HP Anywhere IDE

Windows

Software Version: 10.02

Developer Guide (Eclipse)

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

Why should I create my app using HP Anywhere IDE?

HP Anywhere provides an Eclipse-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 37, and "Appendix 2: Template Descriptions" on page 40.

3. Add your code.

4. Define your app configuration settings. For details, see "How do I define the configuration for my app?" on page 15.

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 the local server?" on page 17 and "How do I debug my app code?" on page 30.

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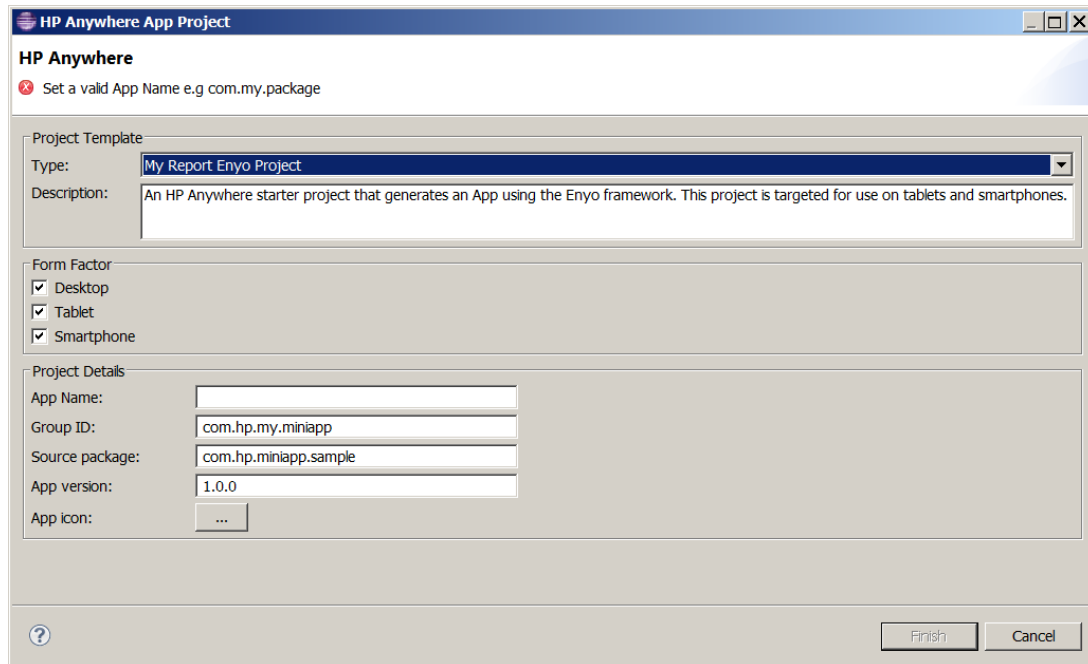
When you create an app using HP Anywhere IDE, an app content pack is generated as a .zip file in the **<App project>\target** folder.

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 44.

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

1. Select **File** > **New** > **HP Anywhere App Project**. The HP Anywhere App Project wizard opens.

2. 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 40.

3. In the Form Factor area, select one or more form factors. The following form factors are available:

- **Desktop**

- **Tablet**

- **Smartphone**

   > **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 13.

4. 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 unique, 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.

5. 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.) After the project is created, you can:

    - View the project folder structure. For details, see "Appendix 1: App Project Folder Structure" on page 37

    - Add code to the app project.

    - Run the app. For details, see "How do I run my app on the local server?" on page 17.

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

If you add a third-party dependency to the **pom.xml** file, you must also add it to the HP Anywhere project, as an Eclipse dependency. This enables code compilation in the IDE.

1. Open a command line (CLI).

2. Change the directory to the root directory of your app.

3. Run the following command: **%btoa_home%\apache-maven-3.0.4\bin\mvn eclipse:eclipse**

4. In the project tree, right-click the app node and select **Refresh** to refresh the app project in HP Anywhere IDE.

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 on a remote server?" on page 20.

