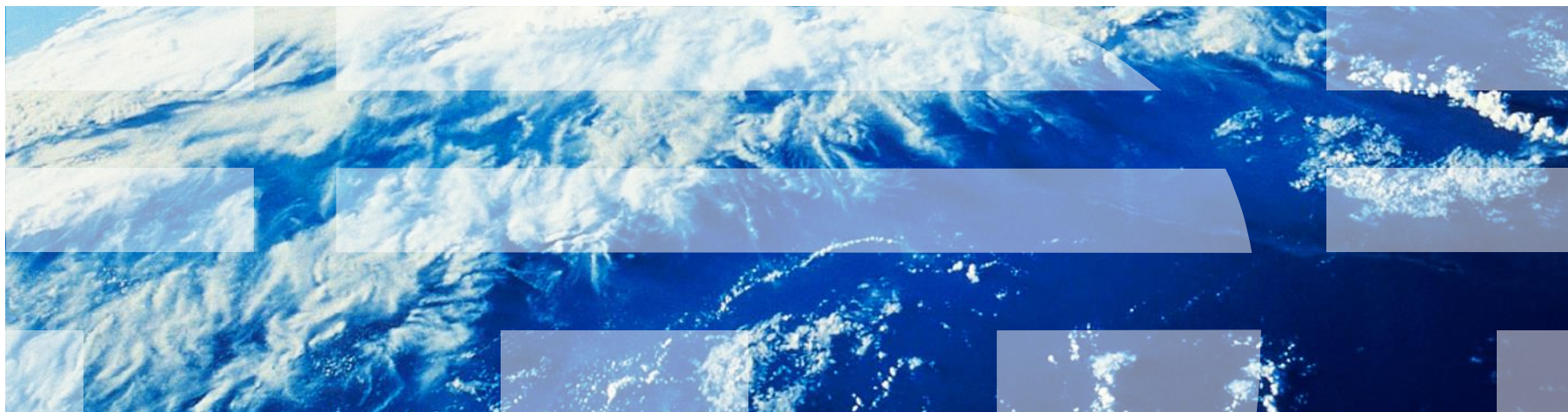


# ***IBM Worklight V6.1.0*** **入門**

ハイブリッド・アプリケーションでサーバー生成ページを統合する



## 商標

- IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com) は、世界の多くの国で登録された International Business Machines Corporation 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。
- 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

- 概要
- Java™ 実装
- JavaScript 実装
- 結果の確認

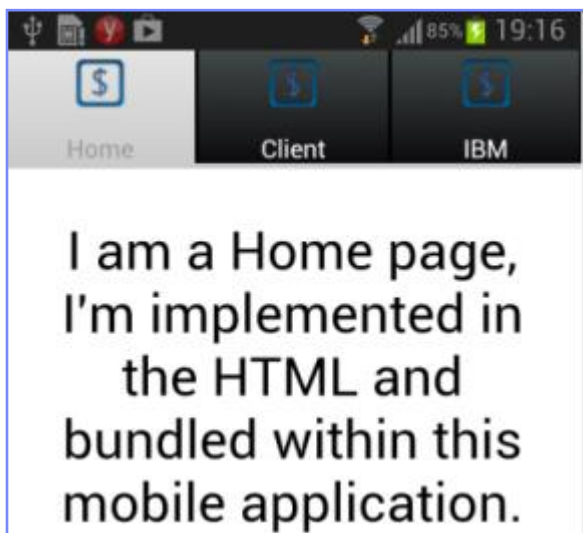
## 概要

- 多くの企業は今日、その独自の顧客対応または従業員対応のモバイル・アプリケーションを開発することを決定しています。
- これらの企業の一部は既に、モバイル対応の Web サイト (モバイル Web) を所有しています。
- 企業は、以下のような重要な決定を行わなければなりません。
  - すべてのモバイル Web 機能をモバイル・アプリケーション内でゼロから実装する必要があるか。これはユーザー・エクスペリエンスの観点からは優れていますが、多大な時間とお金がかかります。
  - モバイル・アプリケーションには新しい機能のみを組み込み、古い機能もモバイル・ブラウザから引き続きアクセスできるようにする必要があるか。この場合、実装は簡単ですが、ユーザー・エクスペリエンスの観点からはあまり優れていません。

