

HP Anywhere IDE

Windows

ソフトウェアバージョン: 10.10

開発者ガイド (IntelliJ IDEA)

ドキュメントリリース日: 2013年12月 (英語版)

ソフトウェアリリース日: 2013年12月 (英語版)



ご注意

保証

HP製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載は、追加保証を提供するものではありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

権利の制限

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。商用コンピューターソフトウェア、コンピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR 12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

著作権について

© Copyright 2012 - 2013 Hewlett-Packard Development Company, L.P.

商標について

Adobe®は、Adobe Systems Incorporated (アドビシステムズ社) の登録商標です。

Microsoft®およびWindows®は、米国におけるMicrosoft Corporationの登録商標です。

OracleとJavaは、Oracle Corporationおよびその関連会社の登録商標です。

ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

- ソフトウェアバージョンの番号は、ソフトウェアのバージョンを示します。
- ドキュメントリリース日は、ドキュメントが更新されるたびに更新されます。
- ソフトウェアリリース日は、このバージョンのソフトウェアのリリース期日を表します。

更新状況、およびご使用のドキュメントが最新版かどうかは、次のサイトで確認できます。

<http://h20230.www2.hp.com/selfsolve/manuals>

このサイトを利用するには、HP Passportへの登録とサインインが必要です。HP Passport IDの登録は、次のWebサイトから行なうことができます。

<http://h20229.www2.hp.com/passport-registration.html>

または、HP Passportのログインページの [**New users - please register**] リンクをクリックします。

適切な製品サポートサービスをお申し込みいただいたお客様は、更新版または最新版をご入手いただけます。詳細は、HPの営業担当にお問い合わせください。

サポート

HPソフトウェアサポートオンラインWebサイトを参照してください。

<http://www.hp.com/go/hpsoftwaresupport>

このサイトでは、HPのお客様窓口のほか、HPソフトウェアが提供する製品、サービス、およびサポートに関する詳細情報をご覧いただけます。

HPソフトウェアオンラインではセルフソルブ機能を提供しています。お客様のビジネスを管理するのに必要な対話型の技術サポートツールに、素早く効率的にアクセスできます。HPソフトウェアサポートのWebサイトでは、次のようなことができます。

- 関心のあるナレッジドキュメントの検索
- サポートケースの登録とエンハンスメント要求のトラッキング
- ソフトウェアパッチのダウンロード
- サポート契約の管理
- HPサポート窓口の検索
- 利用可能なサービスに関する情報の閲覧
- 他のソフトウェアカスタマーとの意見交換
- ソフトウェアトレーニングの検索と登録

一部のサポートを除き、サポートのご利用には、HP Passportユーザーとしてご登録の上、サインインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport IDを登録するには、次のWebサイトにアクセスしてください。

<http://h20229.www2.hp.com/passport-registration.html>

アクセスレベルの詳細については、次のWebサイトをご覧ください。

http://h20230.www2.hp.com/new_access_levels.jsp

目次

開発者ガイド (IntelliJ IDEA)	1
目次	5
HP Anywhere IDE (IntelliJ IDEA) でアプリを作成する理由	6
HP Anywhere IDEでアプリをビルドして実行するための手順	7
HP Anywhere IDEでアプリプロジェクトを作成する方法	8
アプリにサードパーティのライブラリを追加する方法	13
サポートされているサードパーティのライブラリ	13
アプリプロジェクトのフォームファクターを追加/削除する方法	14
アプリの構成を定義する方法	16
ユーザー設定	16
管理者設定	16
データソース設定	17
アプリを実行して表示する方法	18
アプリの実行方法	18
アプリをミニファイする方法	21
アプリの変更時にホットデプロイと同期を有効にする方法	23
付録1: アプリプロジェクトのフォルダー構造	24
付録2: テンプレートについて	27
空のプロジェクトテンプレート	28
ネイティブアプリ用テンプレート	28
Enyoベースのアプリ用テンプレート	29
Senchaベースのアプリ用テンプレート	29
jQuery Mobileベースのアプリ用テンプレート	30
AngularJSベースのアプリ用テンプレート	31
付録3: トラブルシューティング	32
Mavenプロキシを設定する方法	32
getUserNameメソッド (UserInfoService API) でNULLが返る場合の対処方法	32

HP Anywhere IDE (IntelliJ IDEA) でアプリを作成する理由

HP Anywhereには、アプリの作成、ビルド、実行、デバッグができるIntelliJ IDEAベースのIDEがあります。

このIDEはアプリのライフサイクル全体をカバーする包括的な開発環境で、アプリプロジェクトの作成、デバッグ、保守に対応しています。HP Anywhere IDEには次のものが含まれます。

- アプリプロジェクト作成用 **ウィザード**
- アプリの実行とデバッグのための、任意のフォームファクター (デスクトップ、タブレット、スマートフォン) と画面解像度に対応した **内蔵シミュレーター**
- ローカルでデプロイされたアプリ用の **ホットデプロイ**、アプリを常に実行可能にする **自動ミニファイ** など、アプリを公開するデプロイ機能
- データベース不要でデバッグにも対応した、事前構成済み **HP Anywhereサーバー**。開発環境用にカスタマイズ/最適化されています。

HP Anywhere IDEでアプリをビルドして実行するための手順

HP Anywhere IDEでアプリをビルドして実行するには、次の手順を実行します。

1. HP Anywhere IDEがコンピューターにインストールされていることを確認します。
2. HP Anywhere App Projectウィザードで、アプリプロジェクトのパッケージとファイルを作成します。これにより、必要なファイル、クラス、メソッドを含む実行可能なアプリが作成されます。用意されているテンプレートを基にプロジェクトを作成できます。詳細については、「[HP Anywhere IDEでアプリプロジェクトを作成する方法](#)」(8ページ)、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)、「[付録2: テンプレートについて](#)」(27ページ)を参照してください。
3. コードを追加します。
4. アプリの構成を定義します。詳細については、「[アプリの構成を定義する方法](#)」(16ページ)を参照してください。
5. 内蔵のHP Anywhereローカルサーバーまたはリモートサーバーで、アプリの実行とデバッグを行います。詳細については、「[アプリの実行方法](#)」(18ページ)を参照してください。
6. アプリが完成したら、.zip形式のアプリコンテンツパックをHP Anywhere管理者に渡します。管理者はこのファイルをHP Anywhereサーバーにデプロイし、カタログ内で有効にします。この.zipファイルは、次の命名規則に従います。<アプリ名>-cp.zip。ファイルがある場所は、<アプリプロジェクト>\targetフォルダーです。

