

# ***IBM Worklight V6.1.0*** **入門**

## SiteMinder との統合



## 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

- 背景
- チャレンジ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成 (Android)
- ネイティブ・ページ・ハンドラーの作成 (iOS)
- 結果の確認
- トラブルシューティング
- 演習

## 背景

- SiteMinder では、他のフォーム・ベースの認証方式の場合と同じように、認証が要求されたときにカスタム・チャレンジ・ハンドラーが検出されなければなりません。
- このモジュールでは、SiteMinder 認証フォームがユーザーに直接表示されるため、ネイティブ・ページ・ハンドラーが必要となります。
- このネイティブ・ページ・ハンドラーは、SiteMinder Cookie を IBM® Worklight® アプリケーションに伝搬するために必要です。
- このモジュールでは特に SiteMinder がターゲットになっていますが、ここで説明されている概念は他のフォーム・ベースの認証方式にも当てはまると考えられます。

# アジェンダ

- 背景
- チャレンジ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成 (Android)
- ネイティブ・ページ・ハンドラーの作成 (iOS)
- 結果の確認
- トラブルシューティング
- 演習

## チャレンジ・ハンドラーの作成

- 以下のコードを使用してチャレンジ・ハンドラーの実装を開始します。このコードは、SiteMinder ログイン・ページを検出するために使用されます。

```
var LoginFormChallengeHandler = WL.Client.createChallengeHandler("SampleAppRealm");

LoginFormChallengeHandler.isCustomResponse = function(response) {
  if (!response || response.responseText === null) {
    return false;
  }

  if ((response.status == 200) && (response.responseText.indexOf("login.fcc") !== -1)) {
    return true;
  } else {
    return false;
  }
};
```

## チャレンジ・ハンドラーの作成

- このチャレンジ・ハンドラーでは、以下のような SiteMinder 構成が想定されます。
  - **PreservePostData** が **yes** に設定されている。この値はデフォルトです。
  - 使用されるテンプレート・ファイルの名前は **login.fcc** である。
- SiteMinder 構成で **PreservePostData** が **yes** に設定されている場合:
  - ログイン・ページが即時に表示されることはありません。
  - 代わりに、一時ページが表示され、POST パラメーターが収集されてログイン・ページに転送されます。
  - 次に、チャレンジ・ハンドラーが、この一時ページで特定のテキストを検索します。
    - 通常、検索対象のストリングは FCC テンプレート・ファイルの名前です。今回の例では **login.fcc** です。

## チャレンジ・ハンドラーの作成

- 以下のコードを使用してチャレンジ・ハンドラーの実装を続行します。このコードは、適切なネイティブ・ページ・ハンドラーを開始するために使用されます。

```
LoginFormChallengeHandler.handleChallenge = function(response) {  
    var nativeClass = "LoginFormNativePage";  
    if (WL.Client.getEnvironment() == WL.Environment.ANDROID)  
        nativeClass = "com.TestSiteminderApp." + nativeClass;  
  
    if (WL.NativePage){  
        WL.NativePage.show(nativeClass, LoginFormChallengeHandler.backFromNative, {});  
    } else {  
        alert("Functional on iOS/Android devices only.");  
    }  
};
```

- ネイティブ・ページ・ハンドラーは、リモート認証フォームを処理するブラウザユーザー・インスタンスを開始するネイティブ・コードです。
- このモジュールでは、iOS については **LoginFormNativePage** のクラス名が使用され、Android については **com.TestSiteMinderApp.LoginFormNativePage** のクラス名が使用されます。
  - 実際の値は異なる可能性があります。

## チャレンジ・ハンドラーの作成

- 以下のコードを使用してチャレンジ・ハンドラーの実装を完了します。このコードは、ネイティブ・ログイン・ハンドラーから正常なログインを検出するために使用されます。

```
LoginFormChallengeHandler.backFromNative = function(data){  
    if (data.status == "success") {  
        LoginFormChallengeHandler.submitSuccess();  
    } else {  
        LoginFormChallengeHandler.submitFailure();  
    }  
};
```

# アジェンダ

- 背景
- チャレンジ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成 (Android)
- ネイティブ・ページ・ハンドラーの作成 (iOS)
- 結果の確認
- トラブルシューティング
- 演習

