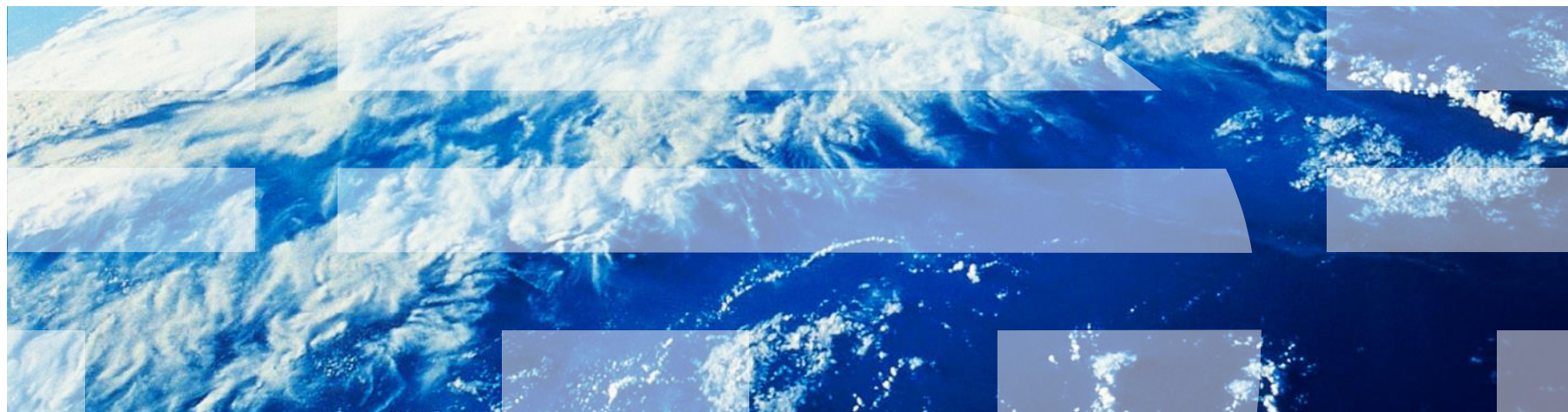


# ***IBM Worklight V6.1.0*** **入門**

## 初めての Worklight アプリケーションの作成



## 商標

- IBM、IBM ロゴおよび [ibm.com](http://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、[www.ibm.com/legal/copytrade.shtml](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

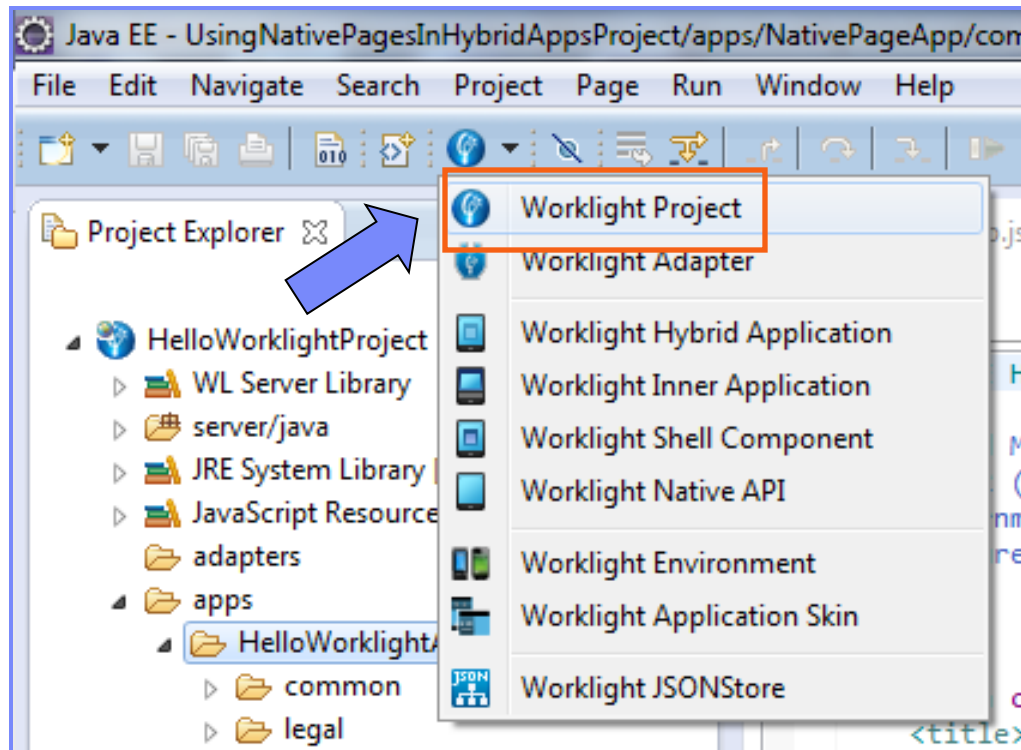
- <http://www.ibm.com/ibm/us/en/> を参照してください。

# アジェンダ

- 初めてのアプリケーションを作成する
- アプリケーション・ファイルと HTML 構造を理解する
- Worklight Studio を使用してアプリケーションをデプロイする
- Worklight Studio に関するヒント
- 演習

