

# ***IBM Worklight Foundation V6.2.0*** **入門**

## 共通 UI コントロール



## 商標

- IBM、IBM ロゴ、ibm.com および Worklight は、世界の多くの国で登録された International Business Machines Corporation の商標です。Worklight は IBM 社の Worklight の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。
- Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。
- Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。
- この資料は、事前に IBM の書面による許可を得ずにその一部または全部を複製することは禁じられています。

## IBM® について

- <http://www.ibm.com/ibm/us/en/> を参照してください。

# アジェンダ

- 共通コントロール
  - 共通コントロールとは
  - ビジー・インディケータ
  - 単純なダイアログ
  - タブ・バー
  - オプション・メニュー
  - スプラッシュ画面
- サンプル

## 共通コントロールとは

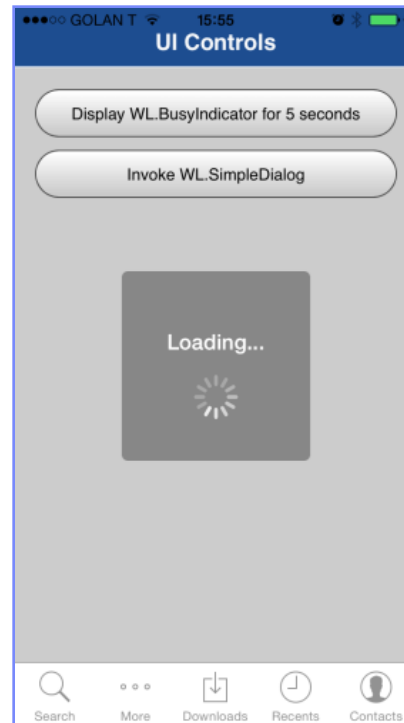
- モーダル・ポップアップ・ウィンドウ、ロード画面、タブ・バーなど、いくつかのコントロールは、ほとんどの環境に共通のものです。
- IBM Worklight® Foundation では、JavaScript™ API を使用して、環境に関係なくこれらのコントロールを呼び出すことができます。この API は、各モバイル・プラットフォームのネイティブな方法で自動的にこれらのコントロールを表示します。

# アジェンダ

- 共通コントロール
  - 共通コントロールとは
  - ビジー・インディケーター
  - 単純なダイアログ
  - タブ・バー
  - オプション・メニュー
  - スプラッシュ画面
- サンプル

# WL.BusyIndicator

- WL.BusyIndicator はモーダル・アクティビティ・インディケーターを表示するための共通 API を実装します。
- これは、Android、iOS、Windows Phone 8、BlackBerry 10、および Windows Phone 8 の各プラットフォームでネイティブ実装を使用します。



## WL.BusyIndicator – 続き

- これは使用前に初期化する必要があります。

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

親エレメント ID (Web のみ)



オプション



- WL.BusyIndicator の親エレメント ID は、iOS 環境、Android 環境、Windows Phone 環境、および BlackBerry 環境では無視されます。
- 使用可能なオプションは以下のとおりです。
  - *text* – モーダル・テキストを設定します。
  - *color* – テキストの色を設定します。
  - *fullScreen* – モーダル・メッセージをフルスクリーンで表示します (iOS のみ)。
  - オプションについては詳しくは、Worklight ユーザー文書を参照してください。

## WL.BusyIndicator – 続き

- WL.BusyIndicator は以下の API を提供します。

- 初期化。

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

- void myBusyIndicator.show() – ビジー・インディケーターを表示する。
- void myBusyIndicator.hide() – ビジー・インディケーターを非表示にする。
- boolean myBusyIndicator.isVisible() – ビジー・インディケーターが表示されているかどうかを返す。