HP Anywhere IDEでアプリプロジェクトを作成する方法

アプリに必要なファイルやクラスを備えた新しいHP Anywhere IDE プロジェクトを作成するには、HP Anywhere App Projectウィザードを使用します。

HP Anywhere IDEでアプリを作成すると、<アプリプロジェクト>\targetフォルダーに、.zipファイル形式でアプリコンテンツバックが生成されます。

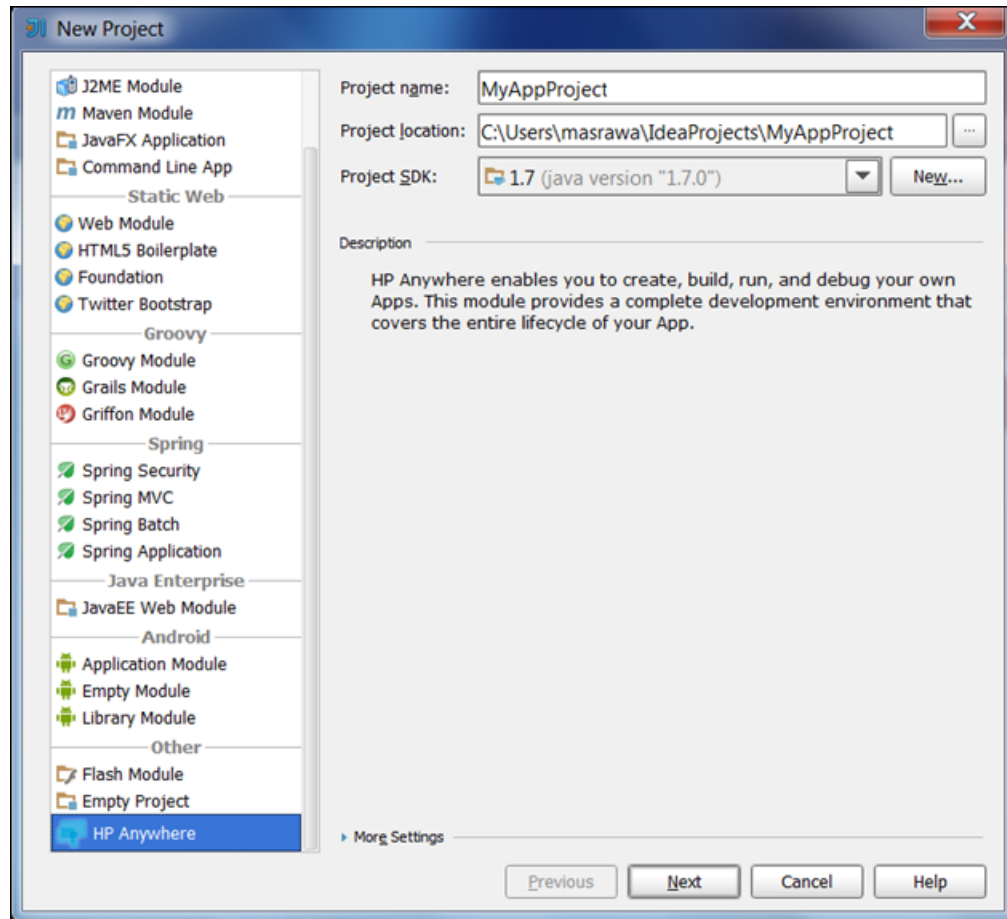
この.zipファイルは、次の命名規則に従います。<アプリ名>-cp.zip。アプリが完成したら、このファイルをHP Anywhere管理者に渡す必要があります。管理者はこのファイルをHP Anywhereサーバーにデプロイし、カタログ内で有効にします。

注: プロキシ経由でインターネットに接続した場合は、アプリの初回作成時にMavenプロキシを構成する必要があります。詳細については、「[Mavenプロキシを設定する方法](#)」(32ページ)を参照してください。

HP Anywhere IDEでHP Anywhereアプリプロジェクトを作成するには、次の手順を実行します。

1. **前提条件:** IntelliJ IDEAのMavenホームディレクトリ ([設定] > [Maven] > [Maven home directory]) を<%BTOA_HOME%>/apache-maven-3.0.4に設定します。
(この設定を適用するには、[Override] チェックボックスを選択する必要があります。)
例: C:\HP\HPAnywhere\apache-maven-3.0.4
2. 次のいずれかの操作を実行します。
 - [IntelliJ IDEA Welcome] ウィンドウの [Quick Start] ペインで、[Create New Project] を選択します。
 - IntelliJ IDEAで、[ファイル] > [New Project] を選択します。

New Projectウィザードが開きます。



注: 次の例に示すフィールドは、IntelliJ IDEAのバージョンによって異なることがあります。

3. 次の手順を実行します。

- 左側のペインで、HP Anywhereを選択します。
- 右側のペインで、必要に応じて **[Project name]** と他のフィールドに入力します。

4. **[Next]** をクリックします。[HP Anywhere IDE New Project] 画面が開きます。

The screenshot shows the 'New Project' dialog box in the HP Anywhere IDE. The dialog is titled 'New Project' and contains several sections:

- Project Template:** A dropdown menu is set to 'My Report Enyo Project'. Below it, a description reads: 'An HP Anywhere starter project that generates an App using the Enyo framework. This project is targeted for use on tablets and smartphones.'
- Form Factor:** Three checkboxes are present: 'Smartphone', 'Tablet', and 'Desktop'. All three are checked.
- App Preferences:** A checkbox labeled 'Allow an app to run in offline mode' is present and is unchecked.
- Project Details:** Several text input fields are filled with the following values:
 - App Name: enyoProject
 - Group ID: com.hp.my.miniapp
 - Source package: com.hp.miniapp.sample
 - App version: 1.0.0
 - App icon: ...

At the bottom right of the dialog, there are four buttons: 'Previous', 'Finish', 'Cancel', and 'Help'.

