

HP Anywhere IDE

Windows

Software Version: 10.02

Developer Guide (IntelliJ IDEA)

Document Release Date: August 2013

Software Release Date: August 2013



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

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

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

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

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

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

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

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

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

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

To find more information about access levels, go to:

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

Contents

Developer Guide (IntelliJ IDEA)	1
Contents	5
Why should I create my app using HP Anywhere IDE?	6
How to build and run your app using HP Anywhere IDE—End-to-end walkthrough	7
How do I create an app project in HP Anywhere IDE?	8
How do I add a third-party library to my app?	13
Supported Third-Party Libraries	13
How do I add or remove form factors in my app project?	14
How do I define the configuration for my app?	16
User Settings	16
Administrator Settings	16
Data Source Settings	17
How do I run and view my app?	18
How do I run my app?	18
How do I minify my app?	20
How do I enable hot deployment and synchronization when my app changes?	21
How do I remove apps?	22
Appendix 1: App Project Folder Structure	23
Appendix 2: Template Descriptions	26
Blank Project Template	26
Templates for Native Apps	27
Templates for Enyo-based Apps	27
Templates for Sencha Touch-based Apps	28
Templates for jQuery Mobile-based Apps	29
Appendix 3: Troubleshooting	30
How do I set up a Maven proxy?	30
What should I do if the getUsername method (UserInfoService API) returns NULL?	30

Chapter 1

Why should I create my app using HP Anywhere IDE?

HP Anywhere provides an IntelliJ IDEA-based IDE that enables you to create, build, run, and debug your own apps.

This IDE provides a complete development environment that covers the entire lifecycle of your app and enables you to create, debug, maintain your app projects. HP Anywhere IDE includes:

- A **wizard** for creating app projects
- A **built-in simulator** that you can use to run and debug your apps in any form factor (desktop, tablet, or smartphone) or screen resolution
- Deployment capabilities that publish your apps for you, including **hot deployment** for locally deployed apps and **automatic minification** so that your apps are always production-ready.
- A complete, pre-configured, database-free, and debug-ready **HP Anywhere server** that is customized and optimized for the development environment.

How to build and run your app using HP Anywhere IDE—End-to-end walkthrough

The following steps describe how to build and run an app using HP Anywhere IDE.

1. Make sure that HP Anywhere IDE is installed on your computer.
2. Use the HP Anywhere App Project wizard to create the package and files for your app project. This provides you with a ready-to-run app that includes all of the necessary files, classes, and methods. You can select from the any of the available templates to get you started. For details, see ["How do I create an app project in HP Anywhere IDE?" on the facing page](#), ["Appendix 1: App Project Folder Structure" on page 23](#), and ["Appendix 2: Template Descriptions" on page 26](#).
3. Add your code.
4. Define your app configuration settings. For details, see ["How do I define the configuration for my app?" on page 16](#).
5. Run and debug your app by running it on the built-in HP Anywhere local server or a remote server. For details, see ["How do I run my app?" on page 18](#).
6. When your app is ready, you can give the app content pack .zip file to the HP Anywhere administrator so that he or she can deploy it on the HP Anywhere server and enable it in the catalog. This .zip file has the following naming convention: **<App name>-cp.zip**, and is located in the **<App project>\target** folder.

How do I create an app project in HP Anywhere IDE?

You use the HP Anywhere App project wizard to create a new HP Anywhere IDE project containing the files and classes for your app.

When you create an app using HP Anywhere IDE, an app content pack is generated as a .zip file in the **<App project>\target** folder.