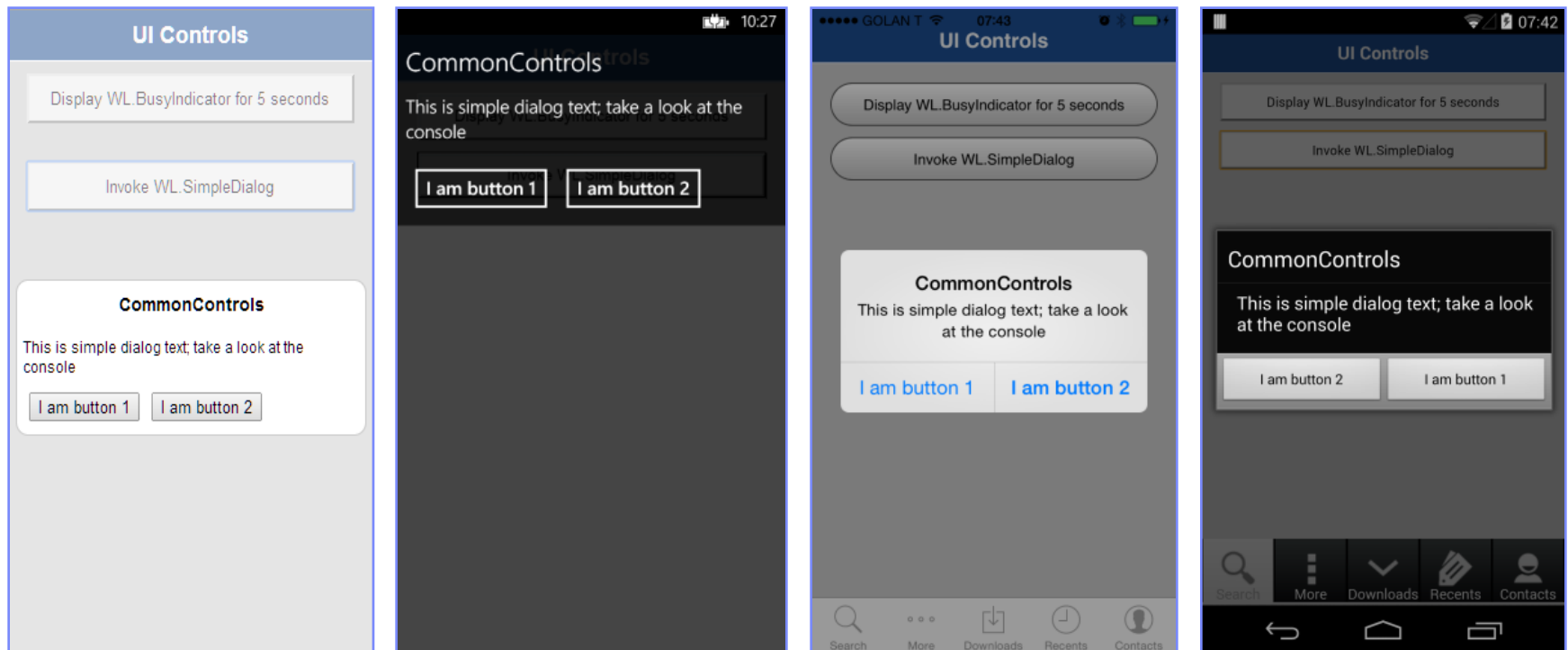


# アジェンダ

- 共通コントロール
  - 共通コントロールとは
  - ビジー・インディケーター
  - 単純なダイアログ
  - タブ・バー
  - オプション・メニュー
  - スプラッシュ画面
- サンプル

# WL.SimpleDialog

- WL.SimpleDialog は、ボタン付きのモーダル・ダイアログ・ウィンドウを表示するための共通 API を実装します。
- Android、iOS、Windows Phone 8、および BlackBerry 10 の各プラットフォームでネイティブ実装を使用します。
- Adobe Air、BlackBerry 6/7、デスクトップ Web ページ、およびモバイル Web では、JS ベースの実装を使用します。