5. [Project Template] 領域で [Type] ドロップダウンリストからテンプレートタイプを選択します。各テンプレートにより、独自のコードを追加できるスタータープロジェクトが生成されます。テンプレートは数種類から選択できます。

- **空のプロジェクト**: アプリケーション内容を含まない、プロジェクトフォルダー構造のみのHP Anywhereスタータープロジェクトです。
- **Hello Worldプロジェクト**: 指定されたフレームワーク (ネイティブ、Enyo、JQM、Sencha、AngularJSなど) を使用してアプリを生成するHP Anywhereスタータープロジェクトです。タブレット やスマートフォン用のプロジェクトです。
- **My Reportプロジェクト**: レポートの一覧を表示できるアプリを生成するHP Anywhereスタータープロジェクトです。これらのプロジェクトでは、指定されたフレームワーク (ネイティブ、Enyo、JQM、Sencha、AngularJSなど) が使用されます。タブレット やスマートフォン用のプロジェクトです。

[Description]ボックスには、選択したテンプレートについての詳細情報が表示されます。

各テンプレートの詳細については、「[付録2: テンプレートについて](#)」(27ページ)を参照してください。

6. [Form Factor] 領域で、1つまたは複数のフォームファクターを選択します。次のフォームファクターを選択できます。

- **Smartphone**
- **Tablet**
- **Desktop**

注: フォームファクターはプロジェクト作成後にも手動で追加できます。詳細については、「[アプリプロジェクトのフォームファクターを追加/削除する方法](#)」(14ページ)を参照してください。

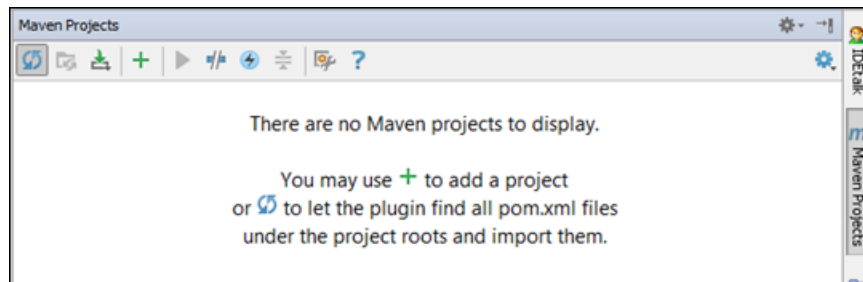
7. インターネットに接続しないでアプリを使用する場合は、[App Preferences] 領域で [**Allow an app to run in offline mode**] チェックボックスを選択します。

注: このチェックボックスを選択しただけでは、オフラインサポートは有効になりません。オフラインモードのサポートには、コードの追加も必要です。詳細については、Developer Zoneのアプリ使用例を参照してください。

8. [Project Details] 領域で、Maven命名規則に基づいて以下の内容を定義します。

- **App Name**: アプリの名前です。たとえば、"HelloWorld"。(英数字しか使用できません。特殊文字や空白は使用できません)。

- **Group ID:** 逆表記のドメイン名です。たとえば、"com.mycompany.mygroupname"。グループIDは、生成されたPOM.XMLプロジェクトファイルで使用されます。
 - **Source package:** 有効なJavaパッケージ名です。たとえば、"com.my.mypackage"。パッケージ名に基づいてワークスペースにフォルダー構造が生成されます。
 - **App version:** アプリのバージョンです。たとえば、"1.0.0"。
 - **App icon:** PNGファイルで作成したアプリのアイコンです。アイコンはHP Anywhereの [マイアプリ] ページに表示されます。
9. [Finish] をクリックします。HP Anywhereによってアプリプロジェクトが生成されます。プロジェクトが生成される間、[Console] タブでログを確認できます。(プロジェクトの初回作成時は少し時間がかかることがあります)。
10. プロジェクトを次のようにインポートします。
- a. [Maven Projects] ペインを開きます。



- b. [Reimport All Maven Projects] ボタンをクリックします。
11. プロジェクトがインポートされた後は、次の操作を行えます。
- プロジェクトのフォルダー構造の確認。詳細については、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)を参照してください。
 - アプリプロジェクトへのコードの追加。
 - アプリの実行。詳細については、「[アプリの実行方法](#)」(18ページ)を参照してください。

アプリにサードパーティのライブラリを追加する方法

`pom.xml`にサードパーティの依存関係を追加できます。このファイルの更新後は、[Maven Projects] ペインの [Reimport All Maven Projects] ボタンをクリックしてIDEを更新する必要があります。

サポートされているサードパーティのライブラリ

アプリでは、次のサードパーティのライブラリを使用できます。

- Spring (バージョン3.0.5)
- Jackson (バージョン1.5.5)
- Jersey (バージョン1.5)
- JSR 311-api (バージョン1.1.1)
- Apache Wink Client (バージョン1.1.1)
- Hibernate Validator (バージョン4.2.0)
- CGLIB (バージョン2.2)
- SLF4J LOG4J (バージョン12 1.5.11)

アプリプロジェクトのフォームファクターを追加/削除する方法

アプリプロジェクトを作成時する際、アプリを実行するフォームファクター (デスクトップ/タブレット/スマートフォン) を指定します。

プロジェクト作成時に指定しなかったフォームファクターがある場合、アプリのパッケージに必要なフォルダーを追加することで後からフォームファクターを追加できます。同様に、関連するプロジェクトフォルダーを削除することでフォームファクターを削除することもできます。

アプリのフォームファクターを追加するには、次の手順を実行します。

1. HP Anywhere IDE パッケージエクスプローラーでメインのアプリフォルダーを展開します。
2. 次の各フォルダーにある既存のフォームファクター (desktop、tablet、またはsmartphone) のフォルダーをコピーアンドペーストします。
 - src > webapp > app
 - src > webapp > css
 - src > webapp > images
3. 新しいフォルダーの名前を追加するフォームファクター (desktop、tablet、またはsmartphone) に変更します。
4. 新しいフォームファクターフォルダーにクライアントコードを追加します。
5. **src/resources > descriptor.xml**に各フォームファクター用のセクションを作成します。次のセクションをコピーして、<既存のフォームファクタータイプ> を <新しいフォームファクタータイプ> に置き換えます。たとえば、次のセクションをコピーして、**DESKTOP**を**SMARTPHONE**に置き換えます。

```
<webResources>
  <webResource>
    <formFactor>DESKTOP</formFactor>
    <jsNames>
      <jsName>app.js</jsName>
    </jsNames>
    <cssNames>
      <cssName>css/common/app.css</cssName>
    </cssNames>
    <iconResource>images/common/default-app.png</iconResource>
  </webResource>
```