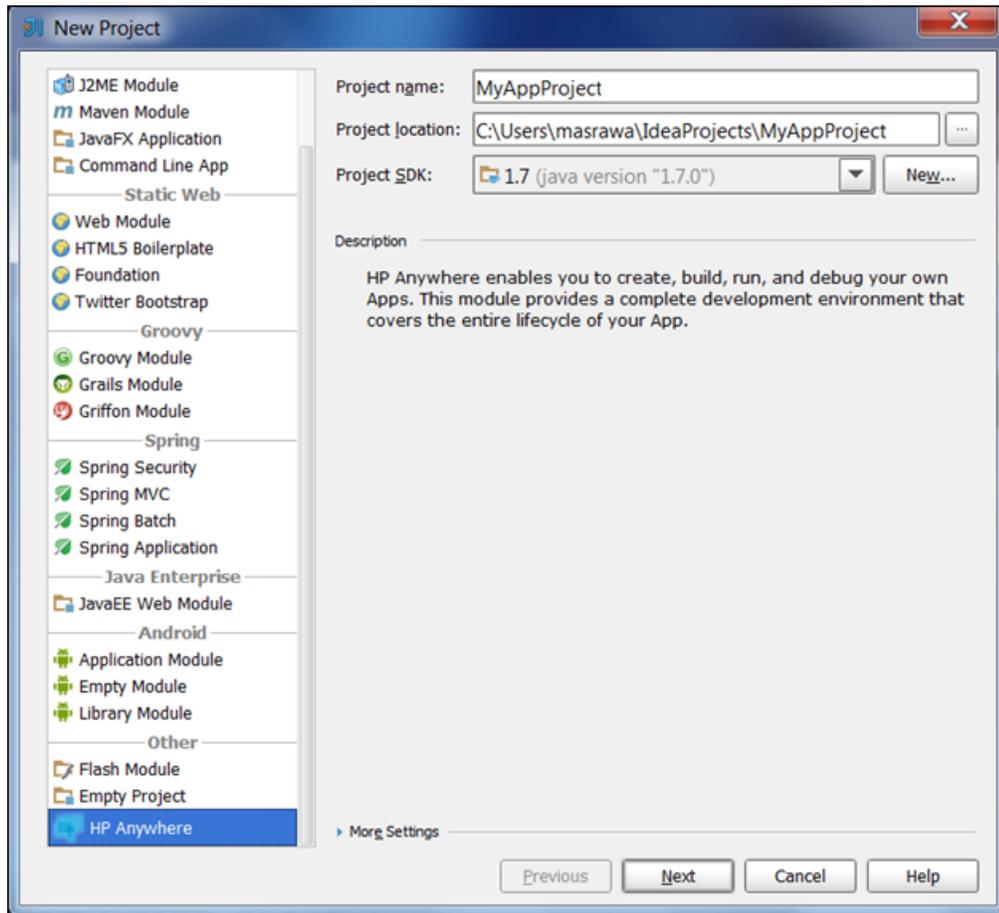
This .zip file has the following naming convention: **<App name>-cp.zip**. You need to give this to the HP Anywhere administrator when the app is ready so that he or she can deploy it on the HP Anywhere server and enable it in the catalog.

Note: If you connect to the Internet via a proxy, you may need to configure a Maven proxy the first time you create an app. For details, see "[How do I set up a Maven proxy?](#)" on page 30.

To create an HP Anywhere app project in HP Anywhere IDE:

1. Do one of the following:
 - In the Quick Start pane of the IntelliJ IDEA Welcome window, select **Create New Project**.
 - In IntelliJ IDEA, select **File > New Project**.

The New Project wizard opens.



Note: The fields shown in the following example may be different depending on your IntelliJ IDEA version.

2. Do the following:
 - In the left pane, select HP Anywhere.
 - In the right pane, fill in the **Project name** and any other fields, as needed.

3. Click **Next**. The HP Anywhere IDE New Project screen opens.

New Project

Project Template

Type: Hello World Project

Description: An HP Anywhere starter project that generates an App using the native framework with JavaScript and HTML code.

Form Factor

- Smartphone
- Tablet
- Desktop

Project Details

App Name: MyAppProject

Group ID: com.hp.my.miniapp

Source package: com.hp.miniapp.sample

App version: 1.0.0

App icon: ...

Previous Finish Cancel Help

4. In the Project Template area, select the type of template from the **Type** drop-down list. Each of the templates generates a starter project to which you can add your own code. Several types are available:
 - **Blank project.** An HP Anywhere starter project that contains the project folder structure without application content.
 - **Hello World projects.** HP Anywhere starter projects that generate apps using the specified framework (such as native, Enyo, JQM, or Sencha). These projects are targeted for use on tablets and smartphones.
 - **My Report projects.** HP Anywhere starter projects that generate apps that can display a list of reports. These projects use the specified framework (such as native, Enyo, JQM, or Sencha). These projects are targeted for use on tablets and smartphones.

The **Description** box provides details for the selected template.

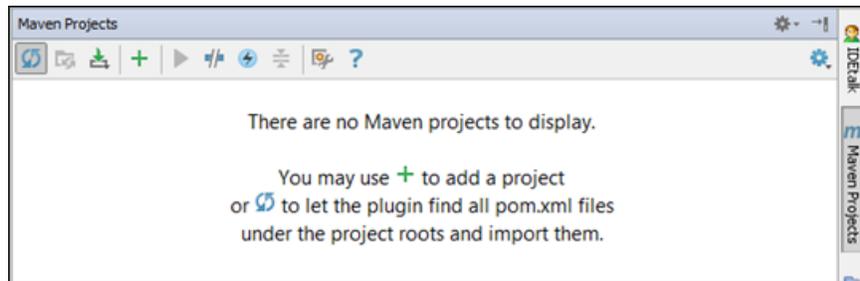
For a detailed description of each of these templates, see "[Appendix 2: Template Descriptions](#)" on page 26.