## WL.SimpleDialog – 続き

- 呼び出し構文は次のとおりです。

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

- パラメーターは、title、text、およびボタン・オブジェクトの配列としての buttons です。
- いずれかのボタンを押すとダイアログが閉じます。

## WL.SimpleDialog – 続き

- 各ボタン・オブジェクトには以下の2つのプロパティがあります。
  - *text* – ボタンに表示されるテキスト。
  - *handler* – ボタンが押された場合に呼び出す関数。

```
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 – 続き

以下の制限があります。

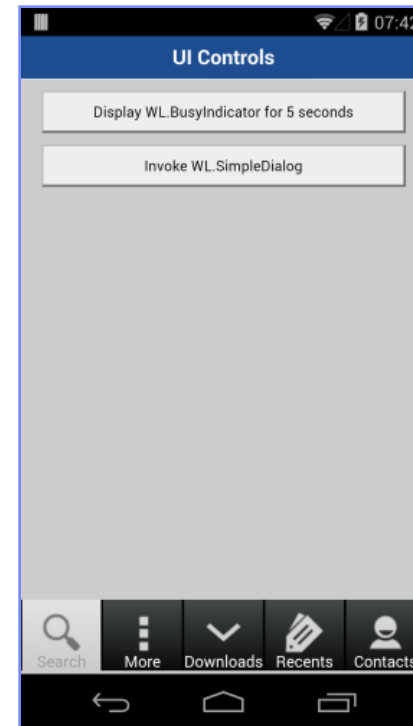
- WL.SimpleDialog を **Windows Phone 8** で使用する場合、WL.SimpleDialog の各インスタンスで使用できるボタンは最大で 4 つです。
- WL.SimpleDialog を **Android** で使用する場合、WL.SimpleDialog の各インスタンスで使用できるボタンは最大で 3 つです。

# アジェンダ

- 共通コントロール
  - 共通コントロールとは
  - ビジー・インディケーター
  - 単純なダイアログ
  - タブ・バー
  - オプション・メニュー
  - スプラッシュ画面
- サンプル

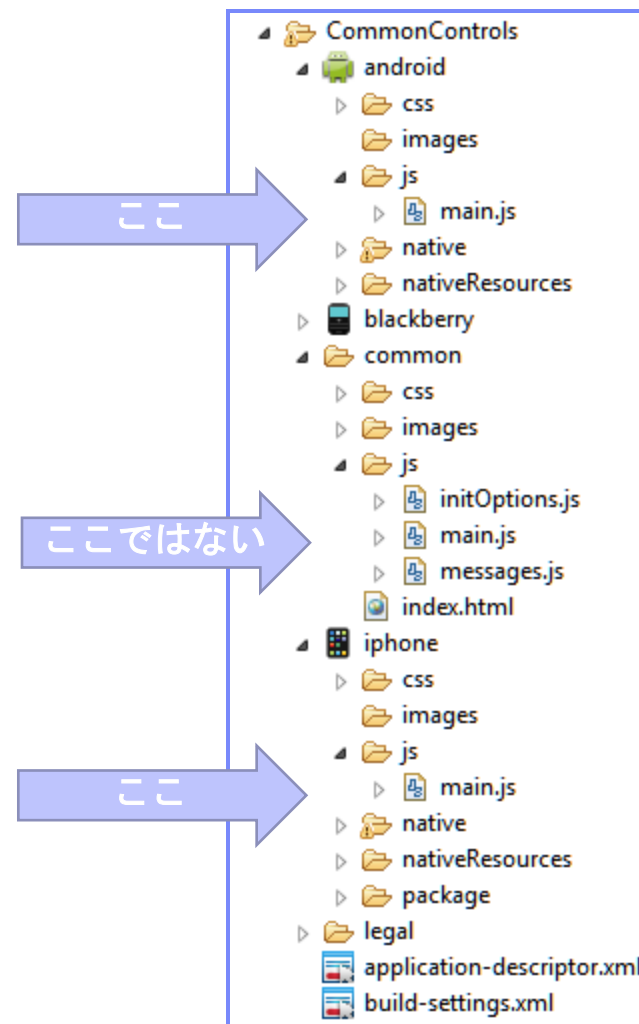
# WL.TabBar

- サポートされる環境: Android および iOS。
- タブ・バー・コンポーネントでのアプリケーション・ナビゲーション。



## WL.TabBar – 続き

- iOS 実装ではネイティブ・コンポーネントを使用しますが、Android は HTML 生成のタブ・バーを使用します。
- 構文は似ていますが、小さな相違がいくつかあります。
- WL.TabBar は、使用前に初期化する必要があります。  
`WL.TabBar.init();`
- 指定の JavaScript ファイルで WL.TabBar を初期化します。





## WL.TabBar – 続き

- タブ・バー項目を追加するには、次の構文を使用します。

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

- `itemID` – このタブの内部参照。
- `callback` – タブ項目が押されたときに実行する JavaScript 関数。
- `title` – タブ・バー項目に表示するテキスト。
- `options` – iOS と Android の間で異なります。次のスライドを参照してください。