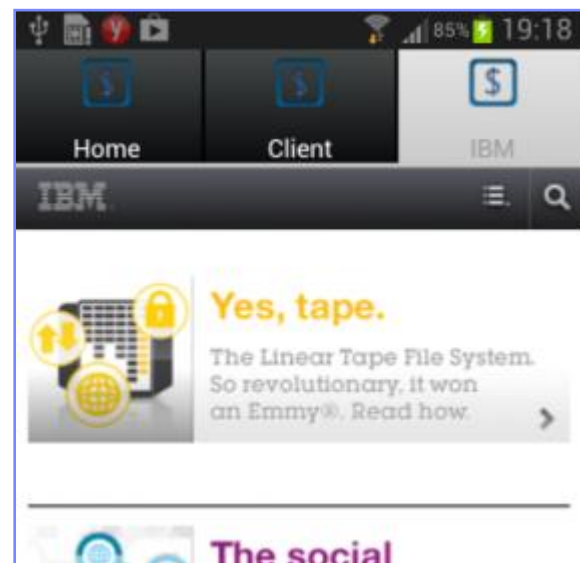
## 概要

- **WebViewOverlay** アプローチにより、新しいモバイル・アプリケーション内で既存のモバイル Web サイトを再利用および統合することが可能になります。
- モバイル・アプリケーション内部のコンポーネントと、外部モバイル Web サイト上の外部コンテンツとの間で、ナビゲーションはスムーズかつシームレスです。

アプリケーション内部に  
バンドルされている Web リソース



外部 Web コンテンツ



## 概要

- このモジュールでは、3つのタブ・バーを持つシンプルなアプリケーションを作成します。
- 最初の2つのタブには内部コンテンツが入ります。3番目のタブには外部 IBM モバイル Web サイトが表示されます。



1 番目と 2 番目のタブ  
には内部 Web リソース  
が入ります



3 番目のタブは別の  
アプリケーション・  
ページのように見え  
ます



ただし技術的には、  
特別な WebView コン  
ポーネントがその上  
位にあります

## 概要

- **WebViewOverlay** は、Apache Cordova プラグイン機能を使用することによって実装されます。
- 開発者は、内部 Web コンポーネントと **WebViewOverlay** コントロールとの間に、独自のプロトコルを容易に作成することができます。
- このモジュールのサンプルを使用して、**WebViewOverlay** の概念を理解することができます。
- 先に進む前に、Cordova プラグインの実装に習熟していることを確認してください。

# アジェンダ

- 概要
- Java 実装
- JavaScript 実装
- 結果の確認



# Java 実装

- 最初に、外部コンテンツを表示するために **webViewOverlay** オブジェクトを宣言します。分かりやすくするために、静的参照を使用します。

```
public class WebViewOverlaySample extends WLDroidGap {  
  
    private static WebView webViewOverlay;  
    public static Activity thisapp;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        super.loadUrl(getWebMainFilePath());  
  
        thisapp = this;  
  
        WebViewClient webViewClient = new WebViewClient() {  
  
            webViewOverlay = new WebView(this);  
            webViewOverlay.setVisibility(View.INVISIBLE);  
            webViewOverlay.setWebViewClient(webViewClient);  
  
            RelativeLayout.LayoutParams webViewOverlayLayoutParams = new RelativeLayout.LayoutParams(  
                RelativeLayout.LayoutParams.FILL_PARENT,  
                RelativeLayout.LayoutParams.FILL_PARENT);  
            webViewOverlayLayoutParams.setMargins(0, 65, 0, 0);  
            webViewOverlay.setLayoutParams(webViewOverlayLayoutParams);  
            webViewOverlay.getSettings().setJavaScriptEnabled(true);  
        }  
    }  
}
```

## Java 実装

- オブジェクトを作成し、そのレイアウト・プロパティを設定し、ルート・エレメントにビューとして追加します。このオブジェクトは初めは表示されていません。

```
public class WebViewOverlaySample extends WLDroidGap {  
  
    private static WebView webViewOverlay;  
    public static Activity thisapp;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        super.loadUrl(getWebMainFilePath());  
  
        thisapp = this;  
  
        WebViewClient webViewClient = new WebViewClient() {  
  
            webViewOverlay = new WebView(this);  
            webViewOverlay.setVisibility(View.INVISIBLE);  
            webViewOverlay.setWebViewClient(webViewClient);  
  
            RelativeLayout.LayoutParams webViewOverlayLayoutParams = new RelativeLayout.LayoutParams(  
                RelativeLayout.LayoutParams.FILL_PARENT,  
                RelativeLayout.LayoutParams.FILL_PARENT);  
            webViewOverlayLayoutParams.setMargins(0, 65, 0, 0);  
            webViewOverlay.setLayoutParams(webViewOverlayLayoutParams);  
            webViewOverlay.getSettings().setJavaScriptEnabled(true);  
        }  
    }  
}
```