フォームファクターを削除する方法

HP Anywhere IDE パッケージエクスプローラーで次の手順を実行します。

1. メインのアプリフォルダーを展開します。
2. **src/resources > descriptor.xml**から関連するセクションを削除します。たとえばデスクトップのフォームファクターを削除するには、次の部分を削除します。

```
<webResources>
  <webResource>
    <formFactor>DESKTOP</formFactor>
    <jsNames>
      <jsName>app.js</jsName>
    </jsNames>
    <cssNames>
      <cssName>css/common/app.css</cssName>
    </cssNames>
    <iconResource>images/common/default-app.png</iconResource>
  </webResource>
```

3. (オプション) 次の各フォルダーで、削除する各フォームファクター (desktop、tablet、または smartphone) のフォルダーを削除します。
 - **src > webapp > app**
 - **src > webapp > css**
 - **src > webapp > images**

アプリの構成を定義する方法

アプリを作成したら、`src/resources`フォルダーにあるアプリの設定を構成できます。

ユーザー設定

アプリのユーザーに関する設定です。

ユーザー設定を変更するには、次の手順を実行します。

1. パッケージエクスプローラーで、/`<アプリのメインノード>/src/resources/<アプリ名>-user-settings.xml`を探し、ダブルクリックして開きます。
2. [`<アプリ名>-user-settings.xml`] タブで、[Design] タブを選択します。
3. [`context`] を右クリックし、[Add Child] > [setting] を選択します。
4. 追加した設定ノードを展開し、次の項目に新しい値を設定します。
 - a. **name**: この設定のIDです。
 - b. **namekey**: ローカライズ用の設定です。
 - c. **integer**: 既定の設定タイプです。変更するには、[Replace With] `<値のタイプ>` を右クリックします。
可能な値:
boolean
enumeration
float
password
string
xml

管理者設定

アプリの管理者に関する設定です。

管理者設定を変更するには、次の手順を実行します。

1. パッケージエクスプローラーで、/`<アプリのメインノード>/src/resources/<アプリ名>-admin-settings.xml`を探し、ダブルクリックして開きます。
2. [`<アプリ名>-admin-settings.xml`] タブで、[Design] タブを選択します。
3. [`context`] を右クリックし、[Add Child] > [setting] を選択します。
4. 追加した設定ノードを展開し、次の項目に新しい値を設定します。
 - a. **name**: この設定のIDです。
 - b. **settingType**: [`global`] を指定します。

- c. **displayInUI**: HP Anywhere 管理者コンソールにアプリを表示するかどうかを指定します。
可能な値:
true
false
- d. **sectionKey**: 管理者コンソールの [Settings] タブに表示されるグループ領域の名前です ([List of Categories] でアプリを選択したときに表示されます)。
- e. **integer**: 既定の設定タイプです。変更するには、[Replace With] <値のタイプ> を右クリックします。
可能な値:
boolean
enumeration
float
password
string
xml

データソース設定

アプリとバックエンドサーバーとの通信に関する設定です。

1. パッケージエクスプローラーで、/**<アプリのメインノード>/src/resources/<アプリ名>-ds-provider.xml**を探し、ダブルクリックして開きます。
2. [**アプリ名>-admin-settings.xml**] タブで、[Source] タブを選択します。
3. 必要に応じて構成を変更します。

アプリを実行して表示する方法

アプリは、ローカルまたはリモートサーバーで、デバッグモードまたは実行モードで実行できます。HP Anywhere IDEには、ホットデプロイに使用できるローカルサーバーが内蔵されています。

アプリを実行すると、HP Anywhereによって次の処理が実行されます。

1. HP Anywhereサーバーが起動されます (すでに起動されていない場合)。
2. 必要に応じて、アプリを指定されたサーバーにデプロイします。
3. デプロイ用のバージョン、およびすぐに実用に移せるようにミニファイされたバージョンという2つのバージョンのアプリを作成します。この2つのバージョンは出力で切り替えることができます。
4. ChromeブラウザでHP Anywhereシミュレーターを開きます。

アプリを初めて実行する場合は、HP Anywhere IDEによって、プロジェクトファイルのコンパイル、zip形式への圧縮、HP Anywhereサーバーへのデプロイが行われます。

その後でプロジェクトファイルを変更してローカルサーバー上でアプリを実行する場合は、JavaコードのホットデプロイをHP Anywhereで行えるため、サーバーを再起動することなくアプリをリアルタイムで変更できます。

内蔵シミュレーターまたはモバイルデバイスでアプリの表示とデバッグができます (デバイスは、サーバーと同じLANに接続されていて、HP Anywhereクライアントがインストールされている必要があります)。詳細については、「[アプリの実行方法](#)」(18ページ)を参照してください。

ヒント: データベースのクリーンアップを行って、デプロイ済みのアプリと構成ファイルをすべて削除するには、[Remove All Deployed Apps]  をクリックします。

アプリの実行方法

アプリは、ローカル、リモートサーバー上、またはLAN上のデバイス上で実行できます。アプリプロジェクトをどこで実行するかは、`pom-project.properties`ファイルを使用して定義します。