> **Note:** For details on running legacy apps that were created using an earlier version of HP Anywhere, see "How do I run an app that was created using an earlier version of HP Anywhere IDE?" on page 45.
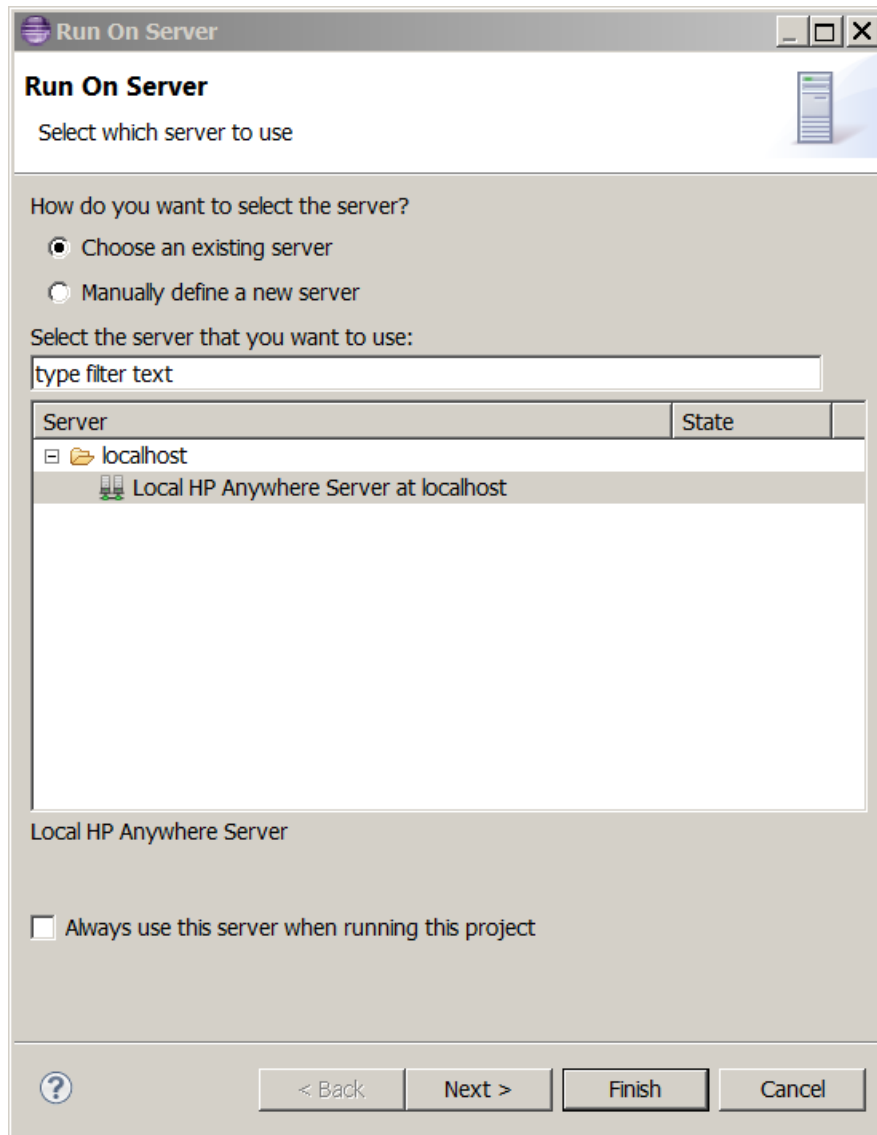
How do I run my app on the local server?

You can run your app on the built-in local server. This enables you to take advantage of hot deployment.

To run your app in run mode on the local server:

1. Do one of the following:

   - In the project explorer, right-click the HP Anywhere project folder and select **Run HP Anywhere App**.

   - Click .

The Run On Server dialog box opens.

2. Select **Local HP Anywhere Server at localhost** and click **Finish** 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.

3. 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 page 32.

How do I run my app on a remote server?

You can run your app on a **remote** server. For general details, see "How do I run and view my app?" on page 17.

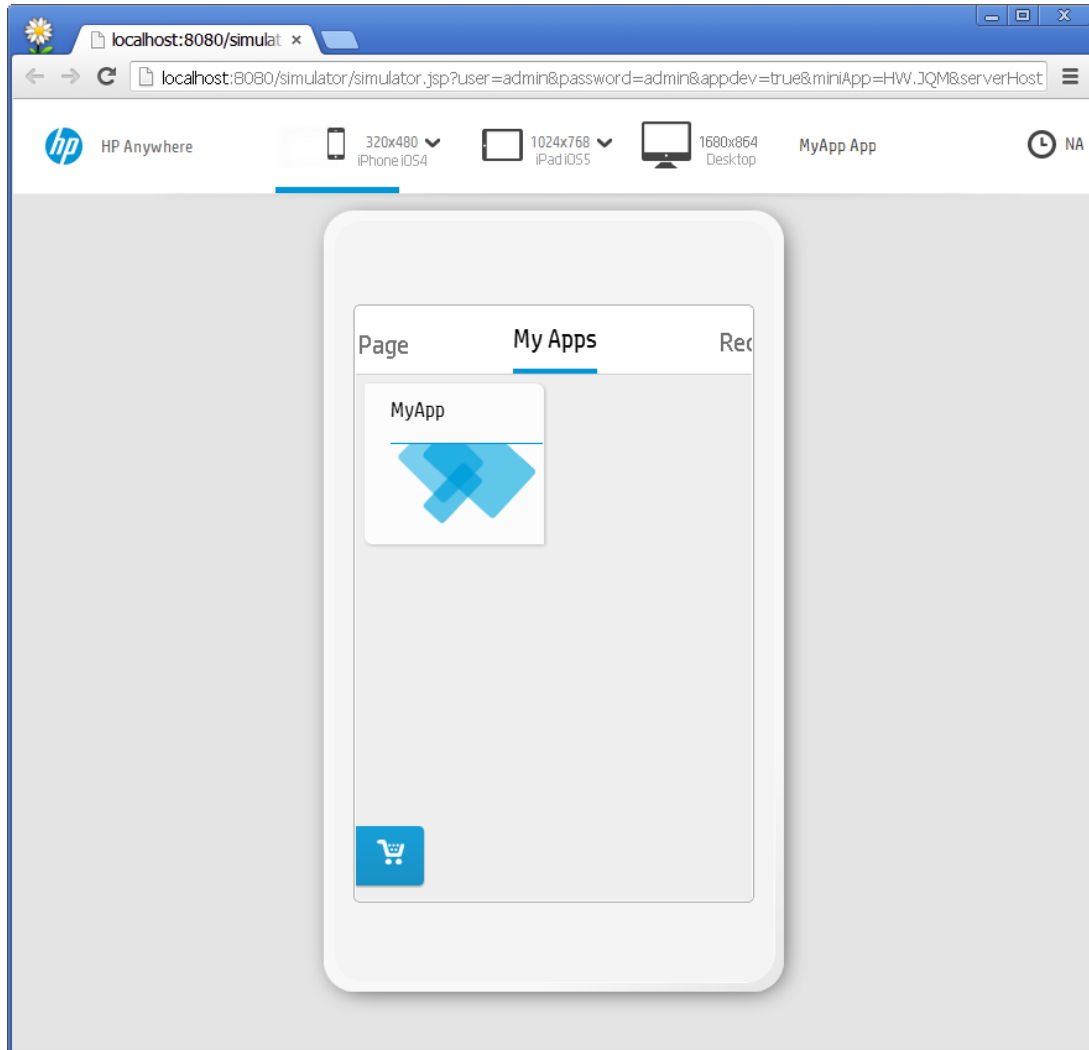**To run your app in run mode on a remote server:**

1. Do one of the following:
   - In the project explorer, right-click the HP Anywhere project folder and select **Run HP Anywhere App**.