## Java 実装

- setMargins API に注目してください。この API は、**webViewOverlay** コンポーネントを画面上で位置付けするために使用できます。

```
public class WebViewOverlaySample extends WLDroidGap {

    private static WebView webViewOverlay;
    public static Activity thisapp;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl(getWebMainFilePath());

        thisapp = this;

        WebViewClient webViewClient = new WebViewClient() {

            webViewOverlay = new WebView(this);
            webViewOverlay.setVisibility(View.INVISIBLE);
            webViewOverlay.setWebViewClient(webViewClient);

            RelativeLayout.LayoutParams webViewOverlayLayoutParams = new RelativeLayout.LayoutParams(
                RelativeLayout.LayoutParams.FILL_PARENT,
                RelativeLayout.LayoutParams.FILL_PARENT);
            webViewOverlayLayoutParams.setMargins(0, 65, 0, 0);
            webViewOverlay.setLayoutParams(webViewOverlayLayoutParams);
            webViewOverlay.getSettings().setJavaScriptEnabled(true);
        };
    }
}
```

## Java 実装

- 注: Android 4.4 では、Chromium に基づいており、WebView の余白に影響する新しいバージョンの WebView が導入されています。この問題について詳しくは、以下の URL を参照してください。

<http://developer.android.com/guide/webapps/migrating.html>

## Java 実装

- 次に、コンテンツ・ビューをセットアップする必要があります。
- ルート・レイアウトとして機能する **RelativeLayout** オブジェクトを作成します。
- **root** インスタンスをその元の親から除去します。
- **root** および **webViewOverlay** を **rootRelativeLayout** に追加します。
- コンテンツ・ビューを **rootRelativeLayout** に設定します。

```
RelativeLayout rootRelativeLayout = new RelativeLayout(this);

((FrameLayout)root.getParent()).removeAllViews();
rootRelativeLayout.addView(root);
rootRelativeLayout.addView(webViewOverlay);
setContentView(rootRelativeLayout);
}
```

# Java 実装

- Cordova プラグインの Java™ コードを実装します。

```
public class WebViewOverlayPlugin extends CordovaPlugin {  
    private final String ACTION_OPEN_URL = "open";  
    private final String ACTION_CLOSE_WEBVIEWOVERLAY = "close";  
    @Override  
    public boolean execute(String action, JSONArray args, CallbackContext callbackContext) {  
        if (action.equals(ACTION_OPEN_URL)) {  
            WebViewOverlaySample.thisapp.runOnUiThread(new Runnable() {  
                public void run() {  
                    WebViewOverlaySample.clearWebViewOverlayHistory();  
                    WebViewOverlaySample.loadWebViewOverlay("http://m.ibm.com/");  
                    WebViewOverlaySample.setWebViewOverlayVisibility(View.VISIBLE);  
                    WebViewOverlaySample.requestWebViewOverlayFocus();  
                    WebViewOverlaySample.clearWebViewOverlayHistory();  
                }  
            });  
        } else if (action.equals(ACTION_CLOSE_WEBVIEWOVERLAY)) {  
            thisapp.runOnUiThread(new Runnable() {  
                {  
                    ySample.loadWebViewOverlayContent("");  
                    ySample.setWebViewOverlayVisibility(View.INVISIBLE);  
                    ySample.clearWebViewOverlayHistory();  
                }  
            });  
        }  
    }  
}
```

プラグインがサポートする  
アクションを宣言します。

# Java 実装

- Cordova プラグインの Java コードを実装します。

```
public class WebViewOverlayPlugin extends CordovaPlugin {  
  
    private final String ACTION_OPEN_URL = "open";  
    private final String ACTION_CLOSE_WEBVIEWOVERLAY = "close";  
  
    @Override  
    public boolean execute(String action, JSONArray args, CallbackContext callbackContext) {  
  
        if (action.equals(ACTION_OPEN_URL)) {  
            WebViewOverlaySample.thisapp.runOnUiThread(new Runnable() {  
                public void run() {  
                    WebViewOverlaySample.clearWebViewOverlayHistory();  
                    WebViewOverlaySample.loadWebViewOverlay("http://m.ibm.com/");  
                    WebViewOverlaySample.setWebViewOverlayVisibility(View.VISIBLE);  
                    WebViewOverlaySample.requestWebViewOverlayFocus();  
                    WebViewOverlaySample.clearWebViewOverlayHistory();  
                }  
            });  
            return true;  
        } else if (action.equals(ACTION_CLOSE_WEBVIEWOVERLAY)) {  
            app.runOnUiThread(new Runnable() {  
                public void run() {  
                    le.loadWebViewOverlayContent("");  
                    le.setWebViewOverlayVisibility(View.INVISIBLE);  
                    le.clearWebViewOverlayHistory();  
                }  
            });  
            return true;  
        }  
    }  
}
```