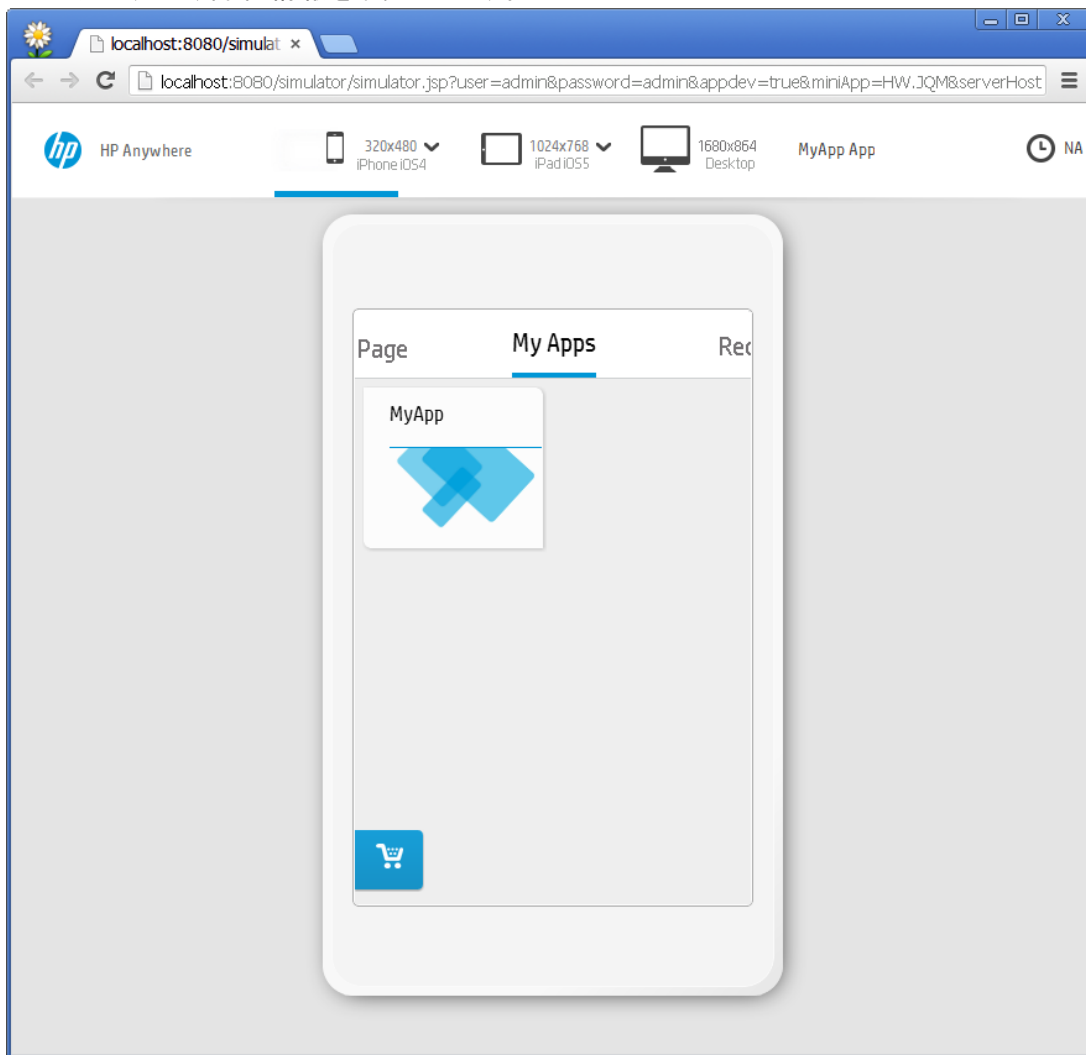
アプリを実行するには、次の手順を実行します。

1. **Senchaユーザーの前提条件 (コンピューターごとに1回実行)**
 - a. [Sencha version 2.0.0 Beta 3 for Windows](#) がコンピューターにインストールされていることを確認します。
 - b. <http://phantomjs.org/> から [phantomjs-1.9.1-windows.zip](#) をダウンロードします。
 - c. .zipファイルから `phantomjs.exe` を解凍し、クリップボードにコピーします。

- d. Senchaの/binフォルダーにある次のようなphantomjs.exeにペースト (上書き) します。C:\Program Files (x86)\SenchaSDKTools-2.0.0-beta3\bin\phantomjs.exe
2. プロジェクトツリーで、**pom-project.properties**をダブルクリックして開きます。次に、ホスト、ポート、および他の設定を確認します。既定では、コンピューターに現在ログオン中のユーザーの資格情報を使用して、ローカルホストが使用されます。
3. プロジェクトツリーで、HP Anywhereプロジェクトフォルダーを右クリックし、**[Run HP Anywhere App]**を選択します。

サーバーが起動し、既定のフォームファクターと解像度を使用してアプリを実行します。アプリの実行中は、[Console] タブでログを確認できます。実行が完了すると、次に示すようにChromeブラウザーにアプリが表示されます。

プロジェクトが他のフォームファクターや解像度をサポートしている場合、シミュレーション対象のフォームファクターの上にある下矢印で選択できます。シミュレーション対象のフォームファクターをダブルクリックすると、詳細情報を確認できます。



アプリのデプロイバージョンとミニファイバージョンを切り替える方法の詳細については、「[アプリをミニファイする方法](#)」(21ページ)を参照してください。

アプリをミニファイする方法

ミニファイは、コードの機能を変更することなく、ソースコードから不要な文字 (ホワイトスペース文字、新規行文字、コメント、ブロックデリミターなど) をすべて削除するプロセスです。

HP Anywhereのアプリプロジェクトファイルは、実用に移す前にミニファイする必要があります。HP Anywhere IDEは、このプロセスを容易に行えるように、アプリプロジェクトを自動的にミニファイする独自のミニファイメカニズムを備えています。

ミニファイプロセスは、フレームワークに依存します。HP Anywhereには、ネイティブ、Enyo、およびjQuery Mobile (JQM) フレームワークを使用して作成されたアプリ用の自動ミニファイサポートが付属しています。(Senchaについては後述します。)また、プロジェクトのルートフォルダーの`minify.bat`ファイルを編集して、独自のミニファイプロセスを追加することもできます。

HP Anywhere IDEでアプリプロジェクトを構築すると、デプロイ用のバージョンに加え、すぐに実用に移せるようにミニファイされたバージョンという2つのバージョンが作成されます。これらのバージョンは、URLの`appdev`フラグを次のように変更し、いつでも切り替えることができます。

アプリのデプロイバージョンとミニファイバージョンを切り替えるには、次の操作を行います。

URLの`appdev`値を次のように設定します。

- `appdev=true` (デプロイモード)
- `appdev=false` (ミニファイモード)

URLの例:

<http://localhost:8080/simulator/simulator.jsp?user=myusername&password=myspassword&appdev=true&miniApp=MR&serverHost=localhost:8080>

ヒント: `&appdev=true`を削除してミニファイバージョンを表示することもできます。

Sencha開発者向けの注記:

1. アプリのミニファイや実行を行う前に、Senchaの`/bin`フォルダーにある`phantomjs.exe`ファイルを置き換える必要があります。(このステップはコンピューターごとに1回実行してください。)
 - a. [Sencha version 2.0.0 Beta 3 for Windows](#) がコンピューターにインストールされていることを確認します。
 - b. <http://phantomjs.org/> から`phantomjs-1.9.1-windows.zip`をダウンロードします。
 - c. `.zip`ファイルから`phantomjs.exe`を解凍し、クリップボードにコピーします。
 - d. Senchaの`/bin`フォルダーにある次のような`phantomjs.exe`にペースト (上書き) します。C:\Program Files (x86)\SenchaSDKTools-2.0.0-beta3\bin\phantomjs.exe
2. Senchaベースのアプリプロジェクトの構築時には、コマンドラインインターフェイス (CLI) で次のコマンドを実行し、ミニファイプロセスを手動で実行する必要があります。

```
cd <your_Sencha-based_project>/src/webapp  
minify sencha
```