5. In the Form Factor area, select one or more form factors. The following form factors are available:
 - **Smartphone**
 - **Tablet**
 - **Desktop**

Note: After you create a project, you can manually add form factors that you did not select while creating the project. For details, see "[How do I add or remove form factors in my app project?](#)" on page 14.

6. In the Project Details area, define the following using standard Maven naming conventions:
 - **App Name.** The name of the app, for example, *HelloWorld*. (Use only alphanumeric characters. Special characters and spaces are not allowed.)
 - **Group ID.** A reverse domain name, for example, *com.mycompany.mygroupname*. The group ID is used in the generated POM.XML project file.
 - **Source package.** A valid Java package name, for example, *com.my.mypackage*. The package name represents the folder structure that is generated in the workspace.
 - **App version.** The version string for your app, for example, *1.0.0*.

- **App icon.** A PNG file that represents your app. The icon is displayed in the app on the My Apps page in HP Anywhere.
7. Click **Finish**. HP Anywhere generates the app project. While the project is being created, you can view the log in the Console tab. (The first time you create a project, this may take some time.)
 8. Import the project as follows:
 - a. Open the Maven Projects pane.



- b. Click the **Reimport All Maven Projects** button .
9. After the project is imported, you can:
 - View the project folder structure. For details, see ["Appendix 1: App Project Folder Structure" on page 23](#)
 - Add code to the app project.
 - Run the app. For details, see ["How do I run my app?" on page 18](#).

How do I add a third-party library to my app?

You can add a third-party dependency to the `pom.xml` file. After you modify this file, you must refresh the IDE by clicking the **Reimport All Maven Projects** button in the Maven Projects pane.

Supported Third-Party Libraries

Your app can use the following third-parties libraries:

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

How do I add or remove form factors in my app project?

When you create your app project, you specify the form factors (desktop/tablet/smartphone) on which your app can run.

If you do not select all form factors when you create the project, you can add them later by adding folders for the required form factors to your app's package. Similarly, you can remove form factors from your app by removing the relevant project folders.

To add form factors for your app:

1. In the HP Anywhere IDE Package Explorer, expand the main app folder.
2. Copy/paste the folders for an existing form factor (desktop, tablet, or smartphone) in each of the following locations:
 - **src > webapp > app**
 - **src > webapp > css**
 - **src > webapp > images**
3. Rename each of the new folders to the required form factor type (desktop, tablet, or smartphone).
4. Add your client code to the new form factor folders.
5. In **src/resources > descriptor.xml**, create a section for each form factor by copying the following section and replacing the *<existing form factor type>* with the *<new form factor type>*. For example, copy the following section and replace **DESKTOP** with **SMARTPHONE**.

```
<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>
```

To remove form factors:

In the HP Anywhere IDE Package Explorer:

1. Expand the main app folder.
2. Remove the relevant section from **src/resources > descriptor.xml**. For example, to remove the desktop form factor, delete the following:

```
<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. (Optional) In each of the following locations, delete the folders for each form factor (desktop, tablet, or smartphone) that you want to remove:

- **src > webapp > app**
- **src > webapp > css**
- **src > webapp > images**

How do I define the configuration for my app?

After you create your app, you can configure its settings, which are located in the **src/resources** folder.

User Settings

You define these settings for your app's users.

To modify the user settings:

1. In the Package Explorer, navigate to: */<app main node>/src/resources/<app name>-user-settings.xml* and double-click the file.
2. In the *<app name>-user-settings.xml* tab, select the **Design** tab.
3. Right-click **context** and select **Add Child > setting**.
4. Expand the setting node you just added and enter values for the following:
 - a. **name**. The ID for this setting.
 - b. **namekey**. A setting used for localization.
 - c. **integer**. The default setting type. You can modify this by right-clicking and selecting **Replace With > <value type>**.
Possible values:
 - boolean**
 - enumeration**
 - float**
 - password**
 - string**
 - xml**

Administrator Settings

You define these settings for your app's administrators.

To modify administrator settings:

1. In the Package Explorer, navigate to: */<app main node>/src/resources/<app name>-admin-settings.xml* and double-click the file.
2. In the *<app name>-admin-settings.xml* tab, select the **Design** tab.
3. Right-click **context** and select **Add Child > setting**.

4. Expand the setting node you just added and enter values for the following:
 - a. **name**. The ID for this setting.
 - b. **settingType**. Make sure that **global** is selected.
 - c. **displayInUI**. Specifies whether the app is visible in the HP Anywhere Administrator Console.
Possible values:
true
false
 - d. **sectionKey**. The name of the group area that is displayed in the Settings tab of the Administrator Console (for the app that is selected in the List of Categories column).
 - e. **integer**. The default setting type. You can modify this by right-clicking and selecting **Replace With > <value type>**.
Possible values:
boolean
enumeration
float
password
string
xml

Data Source Settings

You define the settings that enable the app to communicate with the backend server.

1. In the Package Explorer, navigate to: `/<app main node>/src/resources/<app name>-ds-provider.xml` and double-click the file.
2. In the `<app name>-admin-settings.xml` tab, select the **Source** tab.
3. Modify the configuration as needed.

How do I run and view my app?

You can run your app in **debug mode** or **run mode** on either a **local** or **remote** server. HP Anywhere IDE comes with a built-in local server that enables you to take advantage of hot deployment.

When you run your app, HP Anywhere performs the following:

1. Starts the HP Anywhere server if it is not already up and running.
2. Deploys your app on the specified server, if needed.
3. Creates two versions of your app—one for development use and one that is minified and ready for production. You can toggle between the two in the output.
4. Opens HP Anywhere Simulator in a Chrome browser.

When you run your app project for the first time, HP Anywhere IDE compiles the project files, zips them, and deploys them on the HP Anywhere server.

Later, when you modify your project files and run your app on the local server, HP Anywhere provides **hot deployment** of your Java code, so you can modify your app in realtime without restarting the server.

You can view and debug your app on the built-in simulator or on your own mobile device (if the device is in the same LAN and the HP Anywhere client is installed on it). For details, see "[How do I run my app?](#)" below.

How do I run my app?

You can run your app locally, on a remote server, or on a device on your LAN. You define where to run the app project using the **pom-project.properties** file.

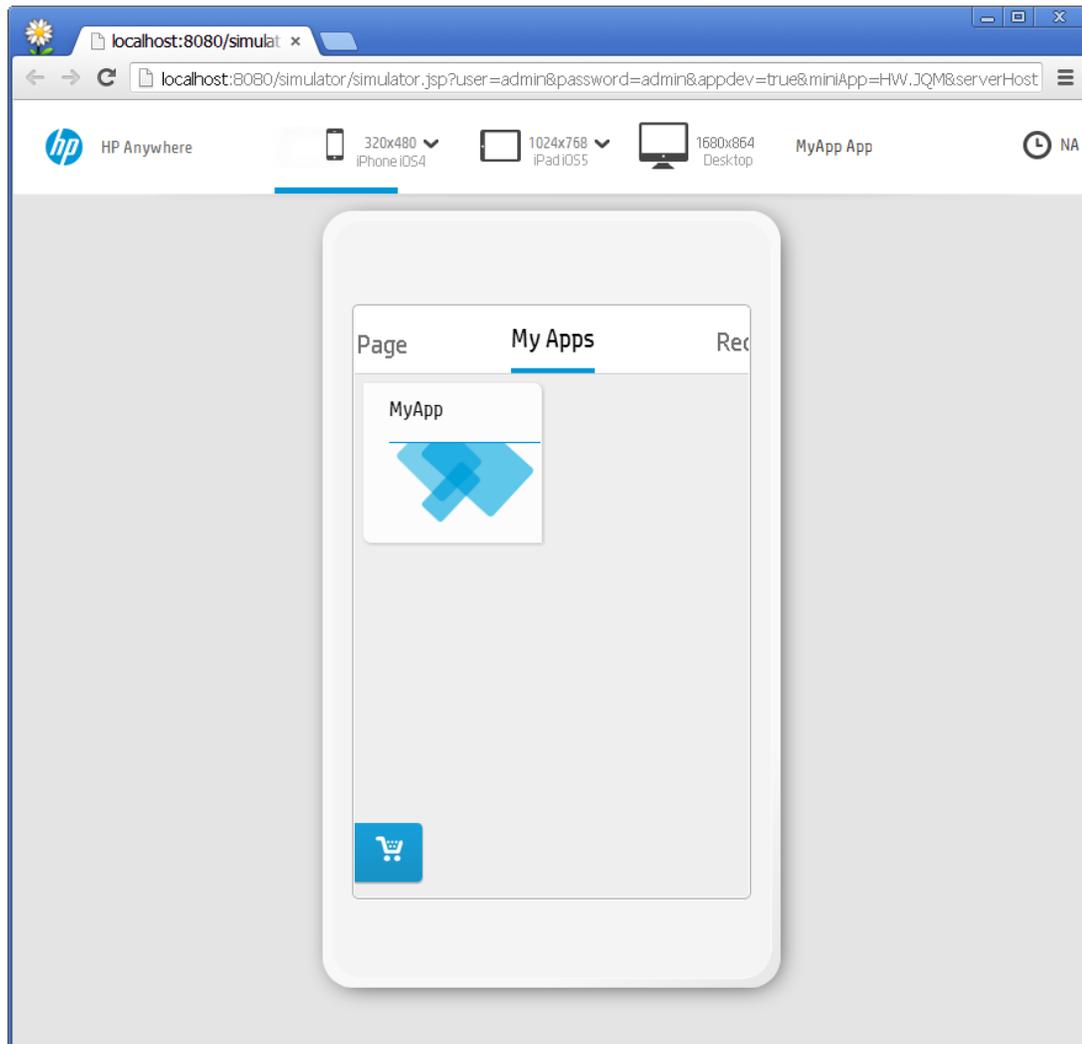
To run your app:

1. In the project tree, double-click **pom-project.properties** to open the file. Then, verify the host, port, and other settings. By default, localhost is used with the credentials for the user that is currently logged on to the computer.
2. In the project tree, right-click the HP Anywhere project folder and select **Run HP Anywhere App**.

The server starts and runs your app using the default form factor and resolution. While the app runs, you can view the log in the Console tab. When the run finishes, the app opens in a Chrome browser, as shown in the following example.

If your project contains code that supports other form factors and resolutions, you can select them from the drop-down arrows above the simulated form factor. You can also double-click the

simulated form factor to view more details.



For details on how to switch between the development version and the minified version of the app, see ["How do I minify my app?"](#) on the facing page.

How do I minify my app?

Minification is the process of removing all unnecessary characters from source code (such as white space characters, new line characters, comments, and block delimiters) without changing the code functionality.

Your HP Anywhere app project files must be minified before going live for production. To ease this process, HP Anywhere IDE provides an out-of-the-box minification mechanism that automatically minifies your app projects for you.

The minification process is framework-dependent. HP Anywhere comes with automatic minification support for apps created using the native, Enyo, and jQuery Mobile (JQM) frameworks. (For Sencha, see note below.) You can also add your own minification process by editing the **minify.bat** file in the root folder of your project.

When HP Anywhere IDE builds an app project, it creates two versions—one for development, and one that is minified and ready for production. You can switch between these versions at any time by modifying the **appdev** flag in the URL, as described below.

To switch between the development version and the minified version of the app:

Set the value of **appdev** in the URL as follows:

- **appdev=true** for development mode
- **appdev=false** for minified mode

Example URL:

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

Tip: You can also delete the **&appdev=true** part to show the minified version.

Note for Sencha Developers: When you build your Sencha-based app project, you must run the minification process manually, as follows:

1. Make sure that the Sencha SDK Tools package is installed (<http://www.sencha.com/products/sdk-tools/download>).
2. Run the following commands using a command line interface (CLI):