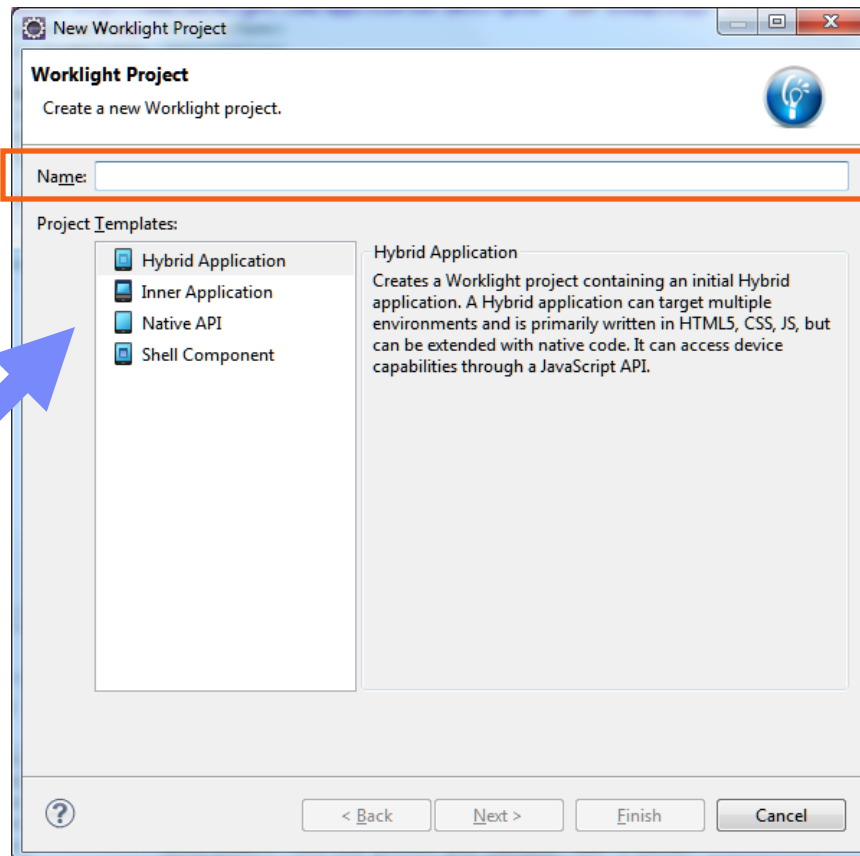
# Hello Worklight Application

- IBM Worklight® プロジェクトを作成します。



# Hello Worklight Application

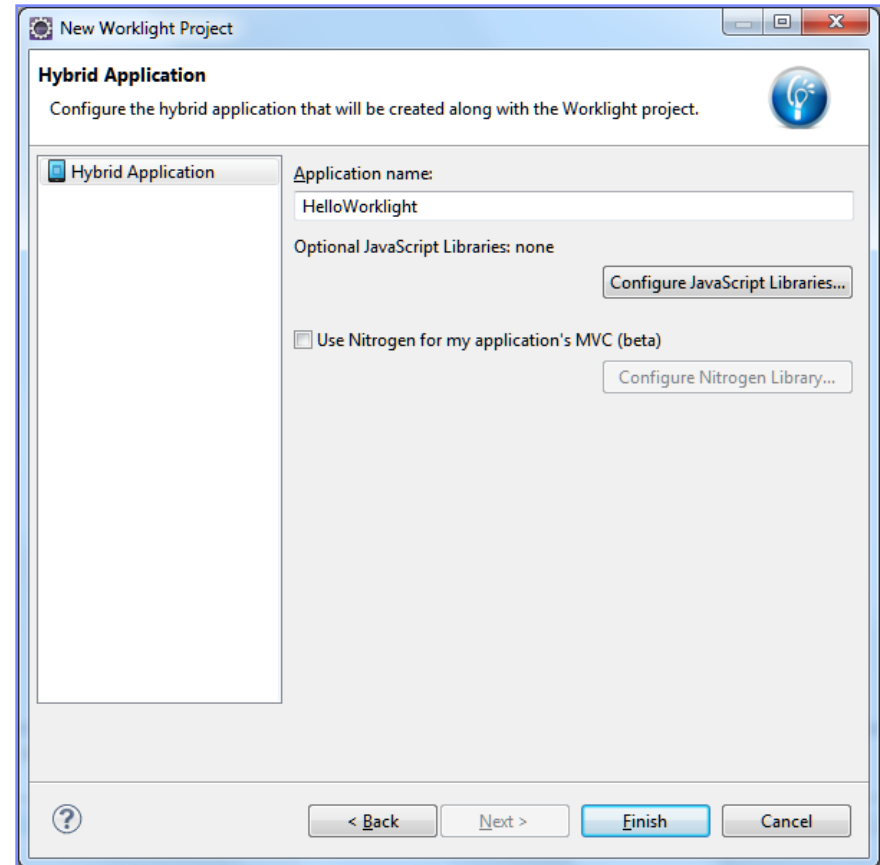
- このプロジェクトに「HelloWorklightProject」という名前を付け、「ハイブリッド・アプリケーション (Hybrid Application)」テンプレートを選択します。



必要なアプリケーション  
・タイプを選択します

# Hello Worklight Application

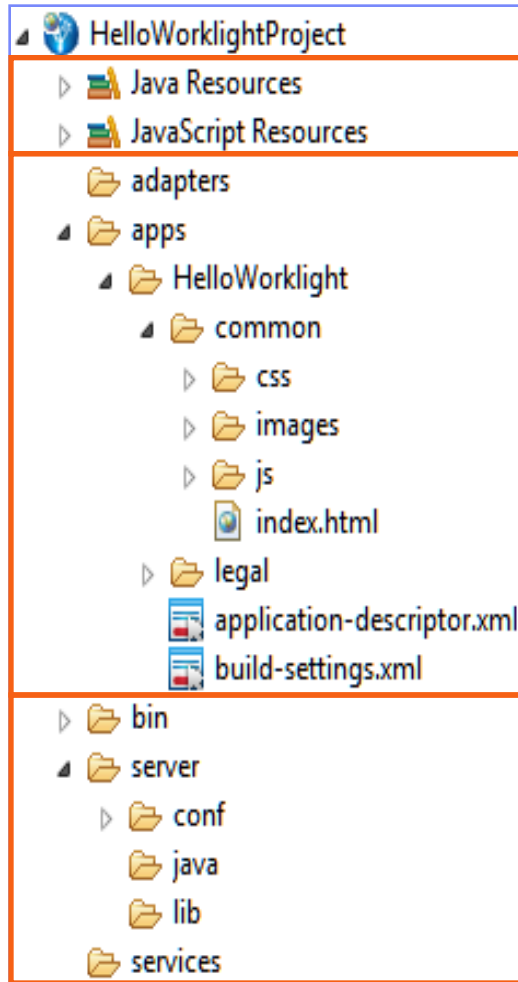
- アプリケーションに「HelloWorklight」という名前を付けます。
- この画面で、JavaScript™ フレームワークをプロジェクトに追加できます。
- アプリケーションへのフレームワークの追加については、「クライアント・サイド開発の基礎 - UI フレームワークの処理」入門モジュールをお読みください。
- 終わったら、「終了 (Finish)」をクリックします。



# アジェンダ

- 初めてのアプリケーションを作成する
- アプリケーション・ファイルと HTML 構造を理解する
- Worklight Studio を使用してアプリケーションをデプロイする
- Worklight Studio に関するヒント
- 演習