アプリの変更時にホットデプロイと同期を有効にする方法

ローカルサーバー上でアプリを最低1回実行すると、ホットデプロイが自動的に有効になります。つまり、プロジェクトを保存するたびに、静的なJavaScriptと構成ファイルが、ローカルHP Anywhereサーバー上のファイルと自動的に同期されます。

Javaコードへのオンデマンドホットデプロイを行うには、次の手順を実行します。

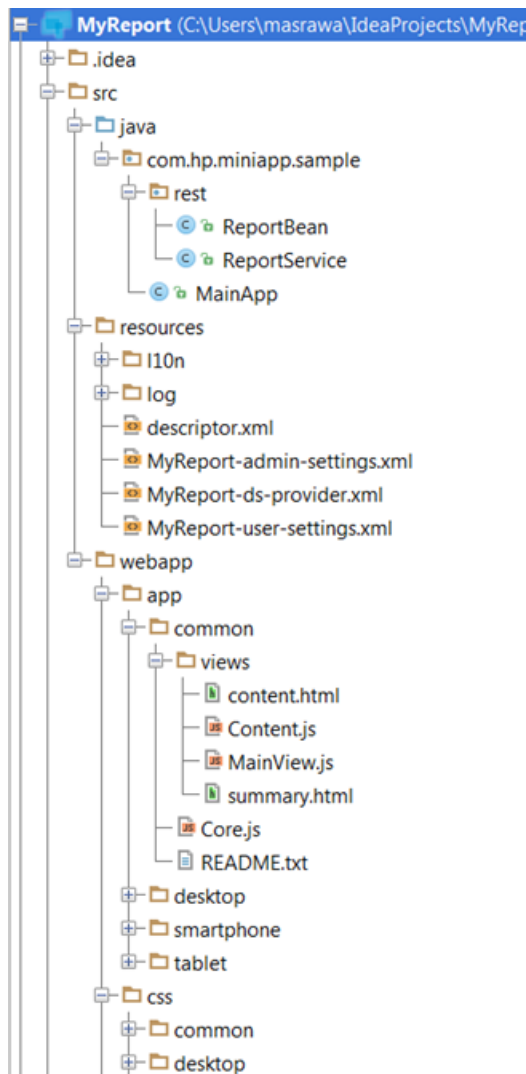
1. 前提条件
 - a. アプリを構築します。詳細については、「[HP Anywhere IDEでアプリプロジェクトを作成する方法](#)」(8ページ)を参照してください。
 - b. ローカルサーバー上でアプリを最低1回実行します。詳細については、「[アプリの実行方法](#)」(18ページ)を参照してください。
 - c. ローカルサーバーが稼働していることを確認します。
2. プロジェクトツリーで、アプリノードを右クリックし、**[Reload HP Anywhere App]**を選択します。Javaコードが変更されたアプリがサーバーにリロードされます。

付録 1: アプリプロジェクトのフォルダー構造

HP Anywhere IDEの各テンプレートは、Sencha、Enyo、JQMまたはネイティブフレームワークなど、特定のフレームワークを使用してアプリを生成します。これらのフレームワークになじみのあるユーザーは、それらを使って作成したプロジェクトと多くの共通点があることに気づくでしょう。

本項では、HP Anywhere IDEのアプリプロジェクトの一部として生成されるアプリ固有のファイルとフォルダーについて概略を説明します。

既定では、アプリプロジェクトのパッケージ構造は次のとおりです。



すべてのプロジェクトには次のフォルダーがあります。

Project name	HP Anywhere App Projectウィザードで指定したアプリプロジェクトの名前です。変更できません。
---------------------	--

src/java	<p>アプリが使用するJavaコードです。すべてのテンプレートには独自のRESTサービスがあります。レポート専用テンプレートは、みな同じサービスを使用してデータを取得します。Javaには2つのクラスがあります。エンティティを取得するJava Bean (ReportBean.java) と、レポートデータを取得するJavaサービス (ReportService.java) です。</p>
src/resources	<p>サーバー側の構成ファイルです (設定、データソース、ローカリゼーションファイルなど)。</p> <p>I10n: アプリが対応している各言語の*.propertiesファイルが格納されているローカリゼーションフォルダーです。</p> <p>log: log4j.propertiesとlogging.propertiesが格納されているログファイルフォルダーです。</p> <p>descriptor.xml: アプリがサポートするフォームファクター (デスクトップ、タブレット、スマートフォン) を指定したファイルです。(実際のリソースファイルは、後述するようにsrc\webapp\<app/css/images>に保存します)。 フォームファクターの追加/削除の詳細については、「アプリプロジェクトのフォームファクターを追加/削除する方法」(14ページ)を参照してください。</p> <p><アプリ名>-admin-settings.xml: アプリ管理者に関する設定を行うファイルです。詳細については、「アプリの構成を定義する方法」(16ページ)の管理者設定に関する項を参照してください。</p> <p><アプリ名>-ds-provider.xml: アプリとバックエンドサーバーとの通信を可能にする、データソース設定のためのファイルです。詳細については、「アプリの構成を定義する方法」(16ページ)のデータソース設定に関する項を参照してください。</p> <p><アプリ名>-user-settings.xml: アプリのユーザーに関する設定を行うファイルです。詳細については、「アプリの構成を定義する方法」(16ページ)のユーザー設定に関する項を参照してください。</p>
Referenced Libraries	<p>アプリで利用できるサードパーティのライブラリです (プロジェクト依存関係)。</p>
JRE System Library	<p>アプリで利用できるJREのライブラリです。</p>
src	<p>フォームファクターごとにフォルダーが分けられているJavaScriptのクライアントコードです。commonフォルダーには、指定したすべてのフォームファクターで利用できるJavaScriptコードが格納されています。</p>