## ネイティブ・ページ・ハンドラーの作成

- ネイティブ・ページ・ハンドラーの基本タスクは、SiteMinder リモート認証フォームをロードすることです。
- このネイティブ・ページ・ハンドラーが作成される原因となったりリモート認証フォームは再利用できません。
- その結果、任意の URL がロードされ、その URL から SiteMinder リモート認証フォームが返されます。
- このモジュールでは、Worklight Server 自体が SiteMinder によって保護されていることを想定します。
- そのため、アクセス先の URL は **/apps/services/reach** になります。
  - この URL は Worklight Server のコンテキスト内にあり、いつでも使用できます。

# アジェンダ

- 背景
- チャレンジ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成 (Android)
- ネイティブ・ページ・ハンドラーの作成 (iOS)
- 結果の確認
- トラブルシューティング
- 演習

## ネイティブ・ページ・ハンドラーの作成 (Android 1/4)

- クラスとその `onCreate()` メソッドを作成して、ネイティブ・ページ・ハンドラーの実装を開始します。

```
public class LoginFormNativePage extends Activity {
    private WebView webView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        webView = new WebView(this);
        webView.setWebViewClient(new MyWebViewClient());
        webView.getSettings().setJavaScriptEnabled(true);
        webView.addJavascriptInterface(new HTMLExtractor(), "HTMLExtractor");

        // load ANY url that will trigger login form
        WLConfig cfg = new WLConfig(this);
        webView.loadUrl(cfg.getRootURL() + "/apps/services/reach");
        setContentView(webView);
    }
}
```

## ネイティブ・ページ・ハンドラーの作成 (Android 2/4)

- 以下のコードを使用してネイティブ・ページ・ハンドラーの実装を完了します。

```
private class MyWebViewClient extends WebViewClient{
    @Override
    public void onPageFinished(WebView view, String url) {
        Log.d("", "onPageFinished :: " + url);
        view.loadUrl("javascript:window.HTMLExtractor.extract(document.body.innerHTML)");
        super.onPageFinished(view, url);
    }
}

private class HTMLExtractor {
    public void extract(String html){
        // In case response does not contain login form - return to hybrid
        if (!html.contains("USER")){
            Intent i = new Intent();
            i.putExtra("status", "success");
            setResult(RESULT_OK, i);
            finish();
        }
    }
}
```

## ネイティブ・ページ・ハンドラーの作成 (Android 3/4)

- ネイティブ・ページ・ハンドラーのページがロードされた後に実行する JavaScript™ インターフェースとして **HTMLExtractor.extract()** メソッドを登録します。
- **HTMLExtractor.extract()** メソッドを調べて、SiteMinder ログイン・フォームを使用することになっていることを確認します。
  - ここでは、カスタマイズされていない多くの SiteMinder ログイン・フォームと同様に、ページにフィールド **USER** が含まれている場合は、SiteMinder ログイン・フォームを使用することを検討します。
- 同様に **HTMLExtractor.extract()** メソッドを調べて、SiteMinder ログイン・フォームが表示される前にネイティブ・ページ・ハンドラーが実行されることがないようにします。

## ネイティブ・ページ・ハンドラーの作成 (Android 4/4)

- 最後に、ネイティブ・ページ・ハンドラーは Android アクティビティの 1 つであるため、ご使用のアプリケーション **AndroidManifest.xml** ファイル内に適切な定義を追加します。

```
<activity android:name=".TestSiteminderApp"
    android:label="@string/app_name"
    android:configChanges="orientation/keyboardHidden"
    android:launchMode="singleTask">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
        <action android:name="com.TestSiteminderApp.TestSiteminderApp.M
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity android:name=".LoginFormNativePage"></activity>
<!-- Preference Activity -->
```

# アジェンダ

- 背景
- チャレンジ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成 (Android)
- ネイティブ・ページ・ハンドラーの作成 (iOS)
- 結果の確認
- 演習