   - Click .

   - In the package explorer or project explorer, right-click the HP Anywhere project folder and select **Run As > Run On Server**.

The Run On Server dialog box opens.
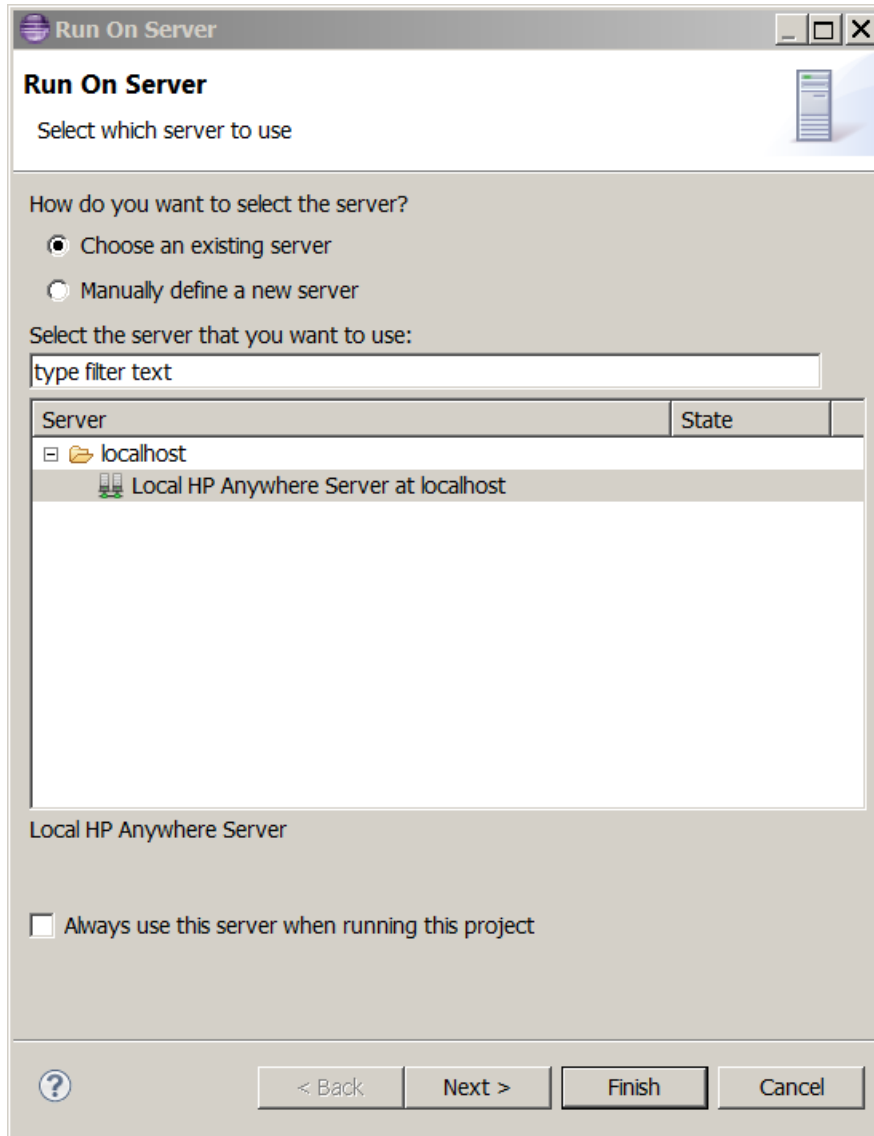
2. Select **Manually define a new server**. The dialog box enables you to specify a different server.
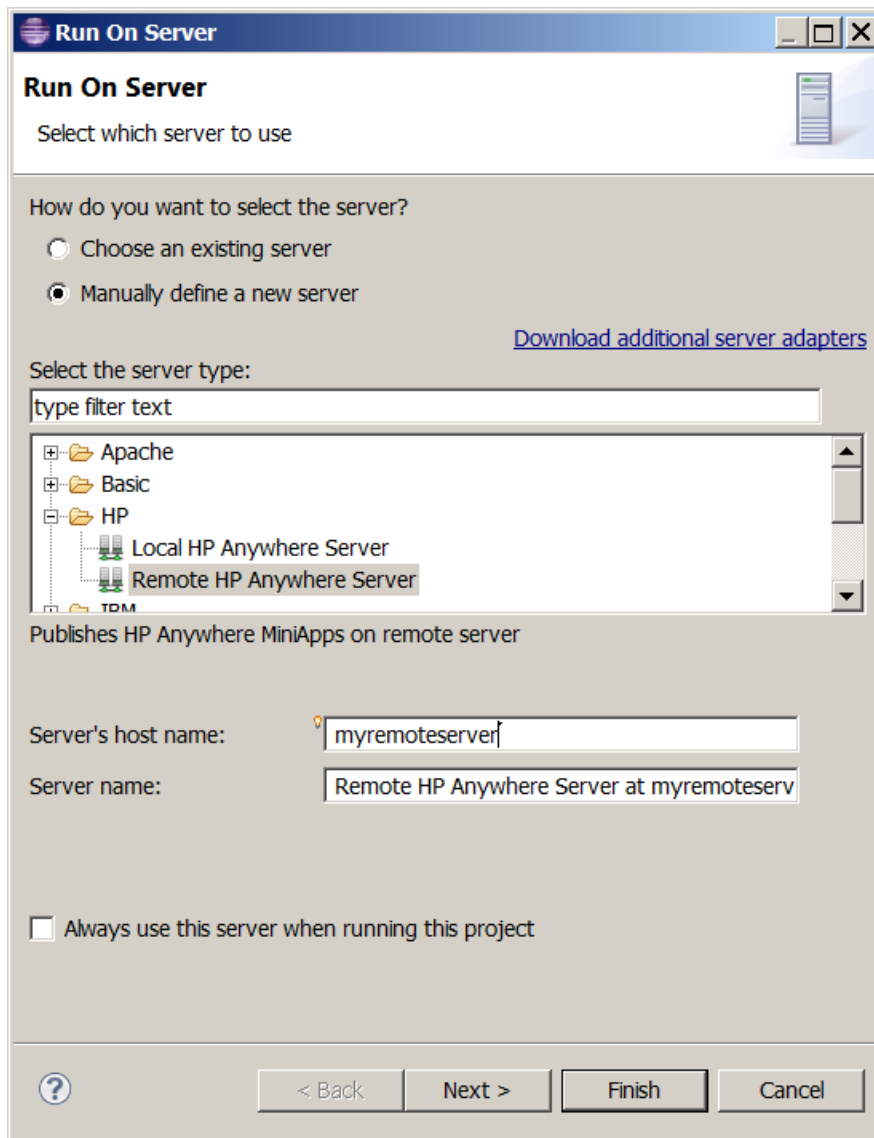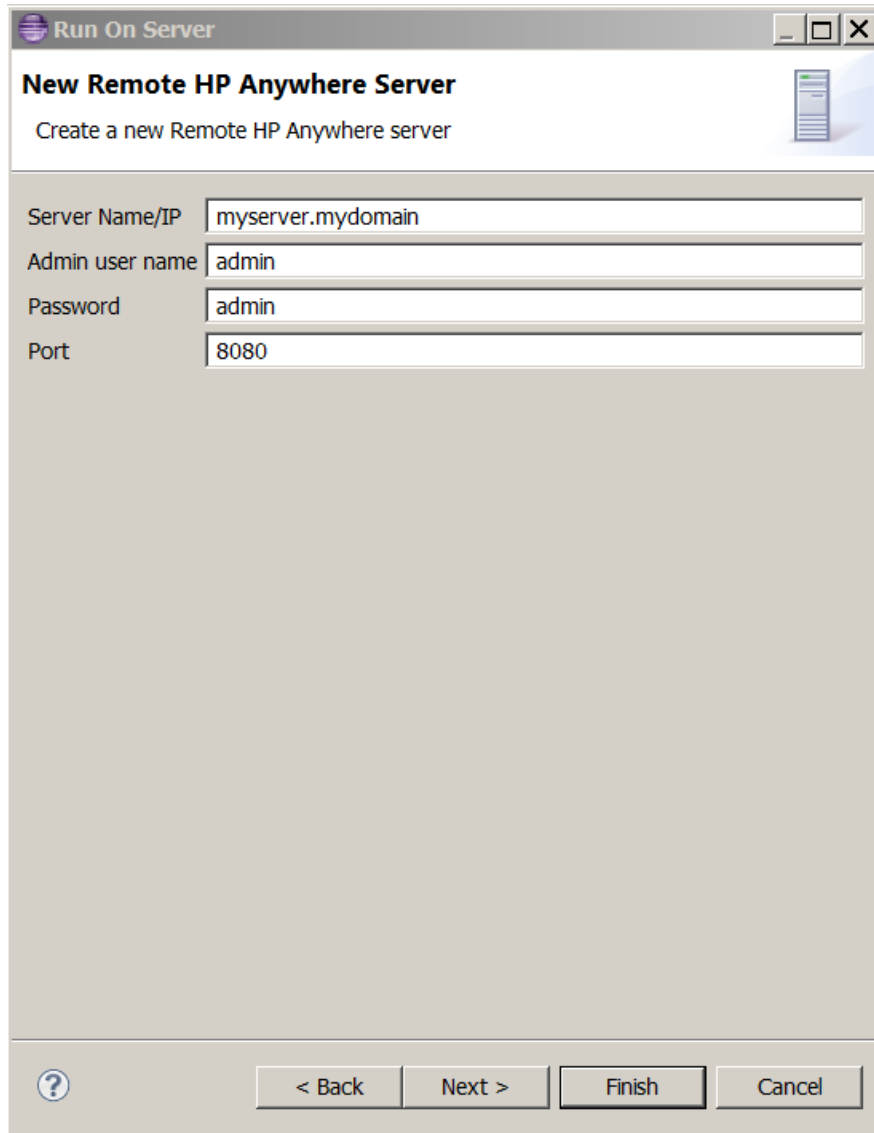
- Select **Remote HP Anywhere Server**.

- In the **Server's host name** box, specify the remote server's name or IP address.

- (Optional) In the **Server name** box, modify the server name.

- (Optional) Select the **Always use this server when running this project** check box if you always want to run your app only on this server.