src\webapp	<p>アプリのクライアント側リソースと設定が格納されています。</p> <p>app: アプリのクライアント側 JavaScriptコードです。</p> <p>css: アプリのグラフィカルユーザーインターフェースの外観と書式を定めるカスケードスタイルシートです。commonフォルダーには、すべてのフォームファクターで使用できる既定のフォントとCSSが格納されています。すべてのフォームファクター用のスタイルシートは必要に応じて追加、変更ができます。フォームファクターごとに変更を加えるには、desktop、smartphone、またはtabletフォルダーを展開し、フォームファクター固有のスタイルシートを追加します。</p> <p>images: アプリが使用する画像です。</p> <p>app、css、imagesフォルダーには、それぞれ次のサブフォルダーがあります。</p> <ul style="list-style-type: none"> • common: すべてのフォームファクターに適用される内容です。 • desktop: デスクトップアプリにだけ適用される内容です。 • tablet: タブレットアプリにだけ適用される内容です。 • smartphone: スマートフォンアプリにだけ適用される内容です。 <p>descriptor.xmlファイル(上述のとおり)</p> <p>WEB-INF: J2EEコンテナ用のディレクトリです。標準的なweb.xmlファイルの他に、spring、log4j、およびサードパーティのログに必要なリソースが格納されています。</p> <p>app.js: アプリ用のJavaScriptコードが格納されています。アプリ実行時のエントリーポイントであるopenEntryPoint関数などが含まれます。詳細については、HP Anywhere API仕様を参照してください。</p>
target	<p>アプリ実行時に作成されるターゲットファイルが格納されています。アプリプロジェクトを作成した時点では空です。</p> <p>アプリをリリースできる状態になったら、<アプリプロジェクト>\target\<App name>-cp.zipファイルをHP Anywhere管理者に渡し、カタログにデプロイしてもらいます。</p>
minify.bat	<p>アプリのミニファイにHP Anywhere IDEが使用するバッチファイルです。</p>
package.js	<p>ミニファイプロセスに関連するアプリ情報が記載されたファイルです。</p>
pom.xml	<p>アプリの名前、バージョン、説明など、アプリの一般情報が記載されたファイルです。</p>
pom-project.properties	<p>テンプレートタイプ、ポート、ホスト名、ユーザーの資格情報など、アプリの一般情報が記載されたファイルです。</p>

付録2: テンプレートについて

HP Anywhere IDEには、アプリプロジェクトの土台として使用できるスタータープロジェクトが用意されています。すべてのプロジェクトテンプレートは、あらかじめ必要なプロジェクトフォルダー構造を備えています。また、**My Report**プロジェクトには、REST APIと通信してレポートを生成するためのコードも含まれています。

テンプレート	説明
空のプロジェクト	アプリケーション内容を含まない、プロジェクトフォルダー構造のみのHP Anywhereスタータープロジェクトです。
Hello World	JavaScriptとHTMLコードからなるネイティブフレームワークを使用してアプリを生成するHP Anywhereスタータープロジェクトです。
Hello World (Enyo)	Enyoフレームワークを使用してアプリを生成するHP Anywhereスタータープロジェクトです。タブレットやスマートフォン用のプロジェクトです。
Hello World (Jqm)	jQuery Mobileフレームワークを使用してアプリを生成するHP Anywhereスタータープロジェクトです。タブレットやスマートフォン用のプロジェクトです。
Hello World (Sencha)	Senchaフレームワークを使用してアプリを生成するHP Anywhereスタータープロジェクトです。タブレットやスマートフォン用のプロジェクトです。
Hello World (AngularJS)	AngularJSフレームワークを使用してアプリを生成するHP Anywhereスタータープロジェクトです。タブレットやスマートフォン用のプロジェクトです。
My Report	JavaScriptとHTMLコードからなるネイティブフレームワークを使用してレポート表示用アプリを生成するHP Anywhereスタータープロジェクトです。
My Report (Enyo)	Enyoフレームワークを使用してレポート表示用アプリを生成するHP Anywhereスタータープロジェクトです。タブレットやスマートフォン用のプロジェクトです。
My Report (Jqm)	jQuery Mobileフレームワークを使用してレポート表示用アプリを生成するHP Anywhereスタータープロジェクトです。タブレットやスマートフォン用のプロジェクトです。
My Report (Sencha)	Sencha Touchフレームワークを使用してレポート表示用アプリを生成するHP Anywhereスタータープロジェクトです。タブレットやスマートフォン用のプロジェクトです。
My Report (AngularJS)	AngularJSフレームワークを使用してレポート表示用アプリを生成するHP Anywhereスタータープロジェクトです。タブレットやスマートフォン用のプロジェクトです。

空のプロジェクトテンプレート

アプリ固有の内容を含まない、標準的なアプリフォルダーとファイルからなるHP Anywhereスタータープロジェクトです。フォルダーとファイルの詳細については、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)を参照してください。

ネイティブアプリ用テンプレート

Hello Worldは、HP Anywhereの[マイアプリ] ページにHP Anywhereアイコンを表示するネイティブHTML 5アプリです。このプロジェクトには、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)で示した標準的なフォルダーとファイルの他に、次の内容が含まれます。

- **src\webapp\app\common**: クライアント側 JavaScript、CSS、画像ファイルを格納したフォルダーです。commonフォルダー内のコードはすべてのフォームファクターで使用できます。
- **src\webapp\app\common\views**: このフォルダーには次のファイルが格納されています。
 - **content.html**: このアプリのHTMLテンプレートです。
 - **MainView.js**: JavaScriptファイルとHTMLファイルの内容を読み込む初期ビューです。
 - **summary.html**: デスクトップフォームファクター用の[マイアプリ] ページのHello Worldアプリカードに表示される内容です。通常は、アプリ名、アイコン、オプションの内容 (テキスト形式の説明または概要など)、アイテム一覧、ダッシュボードなどが表示されます。

My Reportは、レポートを表示するためのネイティブHTML5アプリです。このプロジェクトには、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)で示した標準的なフォルダーとファイルの他に、次の内容が含まれます。

- **src\webapp\app\common**: レポート用のクライアント側 JavaScript、CSS、画像ファイルを格納したフォルダーです。commonフォルダー内のコードはすべてのフォームファクターで使用できます。
- **src\webapp\app\common\views**: このフォルダーには次のファイルが格納されています。
 - **content.html**: このアプリのHTMLテンプレートです。
 - **Content.js**: レポート用のデータをサーバーから取得するためのファイルです。
 - **MainView.js**: JavaScriptファイルとHTMLファイルの内容を読み込む初期ビューです。
 - **summary.html**: デスクトップフォームファクター用の[マイアプリ] ページのMy Report Projectアプリカードに表示される内容です。通常は、アプリ名、アイコン、オプションの内容 (テキスト形式の説明または概要など)、アイテム一覧、ダッシュボードなどが表示されます。

Enyoベースのアプリ用テンプレート

各アプリは、Enyoベースの標準アプリ構造に準拠するように設計されています。HP Anywhere IDEでは、Enyoローダーを使用してビューを初期化し、ディスプレイへの出力のレンダリングを行います。