## WL.TabBar – 続き

### ■ iOS の options

- **badge** – 項目のバッジに表示するストリング
- **image** – 使用するイメージのファイル名またはネイティブ iOS ボタン ID:
  - `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 の options

- **image** – 選択解除状態に使用するイメージのファイル名。
- **imageSelected** – 選択済み状態に使用するイメージのファイル名。

## WL.TabBar – 続き

- iOS WL.TabBar.addItem の例:

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

← 項目 ID  
← コールバック  
← タブ・タイトル  
← イメージ

- イメージ・ファイルを指定する際には、以下の2つのパスを使用できます。
  - OSによって提供される内部イメージを使用。
  - イメージがあるフォルダーのパスを指定。

## WL.TabBar – 続き

- Android `WL.TabBar.addItem` の例:
- イメージ・ファイルの指定時には、イメージがあるフォルダーのパスを指定する必要があります。
- `WL.TabBar` について詳しくは、Worklight ユーザー文書を参照してください。

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

← 項目 ID  
← コールバック  
← タブ・タイトル  
← イメージ

## WL.TabBar – 続き

### ■ WL.TabBar API

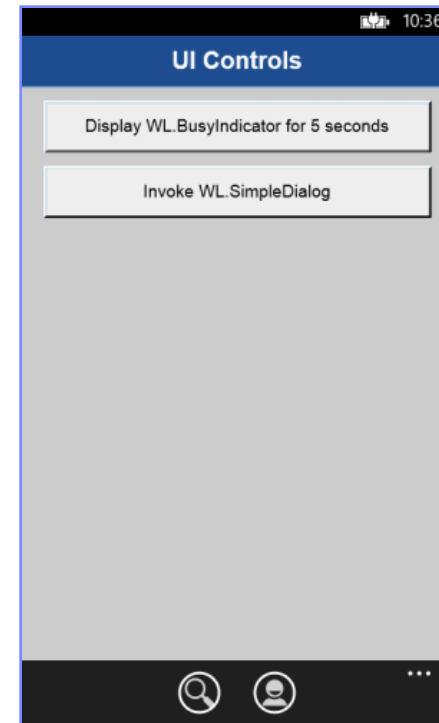
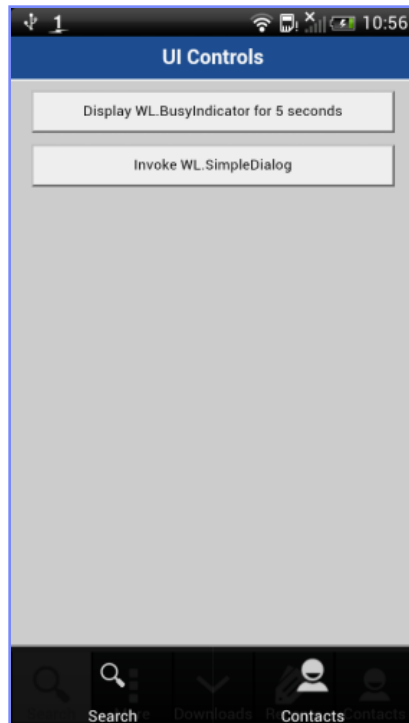
- `WL.TabBar.init()`
- `WL.TabBar.addItem` (`WL.TabBarItem` を返す)
- `WL.TabBar.removeAllItems` (iOS のみ)
- `WL.TabBar.setParentDivId` (Android のみ)
- `WL.TabBar.setVisible(true/false)`
- `WL.TabBar.setSelectedItem(itemId)`
- `WL.TabBar.setEnabled(true/false)`
- `WL.TabBarItem.setEnabled(true/false)`
- `WL.TabBarItem.updateBadge(string)` (iOS のみ)

# アジェンダ

- 共通コントロール
  - 共通コントロールとは
  - ビジー・インディケータ
  - 単純なダイアログ
  - タブ・バー
  - オプション・メニュー
  - スプラッシュ画面
- サンプル