3. Click **Next**. The JRE preferences page opens.
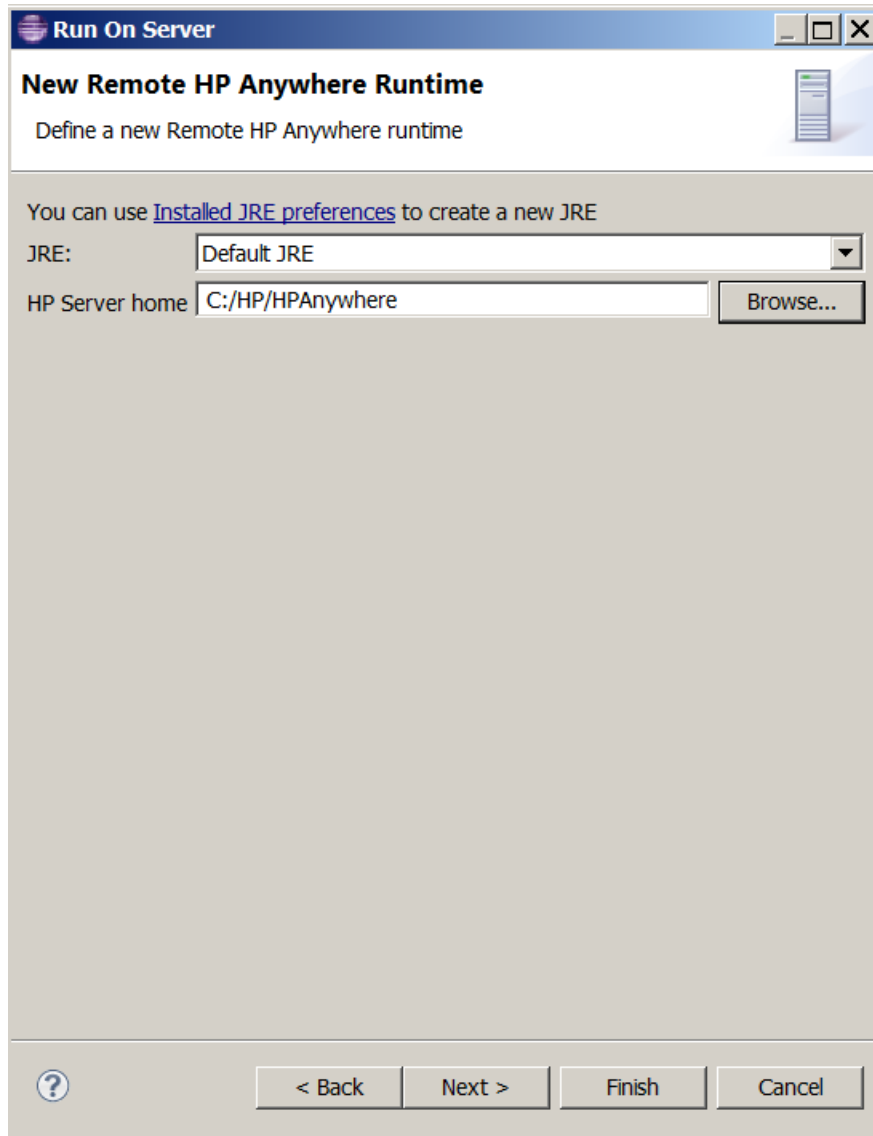
Set the **HP Server home** path to <installation folder>, for example:
C:/HP/HPAnywhere

4. Click **Next**. The login details page opens. Modify the default values, as needed.

5. Click **Next**. The Add and Remove page opens.
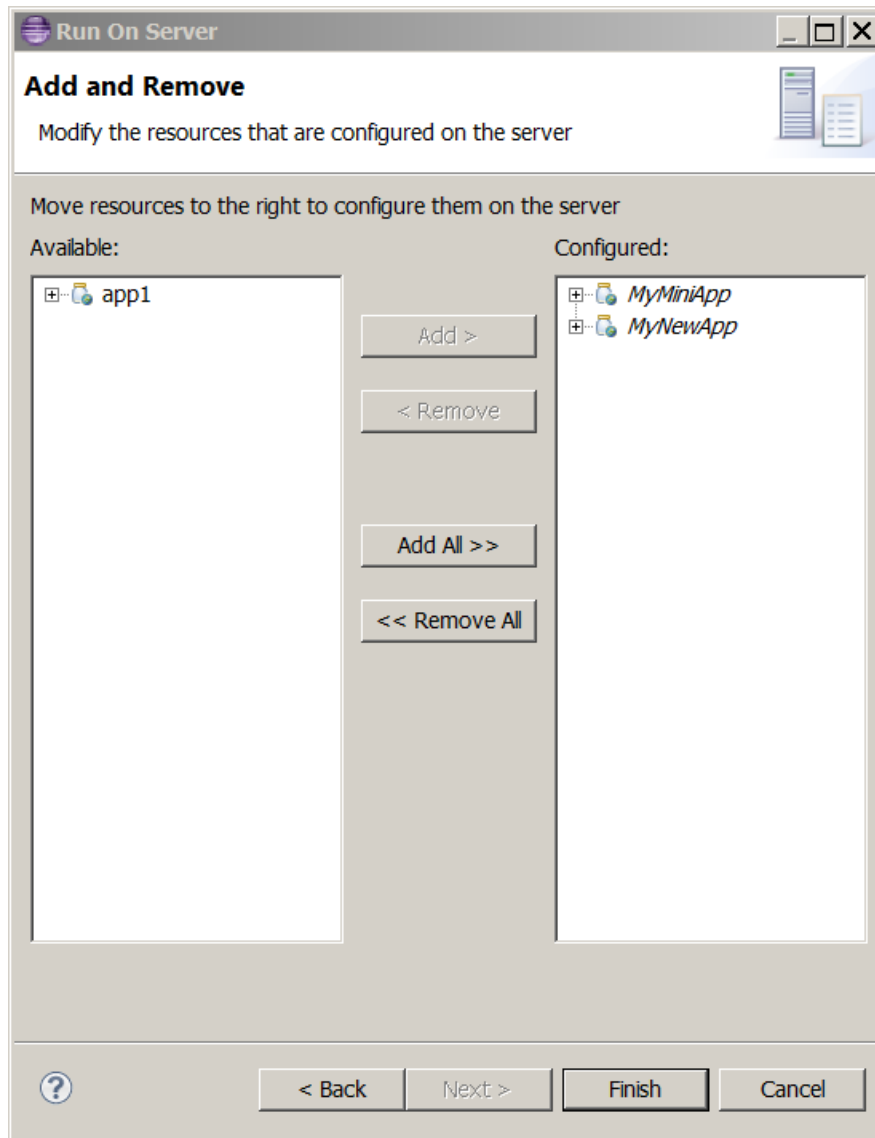
(Optional) You can add or remove projects to run on this server, as needed.