アプリケーションの Web パーツから「open」要求を受け取った場合は、外部コンテンツをロードし、webViewOverlay を可視にします。

# Java 実装

- Cordova プラグインの Java コードを実装します。

アプリケーションの Web パーツから「close」要求を受け取った後、webViewOverlay をクリアし、非表示にします。

```
public class WebViewOverlayPlugin extends CordovaPlugin {  
    private static final String ACTION_OPEN_WEBVIEWOVERLAY = "open";  
    private static final String ACTION_CLOSE_WEBVIEWOVERLAY = "close";  
  
    public boolean execute(String action, JSONArray args, CallbackContext callbackContext) {  
        if (action.equals(ACTION_OPEN_WEBVIEWOVERLAY)) {  
            callbackContext.success();  
        } else if (action.equals(ACTION_CLOSE_WEBVIEWOVERLAY)) {  
            WebViewOverlaySample.thisapp.runOnUiThread(new Runnable() {  
                public void run() {  
                    webViewOverlay.clearWebViewOverlayHistory();  
                    webViewOverlay.loadWebViewOverlay("http://m.ibm.com/");  
                    webViewOverlay.setVisibility(View.VISIBLE);  
                    webViewOverlay.requestFocus();  
                    webViewOverlay.clearWebViewOverlayHistory();  
                }  
            });  
            return true;  
        } else if (action.equals(ACTION_CLOSE_WEBVIEWOVERLAY)) {  
            WebViewOverlaySample.thisapp.runOnUiThread(new Runnable() {  
                public void run() {  
                    webViewOverlay.loadWebViewOverlayContent("");  
                    webViewOverlay.setVisibility(View.INVISIBLE);  
                    webViewOverlay.clearWebViewOverlayHistory();  
                }  
            });  
            return true;  
        } else {  
            return false;  
        }  
    }  
}
```



# Java 実装

- Cordova プラグインの Java コードを実装します。

```
public class WebViewOverlayPlugin extends CordovaPlugin {  
  
    private final String ACTION_OPEN_URL = "http://";  
    private final String  
  
    @Override  
    public boolean execute(String action, String url, JSONArray args, CallbackContext callbackContext) {  
  
        if (action.equals(ACTION_OPEN_URL)) {  
            WebViewOverlaySample.this.app.runOnUiThread(new Runnable() {  
                public void run() {  
                    WebViewOverlaySample.loadWebViewOverlayContent(url);  
                    WebViewOverlaySample.setVisibility(View.VISIBLE);  
                }  
            });  
            return true;  
        } else if (action.equals(ACTION_CLOSE_WEBVIEWOVERLAY)) {  
            WebViewOverlaySample.this.app.runOnUiThread(new Runnable() {  
                public void run() {  
                    WebViewOverlaySample.loadWebViewOverlayContent("");  
                    WebViewOverlaySample.setVisibility(View.INVISIBLE);  
                    WebViewOverlaySample.clearWebViewOverlayHistory();  
                }  
            });  
            return true;  
        } else  
            return false;  
    }  
}
```

UI 関連のアクションは専用の UI スレッドで実行されることに注目してください。

# アジェンダ

- 概要
- Java 実装
- JavaScript 実装
- 結果の確認

## JavaScript 実装

- JavaScript で、**WebViewOverlayPlugin** オブジェクトを作成し、必要なメソッドを設定します。

```
function WebViewOverlayPlugin() {};  
  
WebViewOverlayPlugin.prototype.open = function(successCallback, errorCallback) {  
    cordova.exec(successCallback, errorCallback, 'WebViewOverlayPlugin', 'open', []);  
};  
  
WebViewOverlayPlugin.prototype.close = function(successCallback, errorCallback) {  
    cordova.exec(successCallback, errorCallback, 'WebViewOverlayPlugin', 'close', []);  
};  
  
WebViewOverlayPlugin.prototype.goBack = function(successCallback, errorCallback) {  
    cordova.exec(successCallback, errorCallback, 'WebViewOverlayPlugin', 'goBack', []);  
};  
  
cordova.addConstructor(function() {  
    if (!window.plugins) window.plugins = {};  
    window.plugins.webViewOverlay = new WebViewOverlayPlugin();  
});
```

