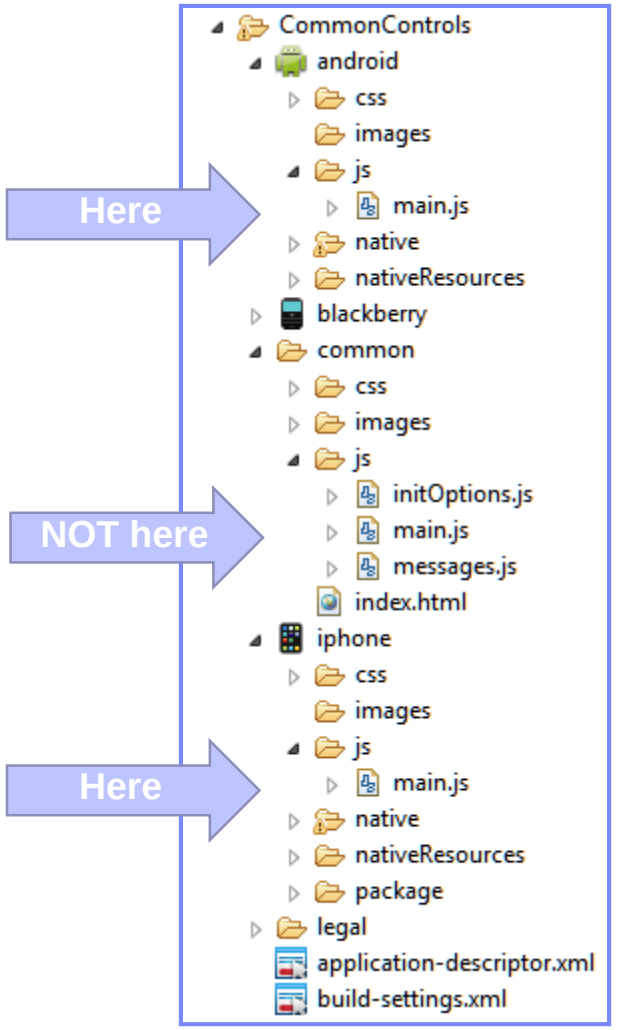


WL.TabBar – continued

- The iOS implementation uses a native component, but Android uses an HTML-generated tab bar.
- The syntax is similar, though with some minor differences.
- `WL.TabBar` must be initialized before it can be used.

```
WL.TabBar.init();
```

- Initialize `WL.TabBar` in the designated JavaScript file.



WL.TabBar – continued

- Use the following syntax to add a tab bar item:

```
WL.TabBar.addItem(id, callback, title, options);
```

  - `itemID` – Internal reference for this tab.
  - `callback` – JavaScript function to run when a tab item is pressed.
  - `title` – The text to display on the tab bar item.
  - `options` – Varies between iOS and Android, see next slide.

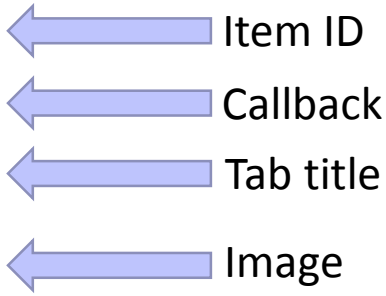
WL.TabBar – continued

- **iOS options**
  - **badge** – string to display on the badge of the item.
  - **image** – file name of an image to use or native iOS button identifier:
    - `tabButton:More`
    - `tabButton:Favorites`
    - `tabButton:Featured`
    - `tabButton:TopRated`
    - `tabButton:Recents`
    - `tabButton:Contacts`
    - `tabButton:History`
    - `tabButton:Bookmarks`
    - `tabButton:Search`
    - `tabButton:Downloads`
    - `tabButton:MostRecent`
    - `tabButton:MostViewed`
- **Android options**
  - **image** – file name of an image to use for an unselected state.
  - **imageSelected** – file name of an image to use for a selected state.

WL.TabBar – continued

- iOS `WL.TabBar.addItem` example:

```
WL.TabBar.addItem("item1",
    function(){ alert("item 1 pressed"); },
    "Item 1",{
        image: "tabButton:Search",
        //image: "images/tabImage.png",}
);
```

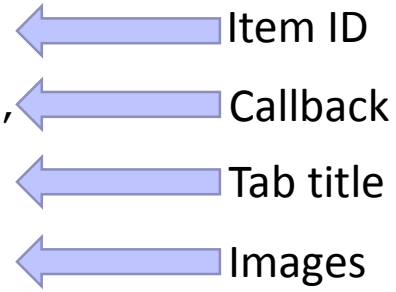


- When you specify an image file, two paths can be used:
  - Using an internal image that is provided by the OS.
  - Specifying a path to the folder in which the image resides.