## WL.OptionsMenu

- サポートされる環境: Android 2.x、Windows 8、および Windows Phone 8。
- オプションのメニューを表示できます。
- Windows Phone 8 では、これはタブ・バーとしても機能します。
- 注: Android では、OptionsMenu は最大で API レベル 15 まで表示され、また、デバイス OS バージョンによって異なります、詳しくは、Worklight ユーザー文書を参照してください。

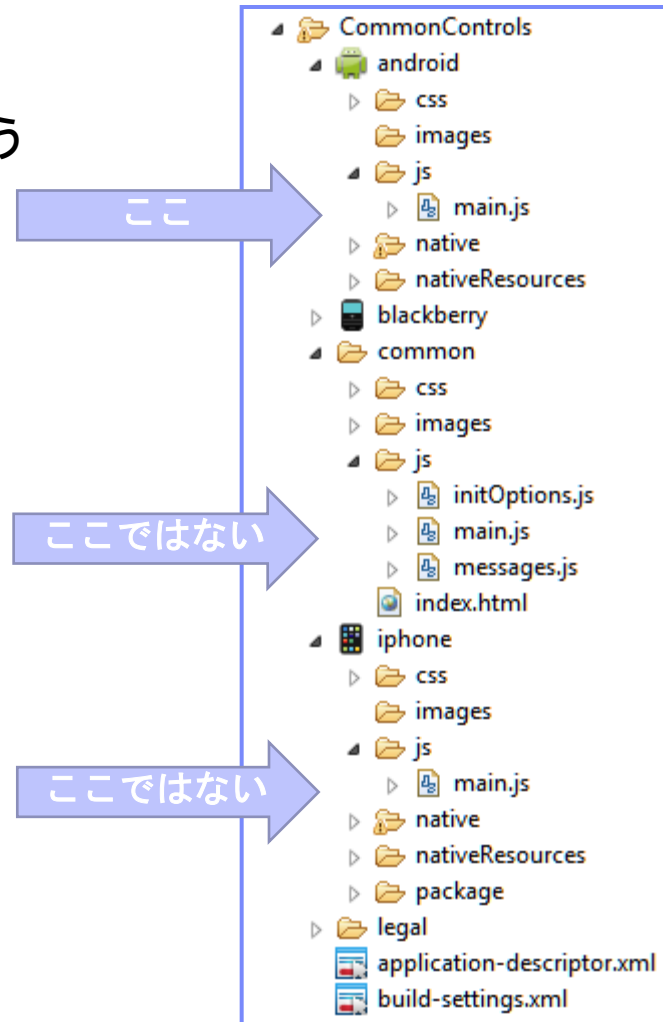


## WL.OptionsMenu – 続き

- WL.OptionsMenu は使用前に以下のように初期化する必要があります。

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

- 指定の JavaScript ファイルで WL.OptionsMenu を初期化します。
- 構文は WL.TabBar の構文と似ていますが、小さな変更がいくつかあります。





## WL.OptionsMenu – 続き

- メニューのオプションを追加するには、次の構文を使用します。

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

- `itemID` – このメニュー・オプションの内部参照
- `callback` – メニュー・オプションが押されたときに実行する JavaScript 関数。
- `title` – メニュー項目のテキスト。
- `options` – 以下のプロパティを持つオプション・オブジェクト:
  - `image` – リソースのルート・ディレクトリーに対して相対的な、指定されたイメージのパス
  - `enabled` – 項目が使用可能であるか使用不可であることを示すブール値

## WL.OptionsMenu – 続き

- WL.OptionsMenu.addItem の例:

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

← 項目 ID  
← コールバック  
← タブ・タイトル  
← イメージ

- イメージ・ファイルのパスを指定してはなりません。代わりに、以下の場所にファイルを置きます。
  - **Android:** nativeResources¥drawable-\*
  - **Windows 8:** Resources¥applicationBar
  - **Windows Phone 8:** nativeResources¥applicationBar
- WL.OptionsMenu について詳しくは、Worklight ユーザー文書を参照してください。

## WL.OptionsMenu – 続き

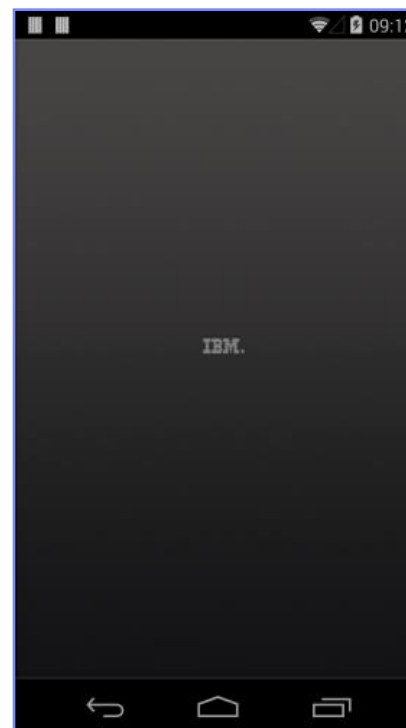