# Worklight プロジェクト構造



アプリケーションの開発および  
デプロイメントに必要な参照



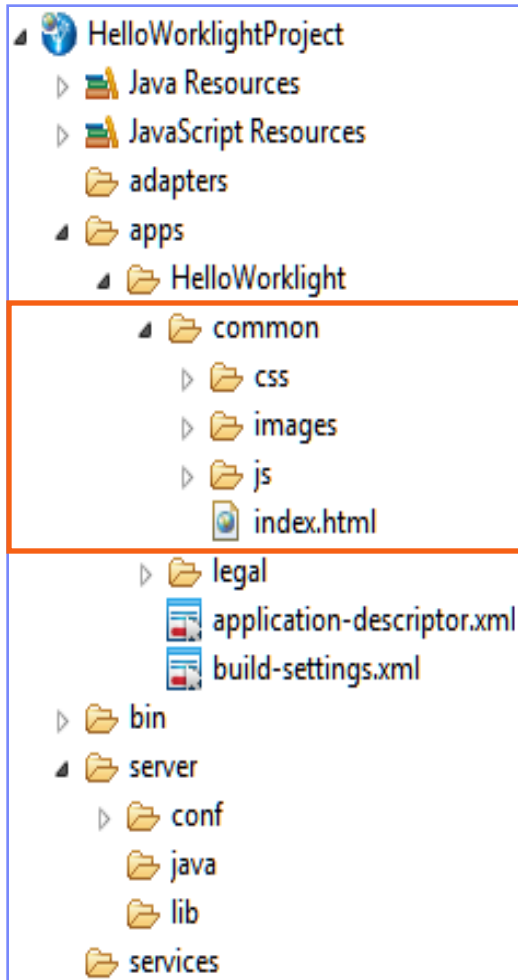
プロジェクトのアプリケーション  
およびアダプター



サーバー・カスタマイズ・コン  
ポーネント

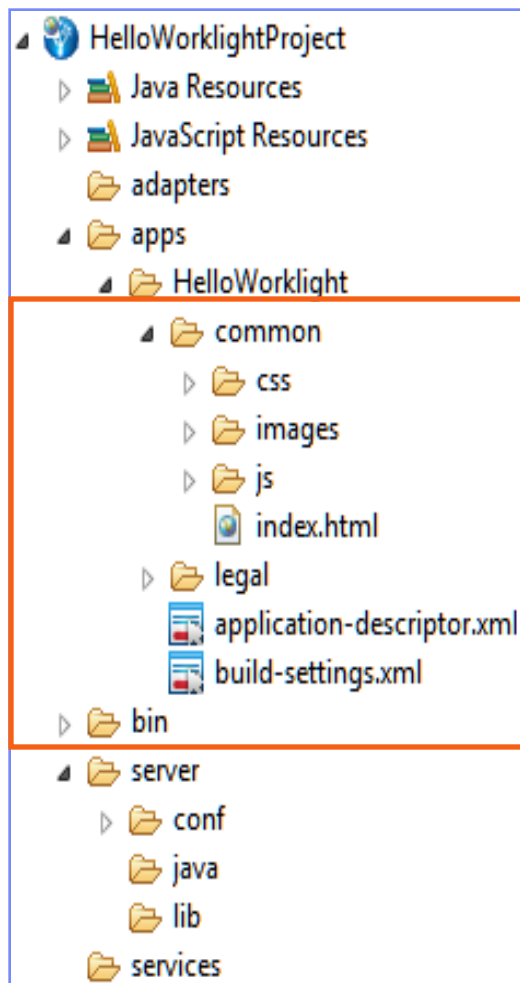


# HelloWorklight – プロジェクト構造



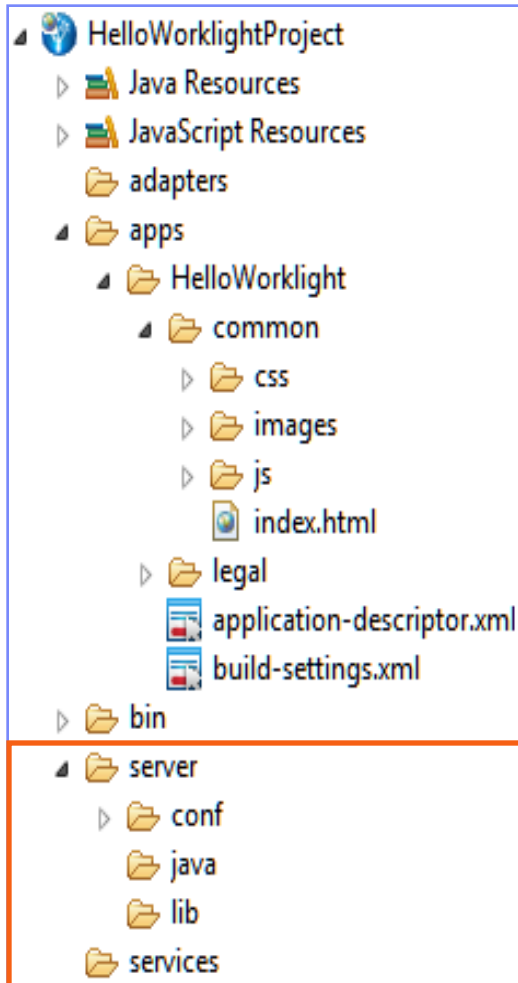
- デフォルト環境は common と呼ばれます。
- **common** 環境には、環境間で共有されるすべてのリソースが入っています。
- **apps** を右クリックし、「新規 (New)」 > 「Worklight 環境 (Worklight environment)」を選択して、新しい環境を追加します。
  - 新しい環境が作成されます。
  - 新しい環境のリソースと共通リソースの間には、以下のような関係があります。
    - **images** – 両方のリソースが同じ名前を持っているとき、共通イメージをオーバーライドします。
    - **css** – 共通 CSS ファイルに対して拡張、オーバーライド、またはその両方を行います。
    - **js** – 共通アプリケーション・インスタンス JS オブジェクトを拡張します。環境クラスは共通アプリケーション・クラスを拡張します。
    - **HTML** – 両方が同じ名前を持つ場合は、共通 HTML コードをオーバーライドします。

# HelloWorklight – プロジェクト構造 - 続き



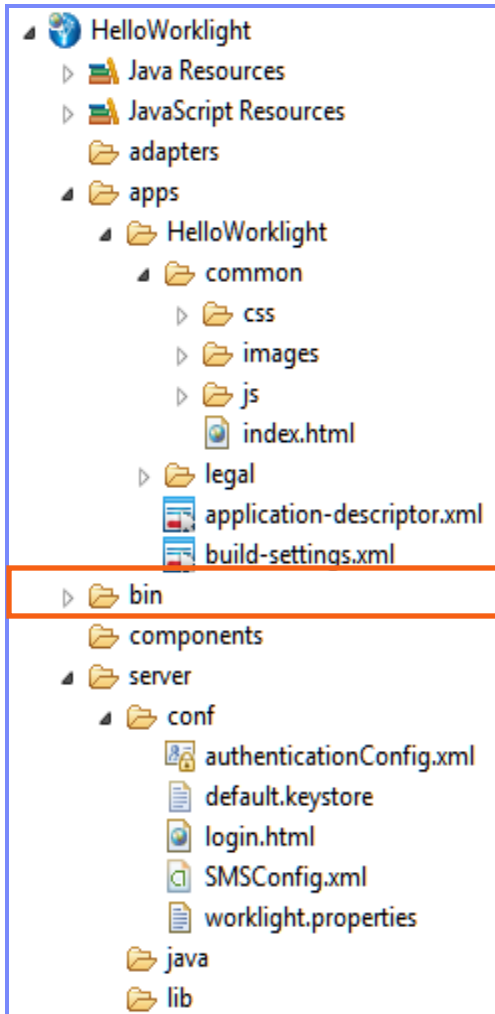
- 共通環境:
  - **Index.html**: メイン HTML ファイル。
  - **css**:
  - **main.css** – メイン・アプリケーション CSS ファイル。
  - **images**: 共通環境のデフォルト Worklight イメージ。
  - **js**:
    - **initOptions.js**: アプリケーションの初期化オプションが入っています。
    - **main.js**: アプリケーションのメイン JavaScript ファイル。
    - **messages.js**: すべてのアプリケーション・メッセージを保持する JSON オブジェクト。翻訳のソースとして使用できます。
- **legal** フォルダーにはすべての法律関連資料があります。
- **application-descriptor.xml** ファイルにはアプリケーション・メタデータが入っています。
- **build-settings.xml** ファイルには、ミニファイおよび連結のための構成オプションが入っています。
- **bin** フォルダーには、リモート Worklight Server へのデプロイに使用できる、生成された **.wlapp** ファイルおよび **.wladapter** ファイルが入っています。

# HelloWorklight – プロジェクト構造 - 続き



- **server** フォルダーには、プロジェクトのサーバー・サイドでのカスタマイズに使用するファイルがあります。
  - **conf** フォルダーには以下のファイルがあります。
    - **authenticationConfig.xml**: 認証レムとセキュリティー・テストを定義します。
    - **default.keystore**: Worklight によって提供されるデフォルトの SSL 証明書です。
    - **login.html**: Web 環境および Worklight Console 用のログイン・フォームを提供します。
    - **SMSConfig.xml**: SMS ゲートウェイを定義します。
    - **worklight.properties**: Worklight Server で使用されるプロパティーを定義します。
  - **java** フォルダーは、アプリケーションのビルド後にコンパイルされて Worklight Server にデプロイされる Java™ クラスを保持するために使用されます。カスタム Java コードをここに入れることができます。カスタム Java コードをここに入れることができます。
  - **lib** フォルダーは、サーバーにデプロイされる JAR ファイルを入れるために使用されます。
- **services** フォルダーには、ディスクカバーされたバックエンド・サービスが入っています。