WL.TabBar – continued

- Android `WL.TabBar.addItem` example:
- When you specify an image file, you need to specify the path to the folder in which the image resides.
- For more information about `WL.TabBar`, see the Worklight user documentation.

```
WL.TabBar.addItem("item1",
    function(){ alert("item 1 pressed"); },
    "Item 1",{
    image: "images/tabImage.png",}
);
```



WL.TabBar – continued

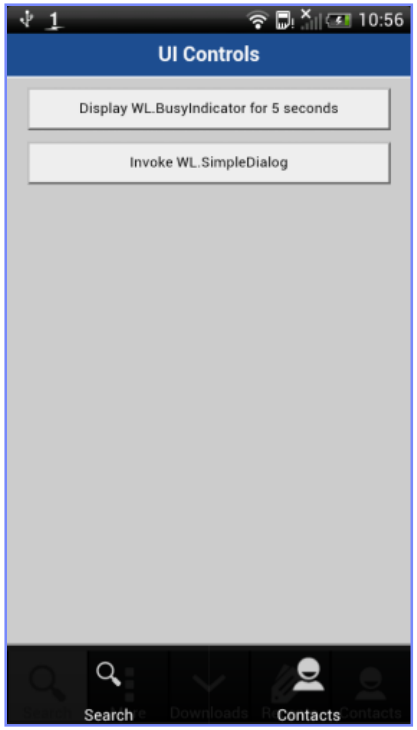
- `WL.TabBar` API
  - `WL.TabBar.init()`
  - `WL.TabBar.addItem` (returns WL.TabBarItem)
  - `WL.TabBar.removeAllItems` (iOS only)
  - `WL.TabBar.setParentDivId` (Android only)
  - `WL.TabBar.setVisible(true/false)`
  - `WL.TabBar.setSelectedItem(itemID)`
  - `WL.TabBar.setEnabled (true/false)`
  - `WL.TabBarItem.setEnabled(true/false)`
  - `WL.TabBarItem.updateBadge(string)` (iOS only)

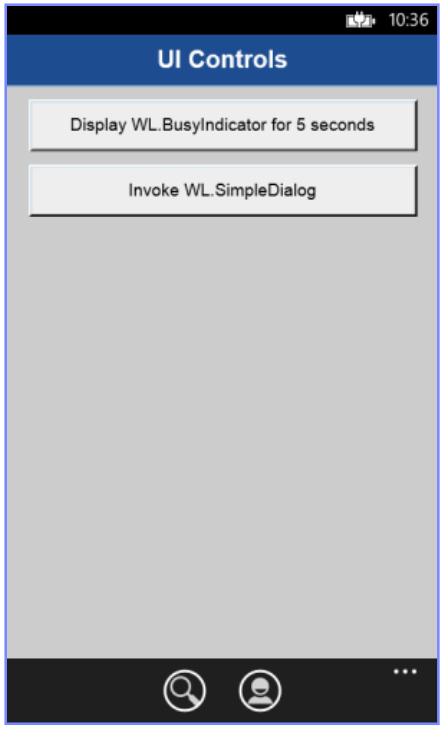
Agenda

- Common controls
  - What is a common control
  - Busy indicator
  - Simple dialog
  - Tab bar
  - Options menu
  - Splash screen
- Sample

WL.OptionsMenu

- Supported environments: Android 2.x, Windows 8, and Windows Phone 8.
- Can display a menu of options.
- In Windows Phone 8 this also functions as a tab bar.
- **Note:** In Android, OptionsMenu is visible up to API Level 15, as well as depending on device OS version. For more information, see the Worklight user documentation.