6. Click **Finish**.The server starts and runs your app on the specified remote server using the default form factor and resolution. While the app runs, you can view the log in the Console tab.

Note: If Chrome generates a profile-related error message, close the message and continue to the next step.

7. 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.
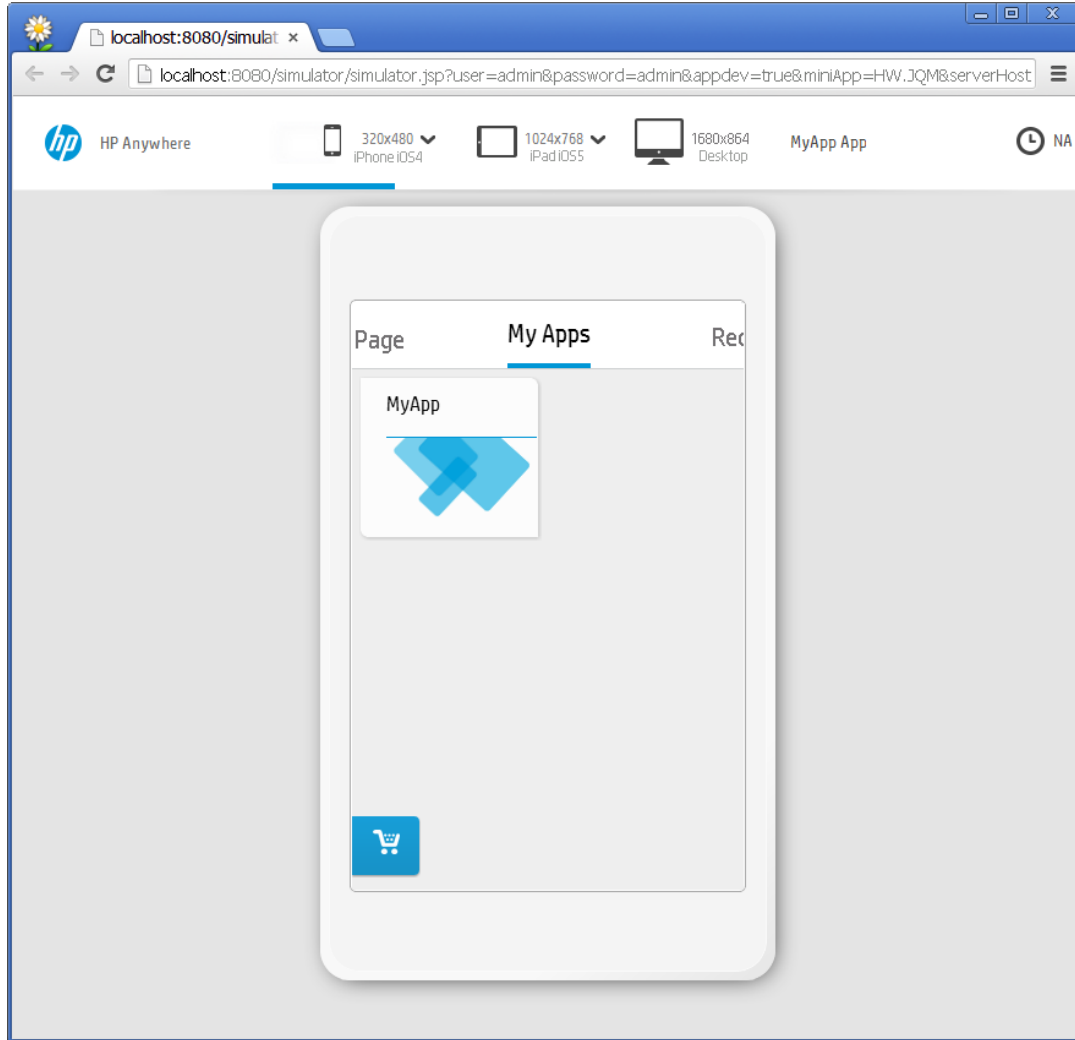
Note: After you run an app on a remote server, you may need to manually enable it in the catalog.

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 page 32.

How do I run my app on a device?

You can run your app from any supported smartphone, tablet, or desktop/laptop computer.