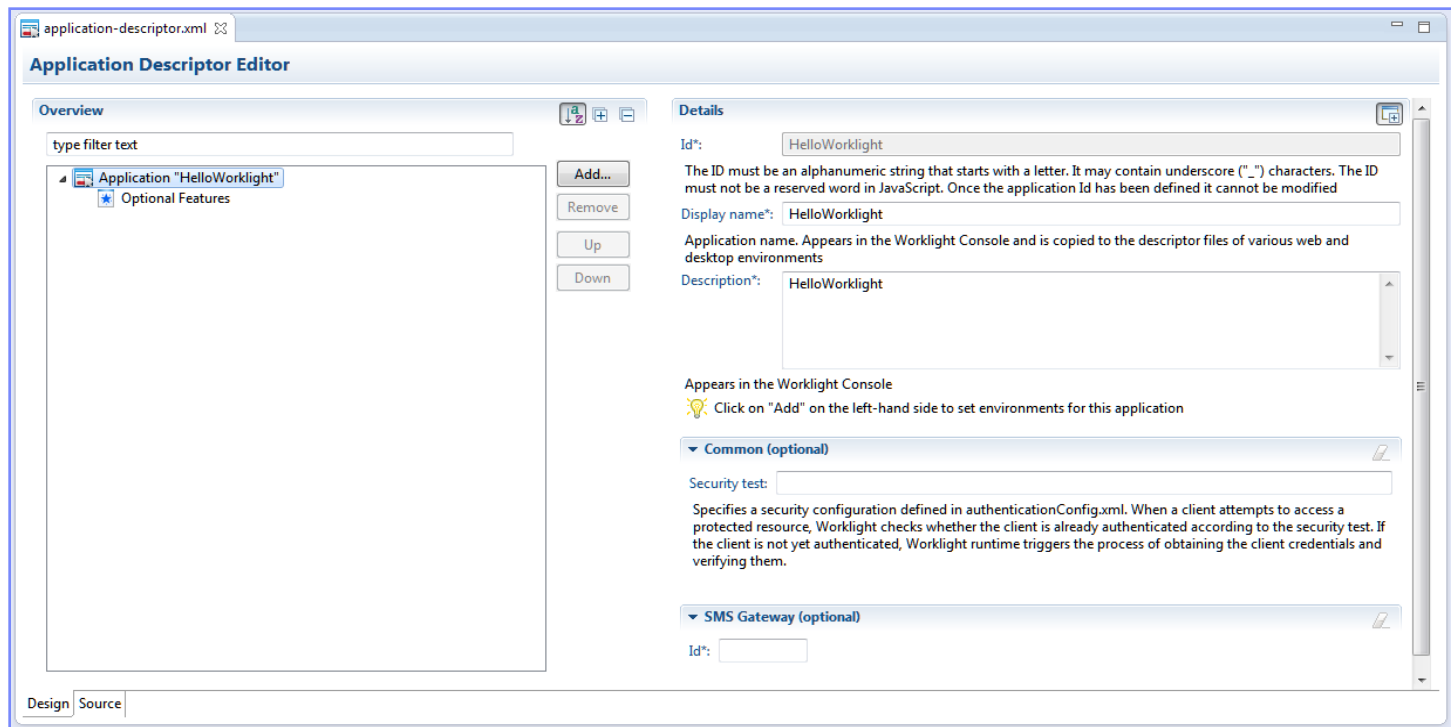
## HelloWorklight – プロジェクト構造 - 続き



- **bin** フォルダーには、Worklight Server にデプロイされるプロジェクト成果物が入っています。
- Worklight Studio の評価版では、これらの成果物は、ビルド・プロセスの一環として自動的に、組み込み Worklight Development Server にデプロイされます。
  - **.wlapp** ファイルはアプリケーション・バンドルです。
  - **.wladapter** ファイルはアダプターです。
  - **.jar** ファイルと **.war** ファイルは、worklight.properties、authenticationConfig.xml、およびカスタム Java コードを含むサーバー・カスタマイズ・ファイルです。

# HelloWorklight – application-descriptor.xml

- アプリケーション・ディスクリプターは、アプリケーション用のメタデータを保管する XML ファイルです。
- このファイルは Design エディターまたは Source エディターで編集できます。



## HelloWorklight - application-descriptor.xml - 続き

- W3C ウィジェットのパッケージ化および構成に基づいています。
- ビルド時に使用されるアプリケーション・プロパティが入っています。

```
<!-- Attribute "id" must be identical to application folder name -->
<application id="HelloWorklight"
  xmlns="http://www.worklight.com/application-descriptor"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <displayName>HelloWorklight</displayName>
  <description>HelloWorklight</description>
  <author>
    <name>application's author</name>
    <email>application author's e-mail</email>
    <copyright>Copyright My Company</copyright>
    <homepage>http://mycompany.com</homepage>
  </author>
  <mainFile>index.html</mainFile>
  <thumbnailImage>common/images/th
  <features>
  </features>
</application>
```

アプリケーション名、説明、作成者の詳細、および Worklight Console に表示するサムネイル・イメージを指定します。

## HelloWorklight - application-descriptor.xml - 続き

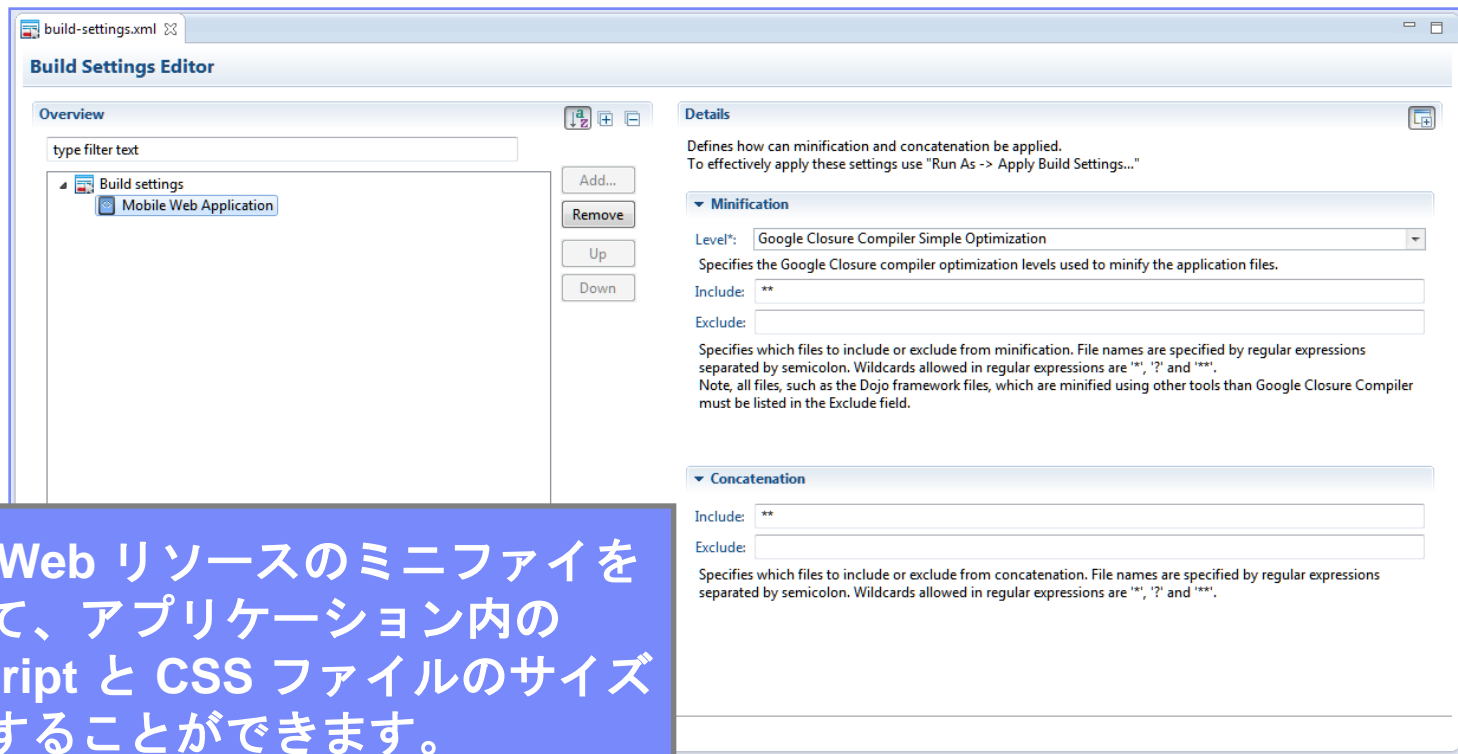
- W3C ウィジェットのパッケージ化および構成に基づいています。
- ビルド時に使用されるアプリケーション・プロパティが入っています。

```
<iphone bundleId="com.HelloWorklight" version="1.0">
  <worklightSettings include="true"/>
  <security>
    <encryptWebResources enabled="false"/>
    <testWebResourcesChecksum enabled="false" ignoreFileExtensions="png, jpg, jpeg, gif, mp4, mp3"/>
  </security>
</iphone>
<android version="1.0">
  <worklightSettings include="true"/>
  <security>
    <encryptWebResources enabled="false"/>
    <testWebResourcesChecksum enabled="false"
    <publicSigningKey>Replace this text with t
  </security>
</android>
```

新しい環境が Worklight プロジェクトに追加されると、環境に固有の情報が自動的に挿入されます。

# HelloWorklight – build-settings.xml

- 「ビルド設定 (Build Settings)」は XML ファイルです。このファイルには、デスクトップ・ブラウザ環境とモバイル Web 環境の Web リソースのミニファイおよび連結のための構成オプションが入っています。
- このファイルは Design エディターまたは Source エディターで編集できます。