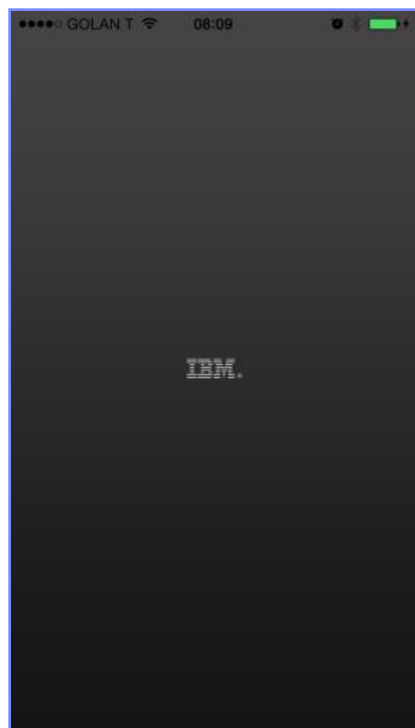
- WL.OptionsMenu API
  - WL.OptionsMenu.init()
  - WL.OptionsMenu.addItem (新規オプション項目への参照を返す)
  - WL.OptionsMenu.getItem(itemID)
  - WL.OptionsMenu.getItem(itemID).setEnabled (true / false)
  - WL.OptionsMenu.setVisible (true / false)
  - WL.OptionsMenu.setEnabled (true / false)
  - WL.OptionsMenu.removeItem (itemID)
  - WL.OptionsMenu.removeItems ()

# アジェンダ

- 共通コントロール
  - 共通コントロールとは
  - ビジー・インディケーター
  - 単純なダイアログ
  - タブ・バー
  - オプション・メニュー
  - **スプラッシュ画面**
- サンプル

## スプラッシュ画面

- サポートされる環境: Android および iOS。
- スプラッシュ画面の表示および非表示を切り替えるタイミングを制御します。
- カスタマイズしたエクスペリエンスを作成します。



# スプラッシュ画面 – 続き

## デフォルトの動作

- Worklight フレームワークは、デフォルトのスプラッシュ画面ロード動作を備えています。
  - スプラッシュ画面は、アプリケーションが起動すると表示されます。
    - これは、`android¥native¥src¥com¥app-name¥app-name.java` および `iphone-or-ipad¥native¥Classes¥app-name.m` で行われます。
  - 次に、Worklight フレームワークの初期化が完了すると、スプラッシュ画面は非表示になります。
- 以降のスライドで、デフォルトの動作を拡張する方法や、完全に新規に動作を作成する方法について説明します。
- 詳しくは、Worklight ユーザー文書のトピック『**JavaScript API を使用したスプラッシュ画面の管理 (Managing the splash screen with JavaScript APIs)**』を参照してください。

# スプラッシュ画面 – 続き

## スプラッシュ画面の表示および非表示

- 前述のとおり、Worklight フレームワークを使用してスプラッシュ画面を非表示にするタイミングを処理できます。これを行うには、以下のようにします。
  1. `initOptions.js` ファイル内の `autoHideSplash` オプションのコメントを外します。
  2. JavaScript コード内のスプラッシュ画面を非表示にする位置で以下の API メソッドを使用します。以下に例を示します。