```
cd <your_Sencha-based_project>/src/webapp  
  
sencha create jsb -a index.html -p app.jsb3 -v  
  
sencha build -p app.jsb3 -d . -v
```

How do I enable hot deployment and synchronization when my app changes?

After you run your app at least once on the local server, hot deployment is automatically enabled. This means that every time you save your project, your static JavaScript and configuration files are automatically synchronized with the files on your local HP Anywhere server.

To perform on-demand hot deployment on your Java code:

1. Prerequisites:
 - a. Build your app. For details, see ["How do I create an app project in HP Anywhere IDE?" on page 8.](#)
 - b. Run your app at least once on the local server. For details, see ["How do I run my app?" on page 18.](#)
 - c. Make sure that the local server is up and running.
2. In the project tree, right-click the app node and select **Reload HP Anywhere App**. The server reloads the app with the modified Java code.

How do I remove apps?

You cannot uninstall an app from HP Anywhere, but you can clean the database as needed to remove all of the apps.

To remove all apps from the HP Anywhere server:

1. Stop the HP Anywhere server by running: `%BTOA_HOME%\stop-server.bat`

2. Delete: `%BTOA_HOME%\conf\H2DB\hpa-db-h2.h2.db`

3. Delete: `%BTOA_HOME%\conf\H2DB\hpa-db-h2.lock.db`

4. Edit the following `%BTOA_HOME%\conf\database.properties`:

- set `database.URL` to:

```
database.url=jdbc:h2:file:conf/H2DB/hpa-db-h2;  
MVCC=TRUE;INIT=RUNSCRIPT FROM 'conf/H2DB/h2-create-tables.sql'
```

- Set `hibernate.hbm2ddl.auto` to:

```
hibernate.hbm2ddl.auto=update
```

5. Start the HP Anywhere server by running: `%BTOA_HOME%\start-server.bat`

6. Edit the following `%BTOA_HOME%\conf\database.properties`:

- set `database.URL` to:

```
database.url=jdbc:h2:file:conf/H2DB/hpa-db-h2'
```

- Set `hibernate.hbm2ddl.auto` to:

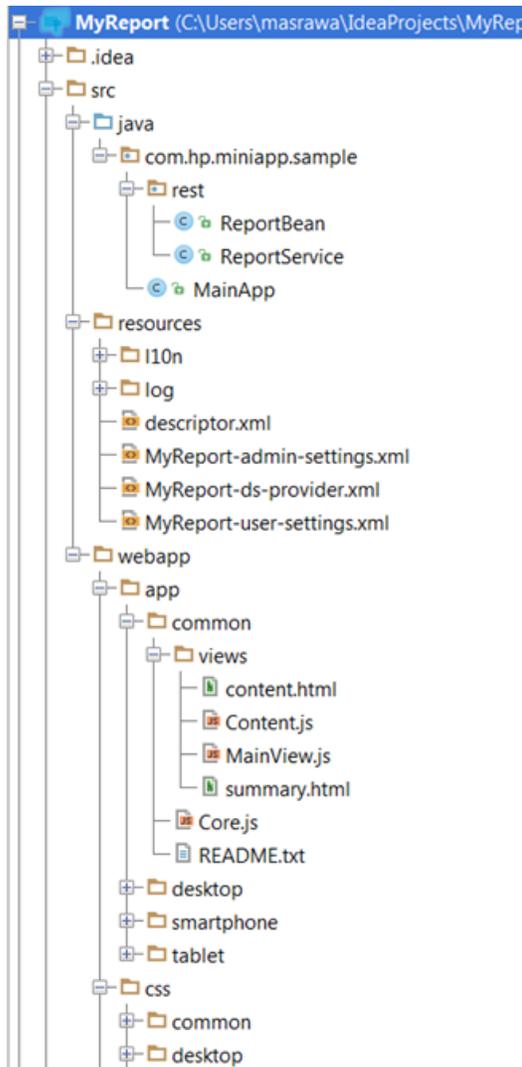
```
hibernate.hbm2ddl.auto=none
```

Appendix 1: App Project Folder Structure

Each of the templates provided with HP Anywhere IDE generates an app using a specific framework, such as Sencha, Enyo, JQM, or a native framework. If you are familiar with these frameworks, you will see many similarities to other projects that you may have already created.

This section provides an overview of the unique files and folders that are generated as part of an HP Anywhere IDE app project.

By default, each app project has the following package structure:



Every project includes the following folders:

Project name	The non-modifiable name of the app project, as defined in the HP Anywhere app Project wizard.
---------------------	---

src/java	<p>The Java code used by the app. Every template has its own REST services. The report-specific templates all use the same service to retrieve data. Java has two classes: one is a Java bean (ReportBean.java), which retrieves the entity; the other is a Java service (ReportService.java) that retrieves the report data.</p>
src/resources	<p>The server-side configuration files (for example, settings, data source, and localization files).</p> <p>l10n. Localization folder that includes *.properties files for each language in which the app can be displayed.</p> <p>log. Log files folder that includes log4j.properties and logging.properties.</p> <p>descriptor.xml. Specifies all of the form factors (desktop, tablet, and/or smartphone) that are supported for the app. (You store the actual resource files under src\webapp\<app>\css\images, described below.) For details on adding or removing form factors in your project, see "How do I add or remove form factors in my app project?" on page 14.</p> <p><app_name>-admin-settings.xml. Specifies the settings for the app's administrators. For details, see the administrator settings section in "How do I define the configuration for my app?" on page 16.</p> <p><app_name>-ds-provider.xml. Specifies the data source settings that enable the app to communicate with the backend server. For details, see the data source settings section in "How do I define the configuration for my app?" on page 16.</p> <p><app_name>-user-settings.xml. Specifies the settings for the app's users. For details, see the user settings section in "How do I define the configuration for my app?" on page 16.</p>
Referenced Libraries	<p>Third-party libraries that are available for the app (project dependencies).</p>
JRE System Library	<p>The JRE libraries that are available for the app.</p>
src	<p>The JavaScript client code, which is divided into folders per form factor. The common folders contain JavaScript code that is used by all defined form factors.</p>

src\webapp	<p>The client-side resources and settings for the app.</p> <p>app. The JavaScript client code for the app.</p> <p>css. The Cascading Style Sheet that defines the look and formatting of the app's graphical user interface. The common folder includes the default fonts and CSS for all form factors. You can add or modify style-sheet code for all form factors, as needed. To modify a specific form factor, expand the desktop, smartphone, or tablet folder and add form-factor-specific style-sheet code, as needed.</p> <p>images. The referenced images used by your app.</p> <p>Each of the app, css, and images folders contains the following subfolders:</p> <ul style="list-style-type: none"> • common. Content that applies to all form factors. • desktop. Content that applies only to the desktop app. • tablet. Content that applies only to the tablet app. • smartphone. Content that applies only to the smartphone app. <p>The descriptor.xml file (described above)</p> <p>WEB-INF. The directory for the J2EE container, which contains the standard web.xml file and resources for spring, log4j, and third-party logging.</p> <p>app.js. Contains the JavaScript code for your app, including the openEntryPoint function that specifies the client location that opens when a user opens your app. For details, see the HP Anywhere API specifications.</p>
target	<p>The target files that are created when your app runs. This folder is empty when your app project is created.</p> <p>When the app is ready for release, you can give the <App project>\target\<App name>-cp.zip file to your HP Anywhere administrator for deployment to the catalog.</p>
minify.bat	A batch file that HP Anywhere IDE uses to minify the app.
package.js	Contains app information that is relevant for the minification process.