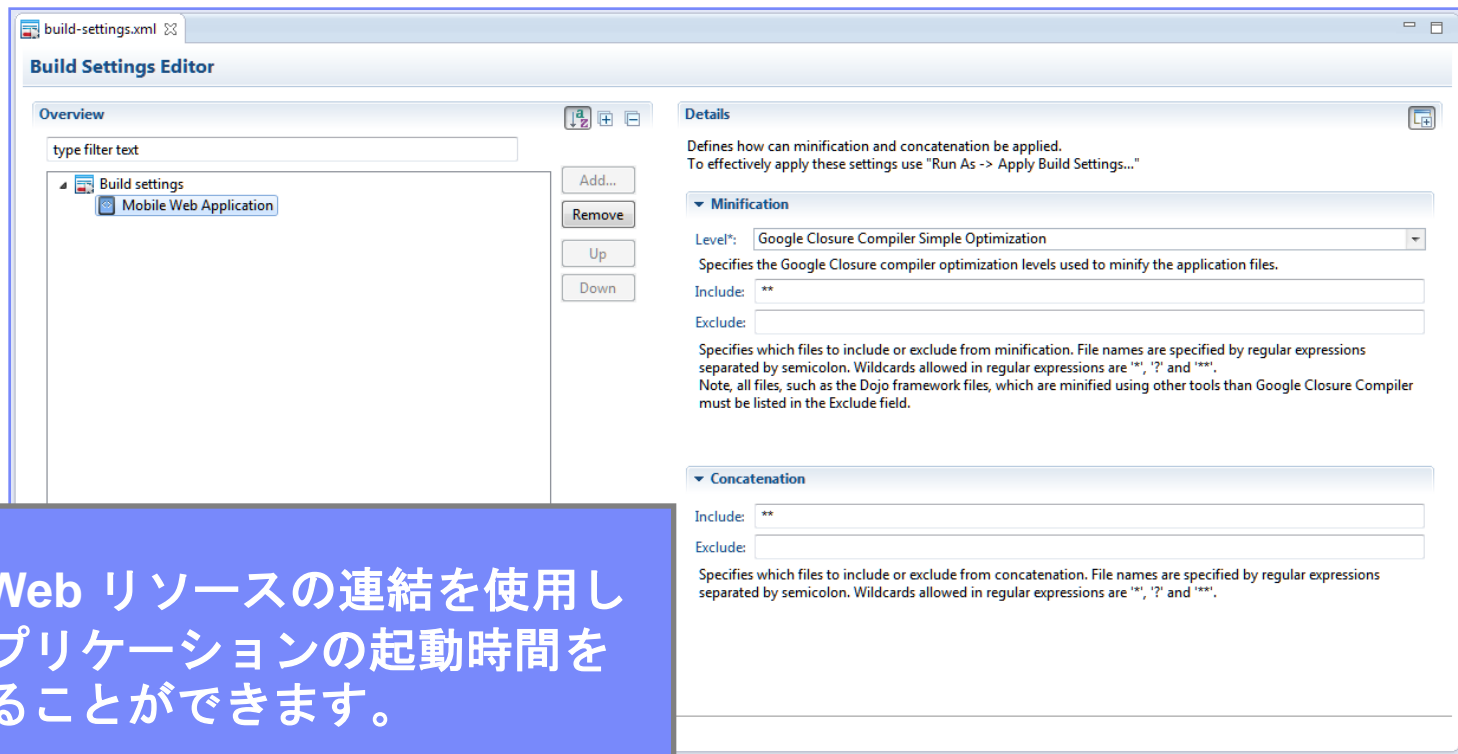


特定の Web リソースのミニファイを使用して、アプリケーション内の JavaScript と CSS ファイルのサイズを削減することができます。



# HelloWorklight – build-settings.xml - 続き

- 「ビルド設定 (Build Settings)」は XML ファイルです。このファイルには、デスクトップ・ブラウザ環境とモバイル Web 環境の Web リソースのミニファイおよび連結のための構成オプションが入っています。
- このファイルは Design エディターまたは Source エディターで編集できます。



# HelloWorklight – シングル DOM モデル

- IBM Worklight ハイブリッド・アプリケーションでは、シングル DOM モデルを使用しています。
- シングル DOM モデルでは、ハイパーリンクを使用したり **window.location** プロパティを変更したりすることによって、さまざまな HTML ファイル間でナビゲートしてはならないことを意味します。
- 代わりに、Ajax 要求を使用して外部 HTML ファイル・コンテンツをロードし、それを既存の DOM に挿入することによって、マルチページ・インターフェースを実装する必要があります。
  - メインのアプリケーション HTML ファイルによって Worklight クライアント・サイド JavaScript フレームワーク・ファイルがロードされると、ブラウザで HTML ファイル間をナビゲートした後 JavaScript のコンテキストやロードされたスクリプトが失われるため、そうする必要があります。
- 今日使用可能なほとんどの JavaScript UI フレームワーク (jQuery Mobile、Sencha Touch、Dojo Mobile など) では、必要なマルチページ・ナビゲーションを行うことができる幅広い API が提供されています。
- このモジュールでは、シングルページ・アプリケーションの原理について説明します。
- シングル DOM モデルを使用してビルドされるマルチページ・アプリケーションの原理については、後続のモジュールで説明します。

# Index.html

- アプリケーションの実行時に、メイン HTML ドキュメントを別の HTML ドキュメントで置換することはできません。

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>index</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="shortcut icon" href="images/favicon.ico">
    <link rel="apple-touch-icon" href="images/apple-touch-icon.png">
    <link rel="stylesheet" href="css/main.css">
    <script>window.$ = window.jQuery = WLJQ; </script>
  </head>
  <body style="display: none;">
    <!--application UI goes here-->
    <p>Hello Worklight</p>
    <script src="js/initOptions.js"></script>
    <script src="js/main.js"></script>
    <script src="js/messages.js"></script>
  </body>
</html>
```

デフォルトのアプリケーション HTML テンプレートは HTML5 標準マークアップに準拠していますが、他の DOCTYPE も指定できます。

## Index.html - 続き

- アプリケーションの実行時に、メイン HTML ドキュメントを別の HTML ドキュメントで置換することはできません。

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>index</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=0">
    <link rel="shortcut icon" href="images/favicon.png">
    <link rel="apple-touch-icon" href="images/apple-touch-icon.png">
    <link rel="stylesheet" href="css/main.css">
    <script>window.$ = window.jQuery = WLJQ;</script>
  </head>
  <body style="display: none;">
    <!--application UI goes here-->
    <p>Hello Worklight</p>
    <script src="js/initOptions.js"></script>
    <script src="js/main.js"></script>
    <script src="js/messages.js"></script>
  </body>
</html>
```

Worklight クライアント・サイド・フレームワークでは、内部機能用に jQuery ライブラリーを使用しています。デフォルトでは、メイン JavaScript ファイル内の内部 jQuery に \$ 文字が割り当てられます。

アプリケーションで jQuery が必要でない場合、または独自の jQuery バージョンを使用したい場合は、この行を削除してください。

## Index.html - 続き

- アプリケーションの実行時に、メイン HTML ドキュメントを別の HTML ドキュメントで置換することはできません。

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>index</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" href="images/favicon.ico">
    <link rel="apple-touch-icon" href="images/apple-touch-icon.png">
    <link rel="stylesheet" href="css/main.css">
    <script>window.$ = window.jQuery = WLJQ;</script>
  </head>
  <body style="display: none;">
    <!--application UI goes here-->
    <p>Hello Worklight</p>
    <script src="js/initOptions.js"></script>
    <script src="js/main.js"></script>
    <script src="js/messages.js"></script>
  </body>
</html>
```

Worklight クライアント・フレームワーク初期化は、initOptions.js ファイル内の onload イベントにバインドされています。初期化オプションについては、IBM Worklight ユーザー文書を参照してください。

## Index.html - 続き

- アプリケーションの実行時に、メインHTMLドキュメントを別のHTMLドキュメントで置換することはできません。

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>index</title>
    <meta name="viewport" content="width=device-widt
    <link rel="shortcut icon" href="images/favicon.png">
    <link rel="apple-touch-icon" href="images/apple-touch-icon.png">
    <link rel="stylesheet" href="css/main.css">
    <script>>window.$ = window.jQuery = WLJQ;</script>
  </head>
  <body style="display: none;">
    <!--application UI goes here-->
    <p>Hello Worklight</p>
    <script src="js/initOptions.js"></script>
    <script src="js/main.js"></script>
    <script src="js/messages.js"></script>
  </body>
</html>
```