```
function wlCommonInit() {  
    WL.App.hideSplashScreen();  
}
```
- 例えば、アプリケーションの起動時に追加の処理時間が必要な場合は、スプラッシュ画面の表示期間を延長するカスタム JavaScript コードを実装することで、スプラッシュ画面の表示期間を延長できます。
- 理由の例:
  - バックエンドからのデータを待機する。
  - 追加フレームワークをロードする。
- 次のスライドでは、スプラッシュ画面の非表示を実装する方法を 1 つ示します。

## スプラッシュ画面 – 続き

### スプラッシュ画面の表示および非表示

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



## スプラッシュ画面 – 続き

### スプラッシュ画面の表示および非表示

- 同様に、JavaScript API メソッドを使用して、スプラッシュ画面を手動で再表示できます。

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

- 理由の例:

- アプリケーションの再ロードを強制する必要がある。

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

# スプラッシュ画面 – 続き

## カスタマイズしたエクスペリエンスの作成

- デフォルトでは、Worklight アプリケーションで使用されるスプラッシュ画面は静的イメージです。
- 別のイメージを使用する場合は、以下を別のイメージに置き換えます。
  - Android: `native¥res¥drawable¥splash.9.png`
  - iOS: `native¥Resources¥Default-*.png`
- 詳しくは、[Android の 9-Patch 資料](#) および [Apple の「Human Interface Guidelines」](#) を参照してください。
  
- スプラッシュ画面には、単なる静的イメージではないイメージも使用できます。
- 例えば、開発者は、以下を行うカスタム・コードを実装できます。
  - スプラッシュ画面が表示される時間を延長する。
  - アニメーション表示の「ロード中...」画面を表示する。
  - 短いビデオ・クリップを表示する。
- カスタム・スプラッシュ画面の作成について詳しくは、Worklight ユーザー文書を参照してください。

# アジェンダ

- 共通コントロール
  - 共通コントロールとは
  - ビジー・インディケーター
  - 単純なダイアログ
  - タブ・バー
  - オプション・メニュー
  - スプラッシュ画面
- サンプル

## サンプル

- このトレーニング・モジュールのサンプルは、Worklight 文書 Web サイト (<http://www.ibm.com/mobile-docs>) の「入門」ページにあります。

# クイズ

確認テストを行います。答えは次のスライドにあります。

- 共通コントロールとは
  - 共通コントロールは、ほとんどの環境に共通のユーザー・インターフェース・エレメントです。
  - 共通コントロールは、エレメントのグループをどのように制御するかを指定する手段です。
  - 共通コントロールは、さまざまなテクノロジーを使用した、デバイスのネイティブ・コントロールのエミュレーションです。
  - 共通コントロールは、JavaScript など、Web の手段で制御できないネイティブ・インターフェース・エレメントです。
- ビジー・インディケータのネイティブ実装を使用するのはどのモバイル環境ですか。
  - iOS、Android、BlackBerry。
  - iOS、Android、Windows Phone 8。
  - Windows Phone 8、BlackBerry、iOS。
  - iOS、Android、BlackBerry、Windows Phone。
- TabBar コントロールをサポートするのはどのモバイル環境ですか。
  - Android、BlackBerry、Windows Phone 8。
  - Android、iOS、Windows Phone 8。
  - iOS、Android、BlackBerry。
  - iOS、Android。

# クイズ - 答え

- 共通コントロールとは
  - 共通コントロールは、ほとんどの環境に共通のユーザー・インターフェース・エレメントです。
  - 共通コントロールは、エレメントのグループをどのように制御するかを指定する手段です。
  - 共通コントロールは、さまざまなテクノロジーを使用した、デバイスのネイティブ・コントロールのエミュレーションです。
  - 共通コントロールは、JavaScript など、Web の手段で制御できないネイティブ・インターフェース・エレメントです。
- ビジー・インディケータのネイティブ実装を使用するのはどのモバイル環境ですか。
  - iOS、Android、BlackBerry。
  - iOS、Android、Windows Phone 8。
  - Windows Phone 8、BlackBerry、iOS。
  - iOS、Android、BlackBerry、Windows Phone。
- TabBar コントロールをサポートするのはどのモバイル環境ですか。
  - Android、BlackBerry、Windows Phone 8。
  - Android、iOS、Windows Phone 8。
  - iOS、Android、BlackBerry。
  - iOS、Android。

# 特記事項

- これらの資料は、以下のご使用条件に同意していただける場合に限りご使用いただけます。
- 本書は米国 IBM が提供する製品およびサービスについて作成したものです。
- 本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、または サービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。
- IBM は、本書に記載されている内容に関して特許権（特許出願中のものを含む）を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。