# ネイティブ・ページ・ハンドラーの作成 (iOS 1/3)

- **LoginFormNativePage.m** ファイルとその **viewDidLoad** メソッドを作成して、ネイティブ・ページ・ハンドラーの実装を開始します。

```
#import "LoginFormNativePage.h"
#import "NativePage.h"

@interface LoginFormNativePage ()
@end

@implementation LoginFormNativePage

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
    }
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    NSUserDefaults *prefs = [NSUserDefaults standardUserDefaults];

    // hit the reachability url, which will trigger the login form to load
    NSString *serverURL = [NSString stringWithFormat:@"%s/apps/services/reach",[prefs objectForKey:@"WLDefaultServerURL"]];

    // Creating a full screen UIWebView to display login form received from a backend
    UIWebView *webView = [[UIWebView alloc] initWithFrame:CGRectMake(0, 0, 320, 480)];
    [webView setDelegate:self];
    [self.view addSubview:webView];
    [webView loadRequest:[NSURLRequest requestWithURL:[NSURL URLWithString:serverURL]];
    [webView release];
}
}
```

## ネイティブ・ページ・ハンドラーの作成 (iOS 2/3)

- 以下のコードを使用してネイティブ・ページ・ハンドラーの実装を完了します。

```
- (void)webViewDidFinishLoad:(UIWebView *)webView{
    NSLog(@"webViewDidFinishLoad");

    // Get the received web content
    NSString *webViewContent = [webView stringByEvaluatingJavaScriptFromString:@"document.body.innerHTML"];

    // Check if it contains something that will tell us that this is a login form
    if ([webViewContent rangeOfString:@"USER" options:NSCaseInsensitiveSearch].location != NSNotFound){
        // In case it's a login form - do nothing.
        NSLog(@"Looks like we're still in a login form");
    } else {
        // In case it is not a login form looks like we've passed authentication. Need to get back to web part.
        NSLog(@"This is not a login form. Looks like auth is done");
        NSDictionary *response = [NSDictionary dictionaryWithObject:@"success" forKey:@"status"];
        [NativePage showWebView:response];
    }
}
```

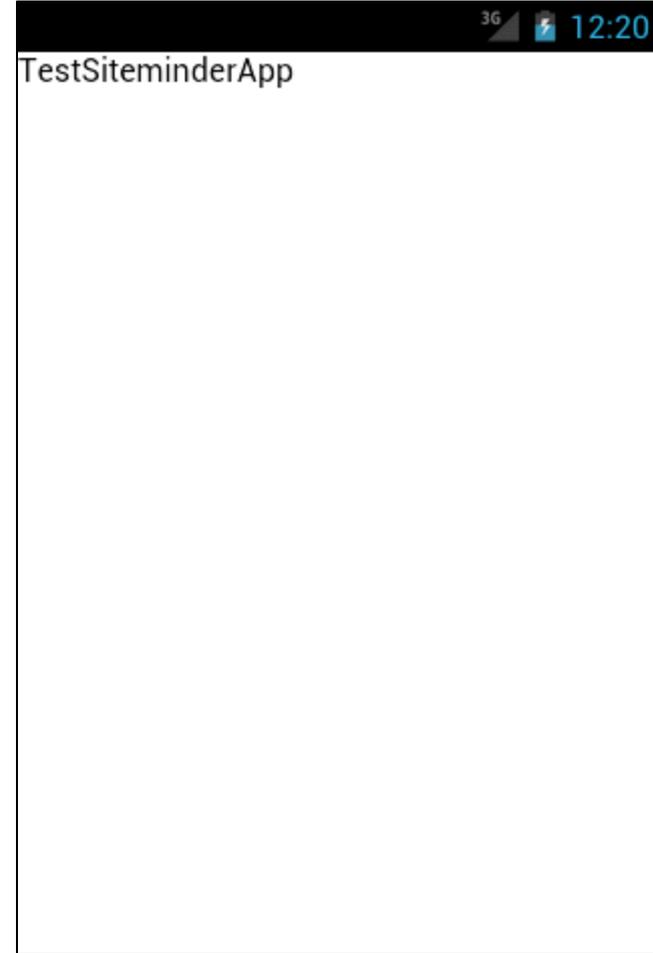
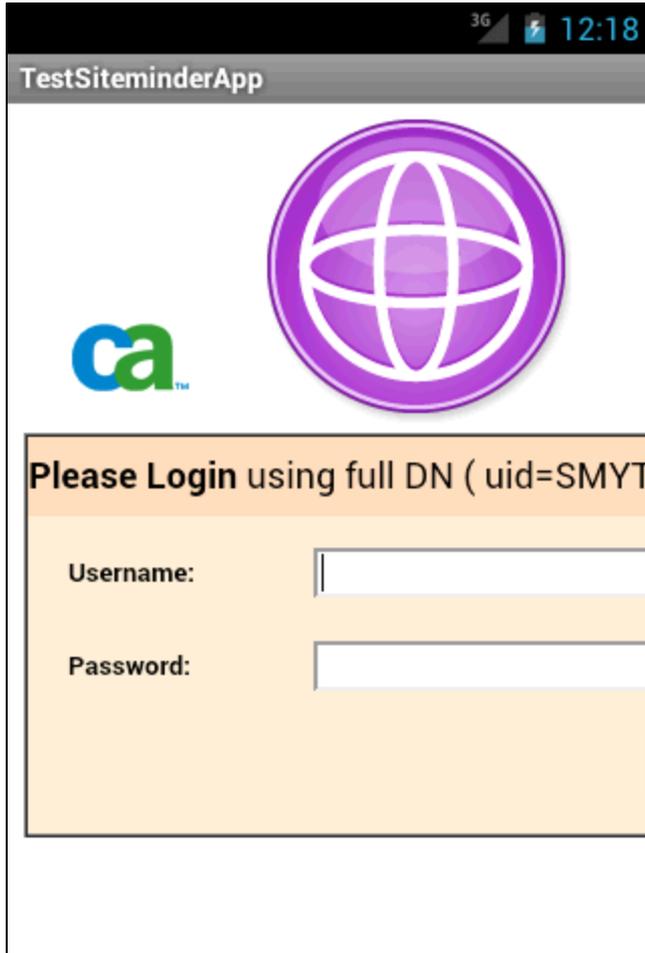
## ネイティブ・ページ・ハンドラーの作成 (iOS 3/3)