青い矢印は、HTMLコードを挿入する場所を示しています。

## initOptions.js

- **initOptions.js** ファイルには、Worklight フレームワーク初期化設定が入っています。
- このファイルはまた、**<body>** 要素のロードが完了した後は、Worklight フレームワークの初期化にも関与します。
- デフォルトでは、Worklight アプリケーションはオフライン・モードで開始します。

```
var wlInitOptions = {  
  connectOnStartup : false,  
  
  logger : {enabled: true, level: 'debug', stringify: true, pretty: false,  
            tag: {level: false, pkg: true}, whitelist: [], blacklist: []},  
  
  analytics : {  
    enabled: false  
  }  
};  
  
if (window.addEventListener) {  
  window.addEventListener('load', function() { WL.Client.init(wlInitOptions); }, false);  
} else if (window.attachEvent) {  
  window.attachEvent('onload', function() { WL.Client.init(wlInitOptions); });  
}
```

- 初期化オプションは、ファイル自体のほかに、IBM Worklight ユーザー文書にも記載されています。

## Main.js

- アプリケーションを作成すると、**main.js** ファイルが作成され、この中にアプリケーションの JavaScript 部分が保持されます。
- ここには、Worklight フレームワークの初期化終了後に自動的に呼び出される **wlCommonInit()** 関数が含まれています。
- アプリケーション初期化コードをそこに追加できます。
- この関数は、共通の初期化開始点を持たせるために、環境に固有の JavaScript ファイルで使用されます。後続の「入門」モジュールに詳しい説明があります。

```
function wlCommonInit(){  
    // Common initialization code goes here  
}
```



## Main.js - 続き

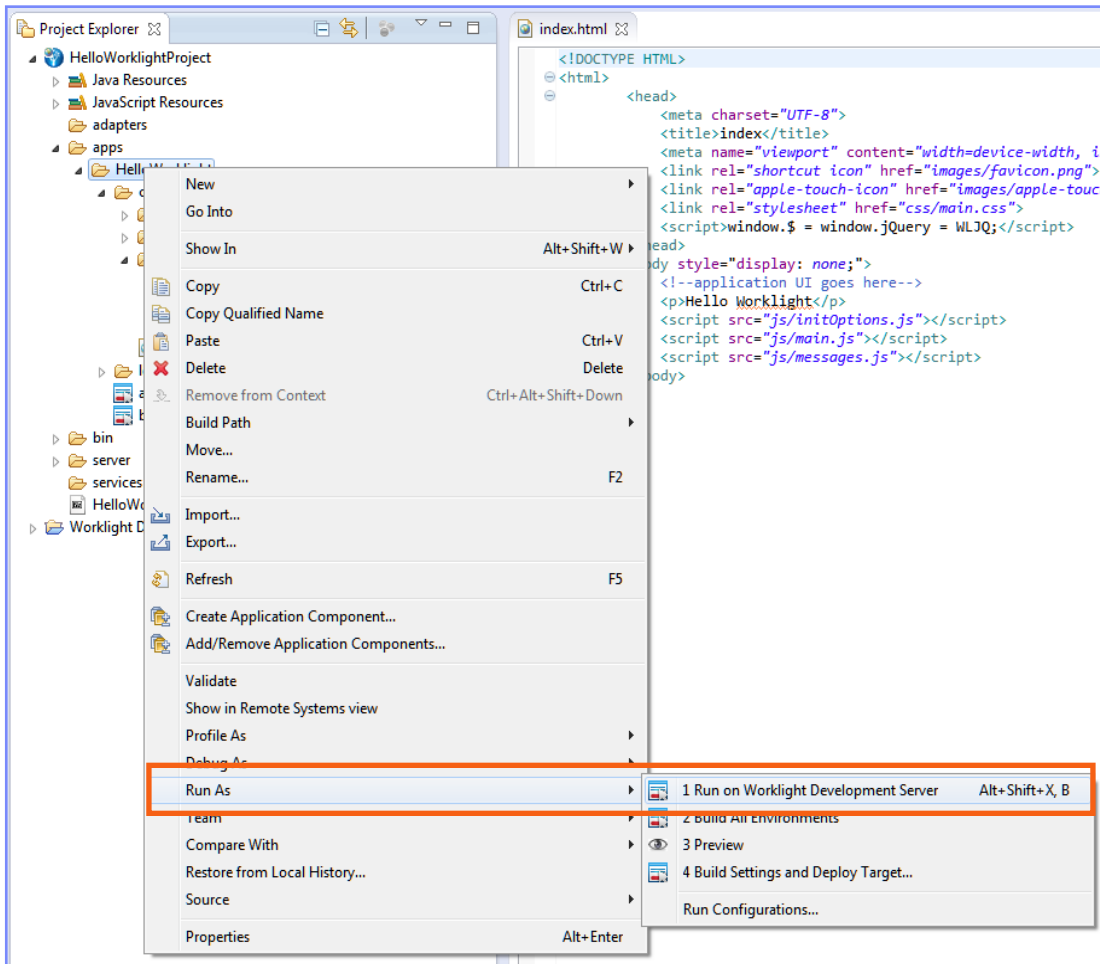
- 前述のとおり、Worklight アプリケーションは、デフォルトではオフライン・モードで開始されます。
- Worklight Server との通信を開始するには、デフォルト **wlCommonInit()** 関数の説明に従う必要があります。

```
function wlCommonInit(){
  /*
   * Application is started in offline mode as defined by a connectOnStartup p
   * In order to begin communicating with Worklight server you need to either:
   *
   * 1. Change connectOnStartup property in initOptions.js to true.
   *    This will make Worklight framework automatically attempt to connect to
   *    Keep in mind - this may increase application start-up time.
   *
   * 2. Use WL.Client.connect() API once connectivity to a Worklight server is
   *    This API needs to be called only once, before any other WL.Client meth
   *    Don't forget to specify and implement onSuccess and onFailure callback
   *
   *    WL.Client.connect({
   *      onSuccess: onConnectSuccess,
   *      onFailure: onConnectFailure
   *    });
   */
}
```

# アジェンダ

- 初めてのアプリケーションを作成する
- アプリケーション・ファイルと HTML 構造を理解する
- Worklight Studio を使用してアプリケーションをデプロイする
- Worklight Studio に関するヒント
- 演習