  - 〒103-8510  
東京都中央区日本橋箱崎町19番21号  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

- この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。
- 本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。
- IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。
- 本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム（本プログラムを含む）との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。
  - IBM Corporation  
Dept F6, Bldg 1  
294 Route 100  
Somers NY 10589-3216  
USA

- 本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。
- 本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。
- IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお問い合わせください。

## 著作権使用許諾:

- 本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめめしたり、保証することはできません。
- それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。
  - © (お客様の会社名) (西暦年) このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. 年を入れる。 All rights reserved.

## プライバシー・ポリシーの考慮事項

- サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品（「ソフトウェア・オファリング」）では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的な事項を確認ください。
- このソフトウェア・オファリングは、展開される構成に応じて、(アプリケーション・サーバーが生成する) セッション情報を収集するセッションごとの Cookie を使用場合があります。これらの Cookie は個人情報を含まず、セッション管理のために要求されるものです。加えて、匿名ユーザーの認識および管理のために持続的な Cookie が無作為に生成される場合があります。これらの Cookie も個人情報を含まず、要求されるものです。
- この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、IBM の『IBM オンラインでのプライバシー・ステートメント』(<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』および『IBM Software Products and Software-as-a-Service Privacy Statement』(<http://www.ibm.com/software/info/product-privacy>) を参照してください。

# サポートおよびコメント

- IBM Worklight の一連の文書、トレーニング資料、および質問をポストできるオンライン・フォーラムはすべて、次の IBM Web サイトからご覧になれます。
  - <http://www.ibm.com/mobile-docs>
- サポート
  - ソフトウェア・サブスクリプション & サポート (ソフトウェア・メンテナンスと呼ばれる場合もあります) は、パスポート・アドバンテージおよびパスポート・アドバンテージ・エクスプレスから購入されたライセンスに含まれています。International Passport Advantage Agreement および IBM International Passport Advantage Express Agreement の追加情報については、次のパスポート・アドバンテージ Web サイトを参照してください。
    - <http://www.ibm.com/software/passportadvantage>
  - ソフトウェア・サブスクリプション & サポートが有効になっている場合、IBM は、インストールおよび使用法 (ハウツー) に関する短期間の FAQ に対するサポートや、コード関連の質問に対するサポートを提供します。詳しくは、次の IBM ソフトウェア・サポート・ハンドブックを参照してください。
    - <http://www.ibm.com/support/handbook>
- ご意見
  - 本資料に関するご意見をお寄せください。本資料の具体的な誤りや欠落、正確性、編成、題材、または完成度に関するご意見をお寄せください。お寄せいただくご意見は、本マニュアルまたは製品の情報、およびその情報の提示方法に関するもののみとしてください。
  - 製品の技術的な質問および情報、および価格については、担当の IBM 営業所、IBM ビジネス・パートナー、または認定リマーケットアーにお問い合わせください。
  - IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。IBM またはいかなる組織も、お客様から提示された問題についてご連絡を差し上げる場合のみ、お客様が提供する個人情報を使用するものとします。
  - どうぞよろしくお願いいたします。
  - 次の IBM Worklight Developer Edition サポート・コミュニティにご意見をお寄せください。
    - <https://www.ibm.com/developerworks/mobile/worklight/connect.html>
  - IBM からの回答を希望される場合は、以下の情報をご連絡ください。
    - 氏名
    - 住所
    - 企業または組織
    - 電話番号
    - Eメール・アドレス



ありがとうございました