pom.xml	Contains general app data, such as the app name, version, and description.
pom-project.properties	Contains general app data, such as the template type, port, hostname, and user credentials.

Appendix 2: Template Descriptions

HP Anywhere IDE provides starter projects that you can use as a basis for your own app projects. All of the project templates include the required project folder structure. The **My Report** projects also include code so that you can use to create an app that communicates with a REST API and generates a list of reports.

Template	Description
Blank Project	An HP Anywhere starter project that contains the project folder structure without application content.
Hello World	An HP Anywhere starter project that generates an app using the native framework with JavaScript and HTML code.
Hello World (Enyo)	An HP Anywhere starter project that generates an app using the Enyo framework. This project is targeted for use on tablets and smartphones.
Hello World (Jqm)	An HP Anywhere starter project that generates an app using the jQuery Mobile framework. This project is targeted for use on tablets and smartphones.
Hello World (Sencha)	An HP Anywhere starter project that generates an app using the Sencha Touch framework. This project is targeted for use on tablets and smartphones.
My Report	An HP Anywhere starter project that uses the native framework with JavaScript and HTML code to generate an app that can display a list of reports.
My Report (Enyo)	An HP Anywhere starter project that uses the Enyo framework to generate an app that can display a list of reports. This project is targeted for use on tablets and smartphones.
My Report (Jqm)	An HP Anywhere starter project that uses the jQuery Mobile framework to generate an app that can display a list of reports. This project is targeted for use on tablets and smartphones.
My Report (Sencha)	An HP Anywhere starter project that uses the Sencha Touch framework to generate an app that can display a list of reports. This project is targeted for use on tablets and smartphones.

Blank Project Template

An HP Anywhere starter project that contains the standard app folders and files, but does not contain any app-specific content. For folder and file details, see ["Appendix 1: App Project Folder Structure" on page 23](#).

Templates for Native Apps

Hello World is a native HTML 5 app that displays the HP Anywhere icon in the HP Anywhere My Apps page. In addition to the standard folders and files described in ["Appendix 1: App Project Folder Structure" on page 23](#), this project includes:

- **src\webapp\app\common**: This folder contains the client-side JavaScript, CCS, and image files. The code is stored in the common folder, making it available for all form factors.
- **src\webapp\app\common\views**: This folder contains:
 - **content.html**: The HTML template for this app.
 - **MainView.js**: The initial view that loads the content of the JavaScript and HTML files.
 - **summary.html**: The content displayed in the Hello World app card on the My Apps page for the desktop form factor. For example, a card usually contains the app name, an icon, and optional content, such as a textual description or summary, a list of items, or a dashboard.

My Report is a native HTML 5 app that displays a report list. In addition to the standard folders and files described in ["Appendix 1: App Project Folder Structure" on page 23](#), this project includes:

- **src\webapp\app\common**: This folder contains the client-side JavaScript, CCS, and image files for the report. The code is stored in the common folder, making it available for all form factors.
- **src\webapp\app\common\views**: This folder contains:
 - **content.html**: The HTML template for this app.
 - **Content.js**: Used to retrieve the data from the server for the report.
 - **MainView.js**: The initial view that loads the content of the JavaScript and HTML files.
 - **summary.html**: The content displayed in the My Report Project app card on the My Apps page for the desktop form factor. For example, a card usually contains the app name, an icon, and optional content, such as a textual description or summary, a list of items, or a dashboard.

Templates for Enyo-based Apps

Each template is designed to conform to the standard app structure of an Enyo-based app. HP Anywhere IDE uses the Enyo loader to initialize the view and render the output to the display.

Hello World (Enyo) is an Enyo-based app that displays the HP Anywhere icon in the HP Anywhere My Apps page. In addition to the standard folders and files described in ["Appendix 1: App Project Folder Structure" on page 23](#), this project includes:

src\webapp\app.js: Initializes the Hello World view and renders the output to the display. This is the controller that contains the logic behind the app.