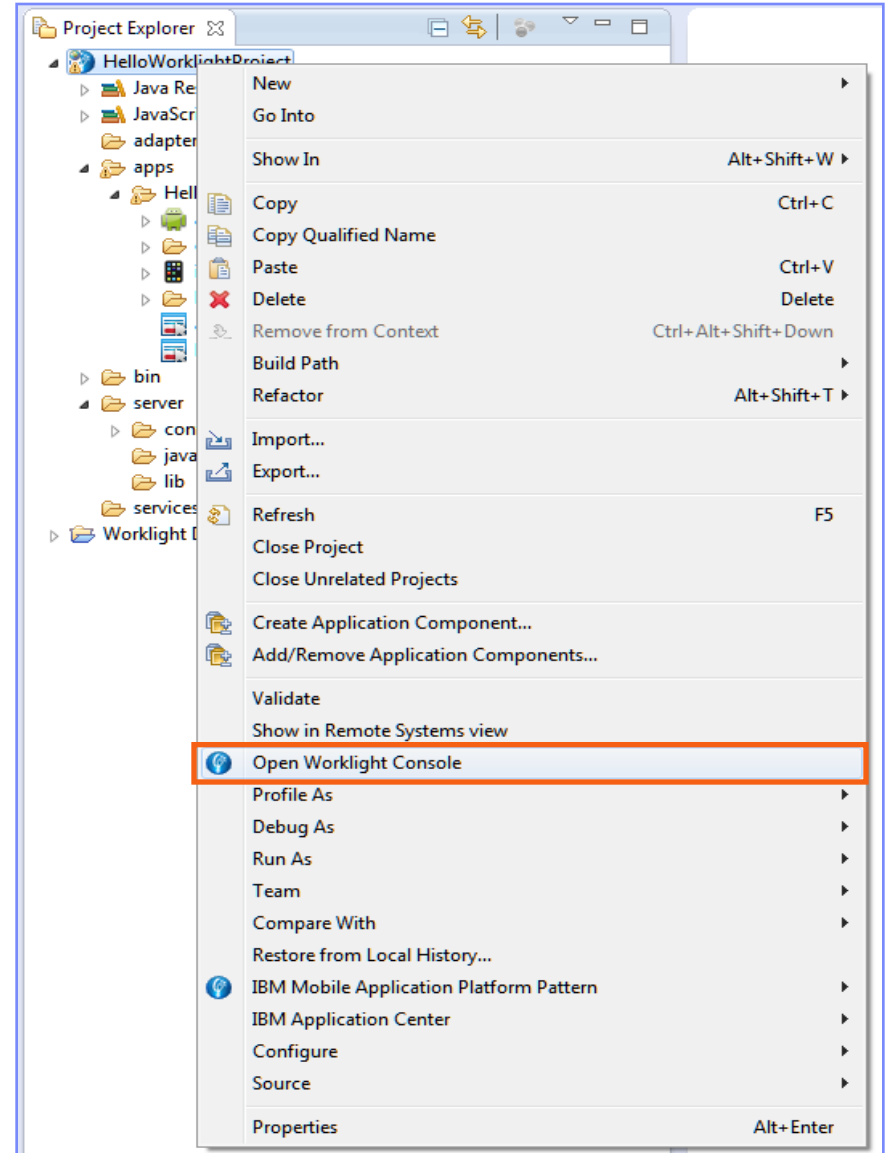
# アプリケーションのビルド



- ビルドするアプリケーションを選択し、アプリケーション名を右クリックします。
- 「実行 (Run As)」 > 「Worklight Development Server 上で実行 (Run on Worklight Development Server)」をクリックします。
- ビルドの完了後、Worklight Console の「カタログ (catalog)」タブでアプリケーションのプレビューを表示できます。
- デフォルトでは、ポート番号は **10080**、コンテキスト・ルートはプロジェクト名です。

# Worklight Console

- Console を実行するには、プロジェクト名を右クリックし、  
**「Worklight Console を開く (Open Worklight Console)」** を選択します。
- 外部ブラウザ・ウィンドウで Worklight Console を表示するには、Eclipse の上部メニュー・バーから  
**「ウィンドウ」 > 「設定」 > 「一般」 > 「Web ブラウザー」** に進み、**「外部 Web ブラウザーを使用」** ラジオ・ボタンを選択します。ここでは、お好みのブラウザを選択することもできます。

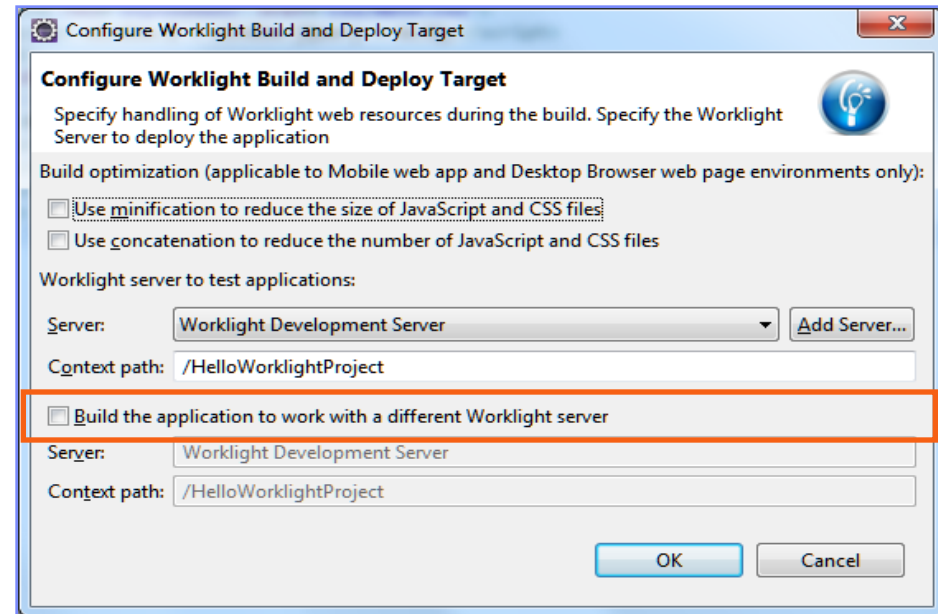


## Worklight Console - 続き

- IBM Worklight V6.0.0.0 以降、Worklight Console URL は以下のように変更されています。
  - ポートは **8080** から **10080** に変更されました。
  - URL にもプロジェクトのコンテキスト・ルートが含まれるようになりました。
    - 例: <http://localhost:10080/MyProject/console>
- 「入門」モジュールが進むにつれ、Worklight Console のさまざまな機能が徐々に明確になり理解できます。

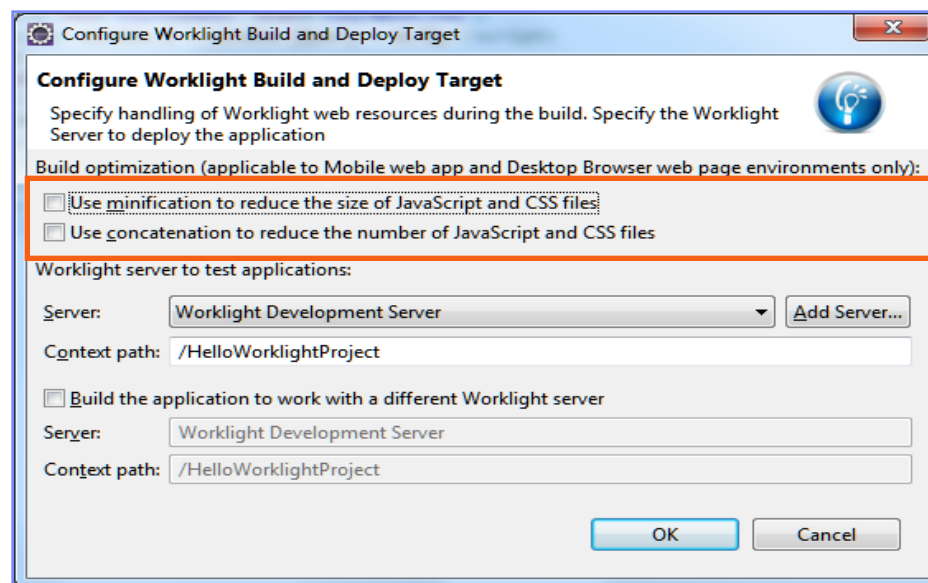
# ビルド設定およびデプロイ・ターゲット

- ターゲット・サーバーを変更するには、アプリケーション名を右クリックし、「実行」>「ビルド設定およびデプロイ・ターゲット (Build Settings and Deploy Target)」を選択します。
- 「他の Worklight サーバーと連携するようにアプリケーションをビルドする (Build the application to work with a different Worklight server)」チェック・ボックスを選択します。
- サーバーとコンテキスト・パスを更新し、「OK」をクリックします。
- 次に、この時点でアプリケーションを再ビルドする必要があります。
- 注: Apache Tomcat および WebSphere® Application Server Liberty プロファイルのみがサポートされます。



## ビルド設定およびデプロイ・ターゲット - 続き

- ミニファイと連結のいずれかまたは両方がアプリケーションに必要な場合は、アプリケーションをビルドする前に、このビルド設定も有効にしておく必要があります。
- そうするには、アプリケーション名を右クリックし、「実行」>「ビルド設定およびデプロイ・ターゲット (Build Settings and Deploy Target)」を選択します。
- 強調表示されたオプションの1つまたはすべてを選択します。
- 「OK」をクリックします。



# アジェンダ

- 初めてのアプリケーションを作成する
- アプリケーション・ファイルと HTML 構造を理解する
- Worklight Studio を使用してアプリケーションをデプロイする
- Worklight Studio に関するヒント
- 演習