## JavaScript 実装

- クリックされたタブ ID を分析し、それに従って webViewOverlay を表示または非表示にします。

外部コンテンツのロードには時間がかかる場合があるため、busyIndicator を追加してユーザー・エクスペリエンスを向上させることを検討してください。

```
function tabClicked(id){
    WL.Logger.debug("tabClicked >> id :: " + id);
    $(".tab").addClass('hidden');

    if (id=="3"){
        openOverlayWebView();
    } else {
        $("#tab" + id).removeClass('hidden');
        closeOverlayWebView();
    }
}

function openOverlayWebView(){
    WL.Logger.debug("openOverlayWebView");
    busyIndicator.show();
    setTimeout(busyIndicator.hide, 5000);
    window.plugins.webViewOverlay.open();
}

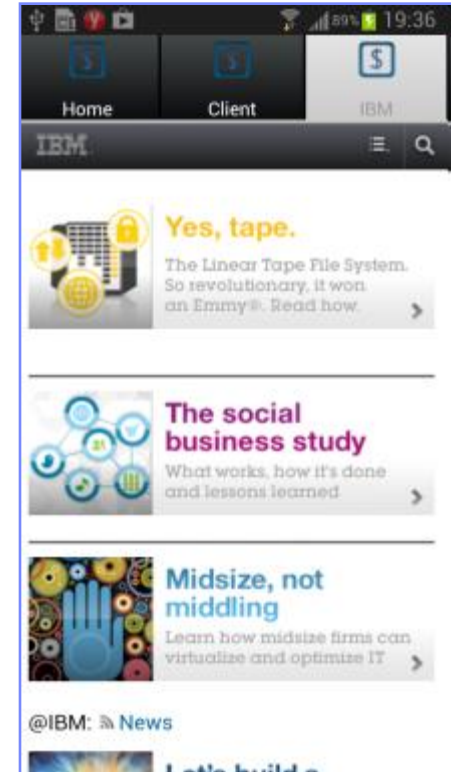
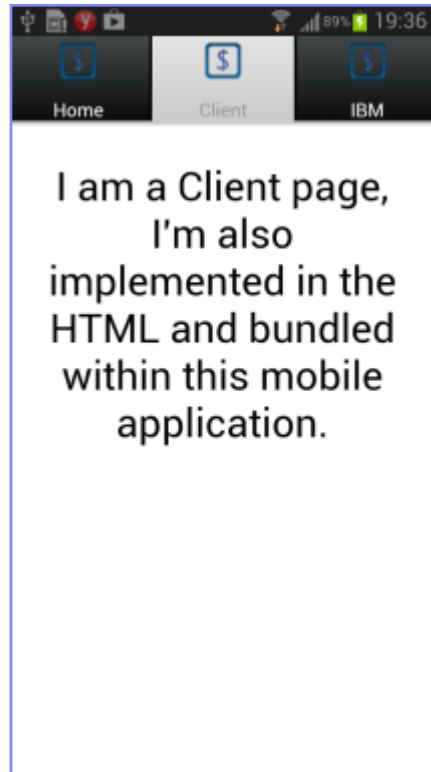
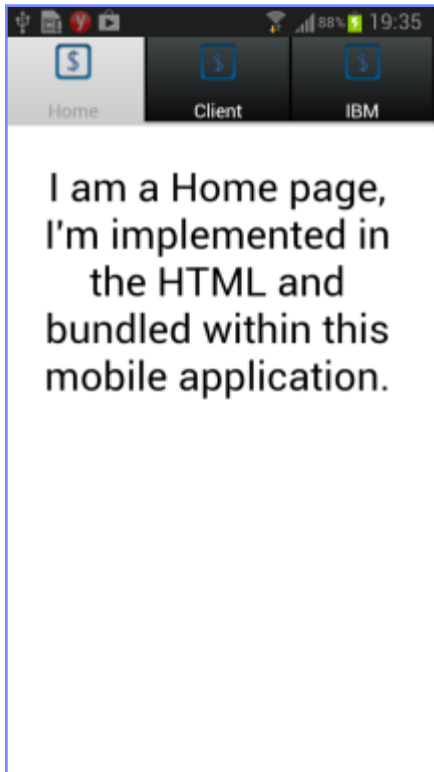
function closeOverlayWebView(){
    WL.Logger.debug("closeOverlayWebView");
    window.plugins.webViewOverlay.close();
}
```

# アジェンダ

- 概要
- Java 実装
- JavaScript 実装
- 結果の確認

## 結果の確認

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



# 特記事項

- これらの資料は、以下のご使用条件に同意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のめかしたり、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 を使用する場合があります。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、または認定リマーカータ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にのみ、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