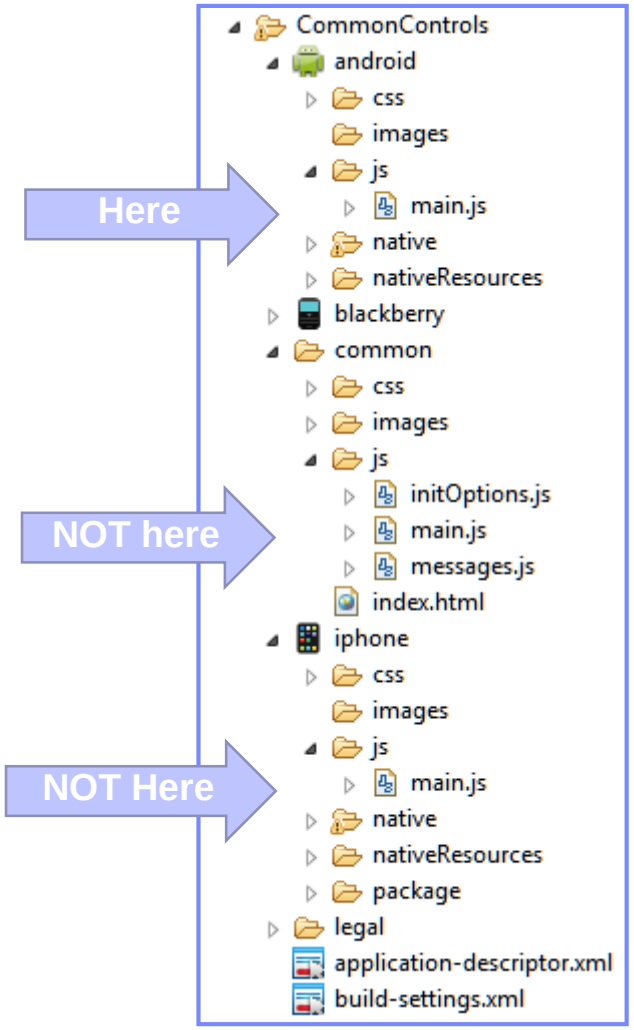


WL.OptionsMenu – continued

- `WL.OptionsMenu` must be initialized before use:

  ```
  WL.OptionsMenu.init();
  ```

- Initialize `WL.OptionsMenu` in the designated JavaScript file.
- The syntax is similar to that of `WL.TabBar` with some minor changes.



WL.OptionsMenu – continued

- Use the following syntax to add an option of a menu:

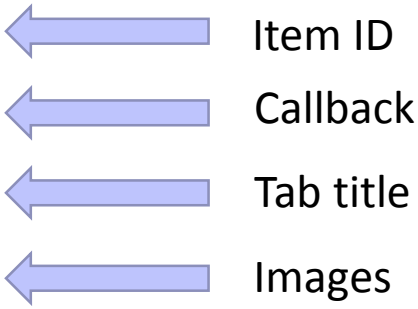
```
WL.OptionsMenu.addItem(id, callbackFunction, title, options);
```

  - `itemID` – Internal reference for this menu option
  - `callback` – JavaScript function to run when the menu option is pressed.
  - `title` – The text of the menu item.
  - `options` – An options object with the following properties:
    - `image` – A path to a designated image, relative to resource root directory.
    - `enabled` – Boolean stating if the item is enabled or disabled.

WL.OptionsMenu – continued

- `WL.OptionsMenu.addItem` example:

```
WL.OptionsMenu.addItem("item2",
    function(){ alert("item 2 pressed");},
    "Contacts", {
    image: "contacts.png"}
);
```



- Paths to image files must not be given; instead, place the files at the following locations:
  - **Android**: `nativeResouces\drawable-*`
  - **Windows 8:** `Resources\applicationBar`
  - **Windows Phone 8:** `nativeResources\applicationBar`
- For more information about `WL.OptionsMenu`, see the Worklight user documentation.

WL.OptionsMenu – continued

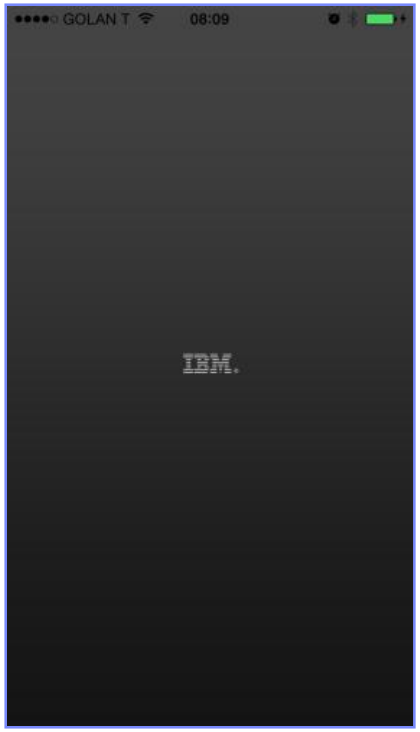
- `WL.OptionsMenu` API
  - `WL.OptionsMenu.init()`
  - `WL.OptionsMenu.addItem` (return reference to a new options item)
  - `WL.OptionsMenu.getItem(itemID)`
  - `WL.OptionsMenu.getItem(itemID).setEnabled (true / false)`
  - `WL.OptionsMenu.setVisible (true / false)`
  - `WL.OptionsMenu.setEnabled (true / false)`
  - `WL.OptionsMenu.removeItem (itemID)`
  - `WL.OptionsMenu.removeItems ()`

Agenda

- Common controls
  - What is a common control
  - Busy indicator
  - Simple dialog
  - Tab bar
  - Options menu
  - Splash screen
- Sample

Splash screen

- Supported environments: Android and iOS.
- Control when to show and hide the splash screen.
- Create a customized experience.