- **webViewDidFinishLoad** メソッドを調べて、SiteMinder ログイン・フォームを使用することになっていることを確認します。
  - ここでは、カスタマイズされていない多くの SiteMinder ログイン・フォームと同様に、ページにフィールド **USER** が含まれている場合は、SiteMinder ログイン・フォームを使用することを検討します。
- 同様に **webViewDidFinishLoad** メソッドを調べて、SiteMinder ログイン・フォームが表示される前にネイティブ・ページ・ハンドラーが実行されることがないようにします。

# アジェンダ

- 背景
- チャレンジ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成 (Android)
- ネイティブ・ページ・ハンドラーの作成 (iOS)
- **結果の確認**
- トラブルシューティング
- 演習

# 結果の確認



# アジェンダ

- 背景
- チャレンジ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成 (Android)
- ネイティブ・ページ・ハンドラーの作成 (iOS)
- 結果の確認
- **トラブルシューティング**
- 演習

## トラブルシューティング

- SiteMinder はカスタマイズ可能であるため、このシナリオもカスタマイズできます。
- トラブルシューティングの際は、以下の 2 つの分野に注目します。
  - ログイン・フォームは正しく検出されたか。
    - `/apps/services/reach` の内容を調べて、予期したとおりのログイン・フォームを受け取っていることを確認します。
    - 返されたコンテンツに、チャレンジ・ハンドラーおよびネイティブ・ページ・ハンドラーが探しているストリングが含まれていることを確認します。ブラウザではポスト・データのリダイレクトを識別できないことがあるため、ネットワーク応答を直接調べなければならない場合があります。
  - 認証は正常に完了したか。
    - ネットワーク・トラフィックを調べて、応答で Cookie が設定されていたかどうかを確認します。
    - `submitSuccess` がローカルに呼び出されることを確認するためにロギングを追加します。

# アジェンダ

- 背景
- チャレンジ・ハンドラーの作成
- ネイティブ・ページ・ハンドラーの作成 (Android)
- ネイティブ・ページ・ハンドラーの作成 (iOS)
- 結果の確認
- トラブルシューティング
- 演習

## 演習

- このモジュールで説明されているリモート・フォーム・ベースの認証を実装します。
- 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