To run your app on a device:

1. Install HP Anywhere on the device. For details, see *HP Anywhere Installation and Configuration Guide*.

2. Connect the device to the Internet via the same LAN as the HP Anywhere server.

3. From the device, log onto the HP Anywhere server on which your app is installed.

How do I view my app on HP Anywhere Simulator?

You can use the HP Anywhere Simulator to view your app in any form factor.

Note: You must run the app project at least once before you can open it in the HP Anywhere Simulator, and the server must already be running.

To view your app project on HP Anywhere Simulator:

1. Do one of the following:
   - In the project explorer, right-click the project and select **View HP Anywhere App Simulator**.

   - Click .

The app opens in a Chrome browser on the last server on which you ran this project.
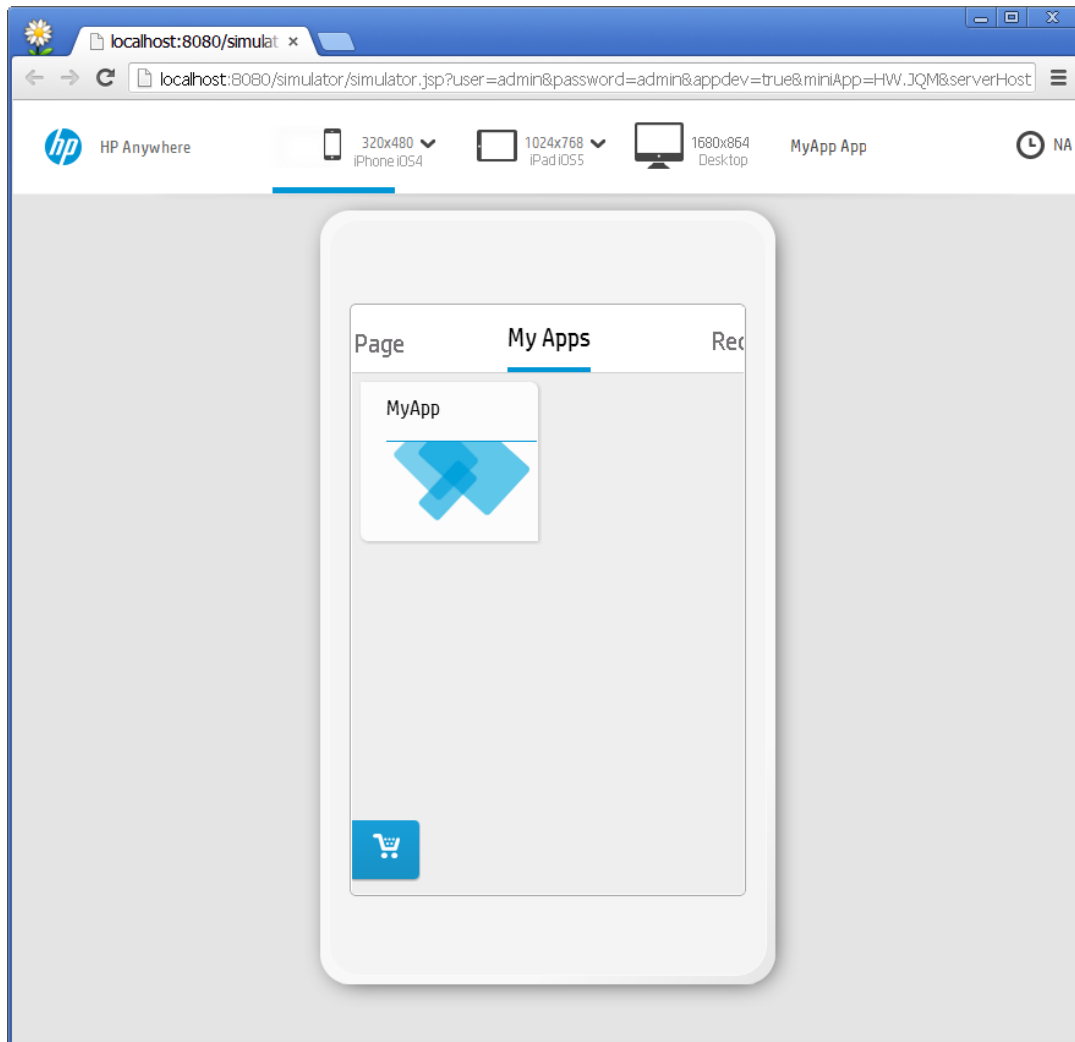
Note: The HP Anywhere Simulator bar at the top of the window times out and hides automatically to enable you to use the full real estate. To restore the bar, click the blue line under the bar.

2. In the HP Anywhere Simulator bar, click the arrow adjacent to the required form factor and select the required screen resolution. (**Desktop** opens in a new tab or window.)

Note: Only the form factors you specified when creating the project are enabled in the HP Anywhere Simulator bar.

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 page 32.

How do I debug my app code?
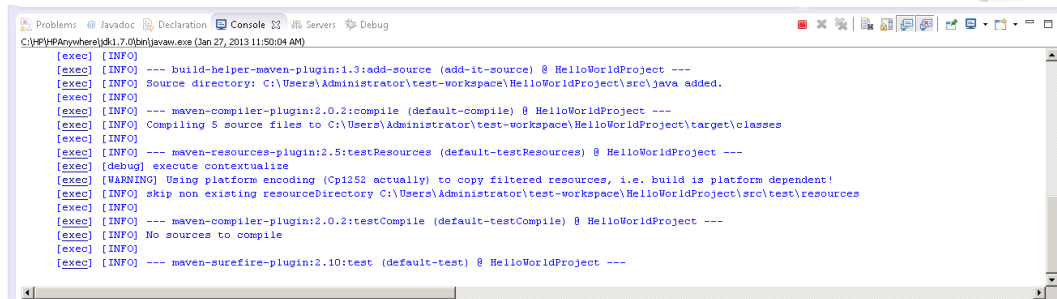
You can run your app in debug mode.