Splash screen – continued

Default behavior

- The Worklight framework provides a default splash screen loading behavior.
  - The splash screen is shown once the application launches
    - This is done in `android\native\src\com\app-name\`*`app-name.java`* and `iphone-or-ipad\native\Classes\`*`app-name.m`*
  - The splash screen is then hidden once the Worklight framework finishes initializing.
- The following slides explain how to extend the default behavior or create a new one altogether.
- For more information, see the Worklight user documentation topic: "**Managing the splash screen with JavaScript APIs**".

Splash screen – continued

Showing and hiding the splash screen

- As mentioned, you can handle when the splash screen to be hidden by using Worklight framework. To do so:
  1. Uncomment the `autoHideSplash` option the `initOptions.js` file.
  2. Use the following API method in at the point in the JavaScript code where you want the splash screen to be hidden. For example:

```
function wlCommonInit() {
    WL.App.hideSplashScreen();
}
```

- If an application requires extra processing time while it launches, the splash screen duration can, for example, be extended by implementing custom JavaScript code that does so.
- Example reasons:
  - Waiting for data from a backend.
  - Loading of more frameworks.
- The next slide shows one possible way for implementing the hiding of the splash screen.

Splash screen – continued

Showing and hiding the splash screen

```
function wlCommonInit(){
    // Custom app logic...
    customLogicCallback();
}

function customLogicCallback() {
    WL.App.hideSplashScreen();
}
```

Splash screen – continued

Showing and hiding the splash screen

- Similarly, the splash screen can be manually displayed again by using the JavaScript API method:

```
WL.App.showSplashScreen();
```

- Example reason:
  - A requirement to force a reload of the application.

```
function wlCommonInit() {
    WL.App.hideSplashscreen();
    // Custom app logic…
    reloadApplication();
}
function reloadApplication() {
    WL.App.showSplashScreen();
    // More custom app code…
    WL.Client.reloadApp();
}
```

Splash screen – continued

Creating a customized experience

- By default, the splash screen that is used in a Worklight application is a static image.
- To use a different image, replace the following with another image:
  - Android: `native\res\drawable\splash.9.png`
  - iOS: `native\Resources\Default-*.png`
- For more information, see Android's 9-Patch documentation as well as the Apple Human Interface Guidelines.

- The splash screen can also be more than a static image.
- A developer can implement custom code that will, for example:
  - Extend the time the splash screen is displayed.
  - Display an animated "loading…" screen.
  - Display a short video clip.
- For more information about creating a custom splash screen, see the Worklight user documentation.

Agenda

- Common controls
  - What is a common control
  - Busy indicator
  - Simple dialog
  - Tab bar
  - Options menu
  - Splash screen
- Sample

Sample

- The sample for this training module can be found in the Getting Started page of the Worklight documentation website at http://www.ibm.com/mobile-docs.

Quiz

Test your knowledge. Answers are in the following slide.

- What is a common control?
  - A common control is a user interface element that is common to most environments.
  - A common control is a way to specify how groups of elements are controlled.
  - A common control is an emulation of a device's native controls by using various technologies.
  - A common control is a native interface element that cannot be controlled by a web means such as JavaScript.
- Which mobile environments use native implementation of a busy indicator?
  - iOS, Android, BlackBerry.
  - iOS, Android, Windows Phone 8.
  - Windows Phone 8, BlackBerry, iOS.
  - iOS, Android, BlackBerry, Windows Phone.
- Which mobile environments support TabBar control?
  - Android, BlackBerry, Windows Phone 8.
  - Android, iOS, Windows Phone 8.
  - iOS, Android, BlackBerry.
  - iOS, Android.

Quiz - answers

- What is a common control?
  - A common control is a user interface element that is common to most environments.
  - A common control is a way to specify how groups of elements are controlled.
  - A common control is an emulation of a device's native controls by using various technologies.
  - A common control is a native interface element that cannot be controlled by a web means such as JavaScript.
- Which mobile environments use native implementation of a busy indicator?
  - iOS, Android, BlackBerry.
  - iOS, Android, Windows Phone 8.
  - Windows Phone 8, BlackBerry, iOS.
  - iOS, Android, BlackBerry, Windows Phone.
- Which mobile environments support TabBar control?
  - Android, BlackBerry, Windows Phone 8.
  - Android, iOS, Windows Phone 8.
  - iOS, Android, BlackBerry.
  - iOS, Android.

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/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