Hello World (Enyo)は、HP Anywhereの[マイアプリ] ページにHP Anywhereアイコンを表示するEnyoベースのアプリです。このプロジェクトには、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)で示した標準的なフォルダーとファイルの他に、次の内容が含まれます。

src\webapp\app.js: Hello Worldビューを初期化し、ディスプレイへの出力のレンダリングを行います。アプリの背後にある論理を含むコントローラーです。

My Report (Enyo)は、レポートを表示するためのEnyoベースのアプリです。このプロジェクトには、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)で示した標準的なフォルダーとファイルの他に、次の内容が含まれます。

- **src\java\com\hpl<プロジェクト名>\...\rest:** このフォルダーには、レポート一覧を取得するための、以下のRESTサービスクラスが格納されています。
 - **ReportBean.java**クラス: レポートエンティティです。
 - **ReportService.java**クラス: 組み込みのモックデータを含むRESTの実装です。
- **src\webapp\app\common:** レポートアプリのクライアント側Enyoリソースを格納したフォルダーです。次の内容が含まれます。
 - **enyo:** Enyoライブラリのコードです。
 - **report:** レポート一覧の表示方法を定義したReport.jsおよびReportItem.jsクラスです。
- **src\webapp\css\common:** Enyo固有のレポート用スタイルシートを格納したフォルダーです。
- **src\webapp\app.js:** レポートビューを初期化し、ディスプレイへの出力のレンダリングを行います。

Senchaベースのアプリ用テンプレート

各テンプレートの構造は、Model View Controlなど、Sencha Touchベースのアプリ作成のベストプラクティスに準拠するように設計されています。

Hello World (Sencha)は、HP Anywhereの[マイアプリ] ページにHP Anywhereアイコンを表示するSencha Touchベースのアプリです。このプロジェクトには、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)で示した標準的なフォルダーとファイルの他に、次の内容が含まれます。

- **src\webapp\app\common:** アプリのクライアント側Sencha Touchリソースを格納したフォルダーです。controller、data、mode、store、viewなどのフォルダーがあります。commonフォルダー内のリソースはすべてのフォームファクターで使用できます。

My Report (Sencha)は、レポートを表示するためのSencha Touchベースのアプリです。このプロジェクトには、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)で示した標準的なフォルダーとファイルの他に、次の内容が含まれます。

- `src\java\com\hp\<プロジェクト名>\...\rest`: このフォルダーには、レポート一覧を取得するための、以下のRESTサービスクラスが格納されています。
 - `ReportBean.java`クラス: レポートエンティティです。
 - `ReportService.java`クラス: 組み込みのモックデータを含むRESTの実装です。
- `src\webapp\app\common`: レポートアプリのクライアント側 Sencha Touchリソースを格納したフォルダーです。controller、data、mode、store、viewなどのフォルダーがあります。commonフォルダー内のリソースはすべてのフォームファクターで使用できます。

jQuery Mobileベースのアプリ用テンプレート

Hello World (JQM)は、HP Anywhereの[マイアプリ] ページにHP Anywhereアイコンを表示するjQuery Mobileベースのアプリです。このプロジェクトには、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)で示した標準的なフォルダーとファイルの他に、次の内容が含まれます。

- `src\webapp\app\common`: アプリのクライアント側 JQMリソースを格納したフォルダーです。conf、controller、およびviewなどのフォルダーがあります。commonフォルダー内のリソースはすべてのフォームファクターで使用できます。
- `src\webapp\libs\jqmobile`: このフォルダーには、JQM画像、CSS、およびアプリ用の機能が格納されます。
- `src\webapp\libs\jquery`: このフォルダーには、アプリ用のjQueryファイルが格納されます。
- `src\webapp\libs\requirejs`: このフォルダーには、アプリの読み込みに必要なリソースが格納されます。

My Report (JQM)は、レポートを表示するためのjQuery Mobileベースのアプリです。このプロジェクトには、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)で示した標準的なフォルダーとファイルの他に、次の内容が含まれます。

- `src\webapp\app\common`: アプリのクライアント側 JQMリソースを格納したフォルダーです。conf、controller、models、およびviewなどのフォルダーがあります。commonフォルダー内のリソースはすべてのフォームファクターで使用できます。
- `src\webapp\libs\jqmobile`: このフォルダーには、JQM画像、CSS、およびアプリ用の機能が格納されます。
- `src\webapp\libs\jquery`: このフォルダーには、アプリ用のjQueryファイルが格納されます。

- `src\webapp\libs\requirejs`: このフォルダーには、アプリの読み込みに必要なリソースが格納されます。

AngularJSベースのアプリ用テンプレート

各テンプレートの構造は、AngularJSベースのアプリ作成のベストプラクティスに準拠するように設計されています。この構造には、「[付録1: アプリプロジェクトのフォルダー構造](#)」(24ページ)に示す標準フォルダーとファイルのみが含まれています。固有のファイルやフォルダーはありません。

付録3: トラブルシューティング

本項では、発生する可能性のある問題への対処方法を説明します。

Mavenプロキシを設定する方法

プロキシ経由でインターネットに接続した場合は、Mavenプロキシを構成してMavenリポジトリに接続する必要があります。

Mavenプロキシを構成するには、次の手順を実行します。

%btoa_home%/apache-maven-3.0.4/conf/settings.xmlの<proxies>セクションを更新し、プロキシを追加します。

例:

```
<proxy>
  <id>MYNETPROXY</id>
  <active>true</active>
  <protocol>http</protocol>
  <host>myproxyhost.mydomain.com</host>
  <port>8080</port>
  <nonProxyHosts>*.devlab.ad</nonProxyHosts>
</proxy>
```

getUserNameメソッド (UserInfoService API) でNULLが返る場合の対処方法

1. パッケージエクスプローラーで、<アプリのメインノード>/src/webapp/WEB-INF/Web.xmlに移動し、ファイルをダブルクリックして開きます。
2. <filter>セクションを以下に置き換えます。

```
<filter>
  <filter-name>DiamondTenantFilter</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy
</filter-class>
</filter>
```



```
<filter-mapping>
  <filter-name>DiamondTenantFilter</filter-name>
  <url-pattern>/rest/*</url-pattern>
  <url-pattern>/services/*</url-pattern>
</filter-mapping>

<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy
</filter-class>
</filter>
<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<filter>
  <filter-name>DiamondSecurityFilter</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy
</filter-class>
</filter>
<filter-mapping>
  <filter-name>DiamondSecurityFilter</filter-name>
  <url-pattern>/rest/*</url-pattern>
  <url-pattern>/services/*</url-pattern>
</filter-mapping>
```