## Worklight Studio に関するヒント

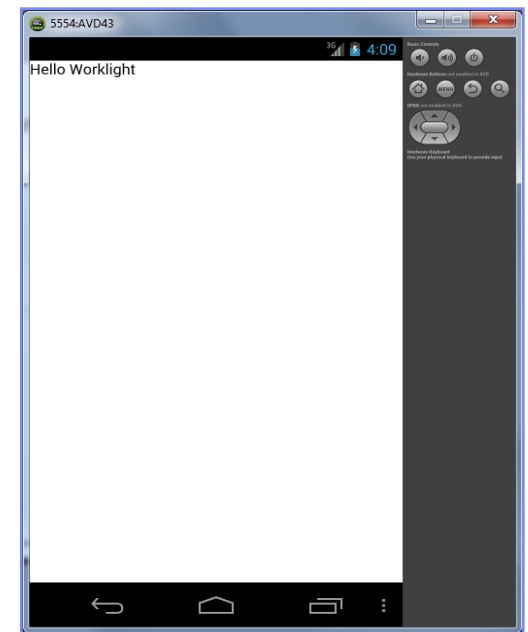
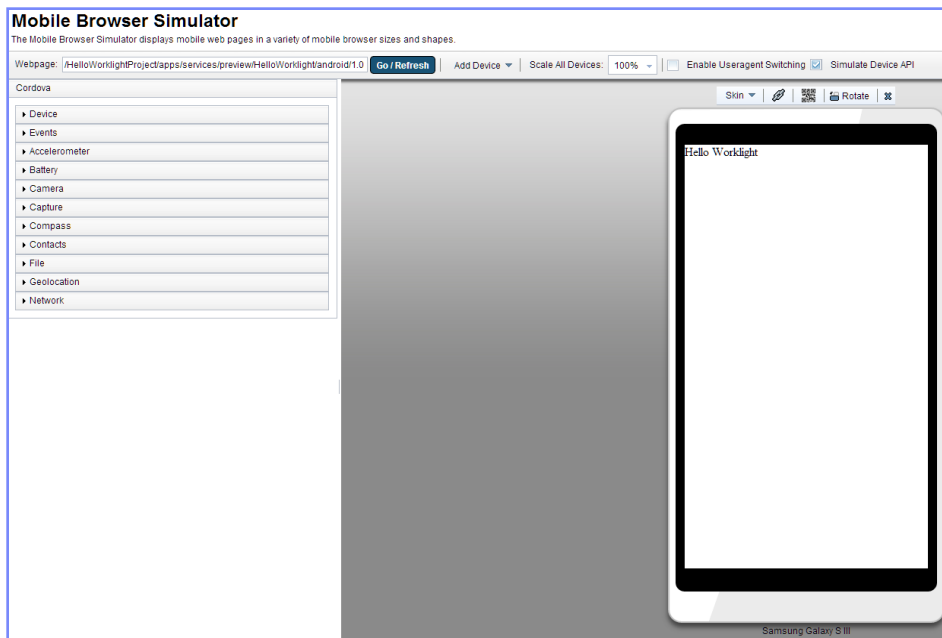
- Worklight Studio V6.1.0 以降、行われた変更を表示するために開発中にアプリケーションを再ビルドしてデプロイする必要はなくなりました。
- それどころか、作業しているファイルを保存するだけです。
- アプリケーションをプレビューし (または既にプレビューしている場合は)、表示しているタブを単に更新すると、Worklight Studio に対して行われた変更がその後の検査およびデバッグに直ちに利用されるようになります。

# アジェンダ

- 初めてのアプリケーションを作成する
- アプリケーション・ファイルと HTML 構造を理解する
- Worklight Studio を使用してアプリケーションをデプロイする
- Worklight Studio に関するヒント
- 演習

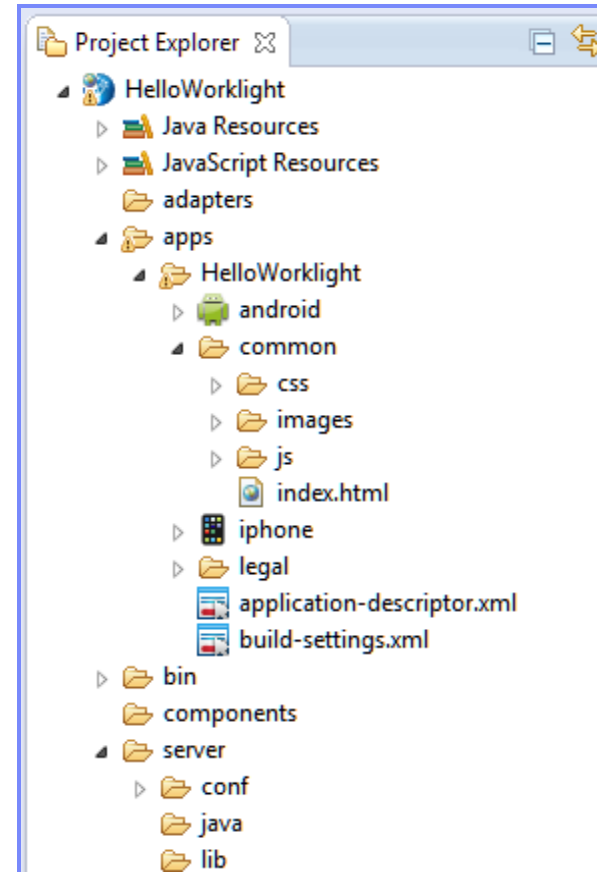
# 演習

- HelloWorklight アプリケーション
  - ウィザードを使用して **HelloWorklightProject** プロジェクトを作成します。
  - ウィザードを使用してアプリケーションを作成し、それに、**HelloWorklight** という名前を付けます。
  - Worklight Studio を使用してアプリケーションをデプロイします。
  - Worklight Console を使用してアプリケーションをプレビューします。



## 演習ソリューション

- この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。
- ソリューションを Eclipse にインポートするには、「プロジェクト・エクスプローラー」ビュー内のブランクの領域を右クリックし、「インポート」>「インポート」>「一般」>「既存プロジェクトをワークスペースへ」を選択して、アーカイブ (圧縮ファイル) またはルート・ディレクトリー (解凍されている場合) のいずれかを選択します。



# 確認テスト

- デフォルト・アプリケーション構造とは
  - 1つのHTMLファイルと、いくつかのCSSファイルおよびJSファイル。
  - いくつかのHTMLファイルおよびJSファイルと、1つのCSSファイル。
  - いくつかのHTMLファイル、JSファイル、およびCSSファイル。
  - 1つのHTMLファイル、CSSファイル、およびJSファイル。
- Worklight プロジェクトの一部として追加できないのは、どのサーバー・サイド・コンポーネントでしょうか
  - カスタム・サーバー・サイド Java クラスのソース・コード。
  - サーバーで使用されるプロパティ。
  - サーバー・サイド Java コードを含む JAR ファイル。
  - サーバー・パフォーマンスをテストするための JMeter アプリケーション。
- アプリケーションで使用する URL を指定するための正しい場所は
  - authenticationConfig.xml
  - worklight.properties
  - application-descriptor.xml
  - URL は専用の設定ファイルでプラットフォームごとに個別に設定される。

## 確認テスト

- デフォルト・アプリケーション構造とは
  - 1つのHTMLファイルと、いくつかのCSSファイルおよびJSファイル。
  - いくつかのHTMLファイルおよびJSファイルと、1つのCSSファイル。
  - いくつかのHTMLファイル、JSファイル、およびCSSファイル。
  - 1つのHTMLファイル、CSSファイル、およびJSファイル。
- Worklight プロジェクトの一部として追加できないのは、どのサーバー・サイド・コンポーネントでしょうか
  - カスタム・サーバー・サイド Java クラスのソース・コード。
  - サーバーで使用されるプロパティ。
  - サーバー・サイド Java コードを含む JAR ファイル。
  - サーバー・パフォーマンスをテストするための JMeter アプリケーション。
- アプリケーションで使用する URL を指定するための正しい場所は
  - authenticationConfig.xml
  - worklight.properties
  - application-descriptor.xml
  - URL は専用の設定ファイルでプラットフォームごとに個別に設定される。

# 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