My Report (Enyo) .is an Enyo-based app that displays a report list. In addition to the standard folders and files described in "[Appendix 1: App Project Folder Structure](#)" on page 23, this project includes:

- **src\java\com\hp\<project_name>\...\rest**: This folder contains the following REST service classes for retrieving the report list:
 - **ReportBean.java** class: The report entity.
 - **ReportService.java** class: The REST implementation, which contains embedded, mock data.
- **src\webapp\app\common**: This folder contains the Enyo resources needed for the client side of the report app, including:
 - **enyo**: The Enyo library code.
 - **report**: The Report.js and ReportItem.js classes, which define how the report list is displayed.
- **src\webapp\css\common**: This folder contains the Enyo-specific app styles for the report.
- **src\webapp\app.js**: Initializes the report view and renders the output to the display.

Templates for Sencha Touch-based Apps

The structure of each template is designed to conform to the best practices for creating a Sencha Touch-based app, including the Model View Control.

Hello World (Sencha) is a Sencha Touch-based app that displays the HP Anywhere icon in the HP Anywhere My Apps page. In addition to the standard folders and files described in "[Appendix 1: App Project Folder Structure](#)" on page 23, this project includes:

- **src\webapp\app\common**: This folder contains the Sencha Touch resources needed for the client side of the app, including controller, data, mode, store, and view folders. These resources are stored in the common folder, making them available for all form factors.

My Report (Sencha) .is a Sencha Touch-based app that displays a report list. In addition to the standard folders and files described in "[Appendix 1: App Project Folder Structure](#)" on page 23, this project includes:

- **src\java\com\hp\<project_name>\...\rest**: This folder contains the following REST service classes for retrieving the report list:

- **ReportBean.java** class: The report entity.
- **ReportService.java** class: The REST implementation, which contains embedded mock data.
- **src\webapp\app\common**: This folder contains the Sencha Touch resources needed for the client side of the report app, including controller, data, mode, store, and view folders. These resources are stored in the common folder, making them available for all form factors.

Templates for jQuery Mobile-based Apps

Hello World (JQM) is a jQuery Mobile-based app that displays the HP Anywhere icon in the HP Anywhere My Apps page. In addition to the standard folders and files described in "[Appendix 1: App Project Folder Structure](#)" on page 23, this project includes:

- **src\webapp\app\common**: This folder contains the JQM resources needed for the client side of the app, including conf, controller, and view folders. These resources are stored in the common folder, making them available for all form factors.
- **src\webapp\libs\jqmobile**: This folder contains the JQM images, CSS, and functions for the app.
- **src\webapp\libs\jquery**: This folder contains the jQuery files for the app.
- **src\webapp\libs\requirejs**: This folder contains the resources needed to load the app.

My Report (JQM) is a jQuery Mobile-based app that displays a report list. In addition to the standard folders and files described in "[Appendix 1: App Project Folder Structure](#)" on page 23, this project includes:

- **src\webapp\app\common**: This folder contains the JQM resources needed for the client side of the app, including conf, controller, models, and view folders. These resources are stored in the common folder, making them available for all form factors.
- **src\webapp\libs\jqmobile**: This folder contains the JQM images, CSS, and functions for the app.
- **src\webapp\libs\jquery**: This folder contains the jQuery files for the app.
- **src\webapp\libs\requirejs**: This folder contains the resources needed to load the app.

Appendix 3: Troubleshooting

This section describes how to handle issues that you may encounter:

How do I set up a Maven proxy?

If you connect to the Internet via a proxy, you may need to configure a Maven proxy to connect to the Maven repository.

To configure a Maven proxy:

Update the `<proxies>` section in `%btoa_home%/apache-maven-3.0.4/conf/settings.xml` and add a proxy.

For example:

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

What should I do if the getUsername method (UserInfoService API) returns NULL?

1. In the Package Explorer, navigate to: `<app main node>/src/webapp/WEB-INF/Web.xml` and double-click the file to open it.
2. Replace the `<filter>` section with the following:

```
<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>
```