To debug your app:

1. In HP Anywhere IDE, right-click the app project folder and select **Debug As > Debug On Server**. The Debug On Server dialog box opens.
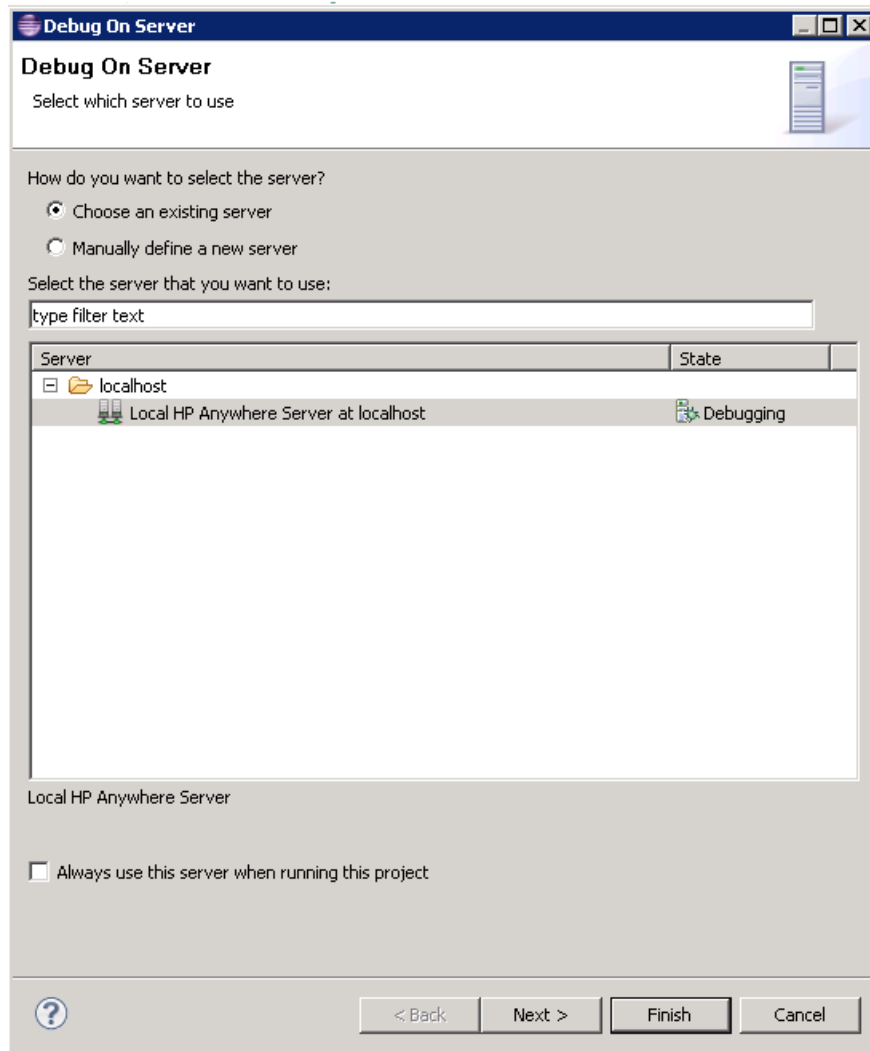
2. Select **Local HP Anywhere Server at localhost** and click **Finish**. The server starts and runs your app on the local server. As the app runs, you can view the log in the Console tab, as shown in the following example:

3. When the run is complete, the app opens in a Chrome browser. The development tools are displayed at the bottom of the page.

4. Debug your code:

   a. Debug your Java code as you would in Eclipse (using breakpoints, and so on). For example, see: http://www.eclipse.org/resources/resource.php?id=503

   b. Debug your JavaScript client code in the Chrome browser. For example, see: https://developers.google.com/chrome-developer-tools/docs/javascript-debugging

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.

If you import an app project that was not created using HP Anywhere IDE, you can instruct HP Anywhere IDE to synchronize any changes you make to your app files with the deployed app on the local server.

This section describes how:

- "To enable/disable automatic hot deployment on your static (JavaScript) files (/src/webapp/):" below. Applies to files, such as CSS or image files.

- "To perform on-demand hot deployment on your Java code:" on the next page.

- "To enable/disable synchronization of your app's configuration files (/src/resources):" on page 35. Applies to <app_name>-admin-settings.xml, <app_name>-ds-provider.xml, <app_name>-user-settings.xml. By default, these files are automatically synchronized.

To enable/disable automatic hot deployment on your static (JavaScript) files (/src/webapp/):

> **Note:** Automatic hot deployment is enabled by default.

1. Prerequisites:

   a. Build your app. For details, see "How do I create an app project in HP Anywhere IDE?" on page 9.

   b. Run your app at least once on the local server. For details, see "How do I run my app on the local server?" on page 17.

   c. Make sure that the local server is up and running.

2. In the project tree:

   a. Right-click the app node and select **Properties**.

   b. In the left pane of the Properties dialog box, select **File synchronization**.

c. In the File synchronization pane, select the **Enable FileSync builder for project** check box and click **OK** to enable hot deployment.

Clear the **Enable FileSync builder for project** check box to disable automatic hot deployment.
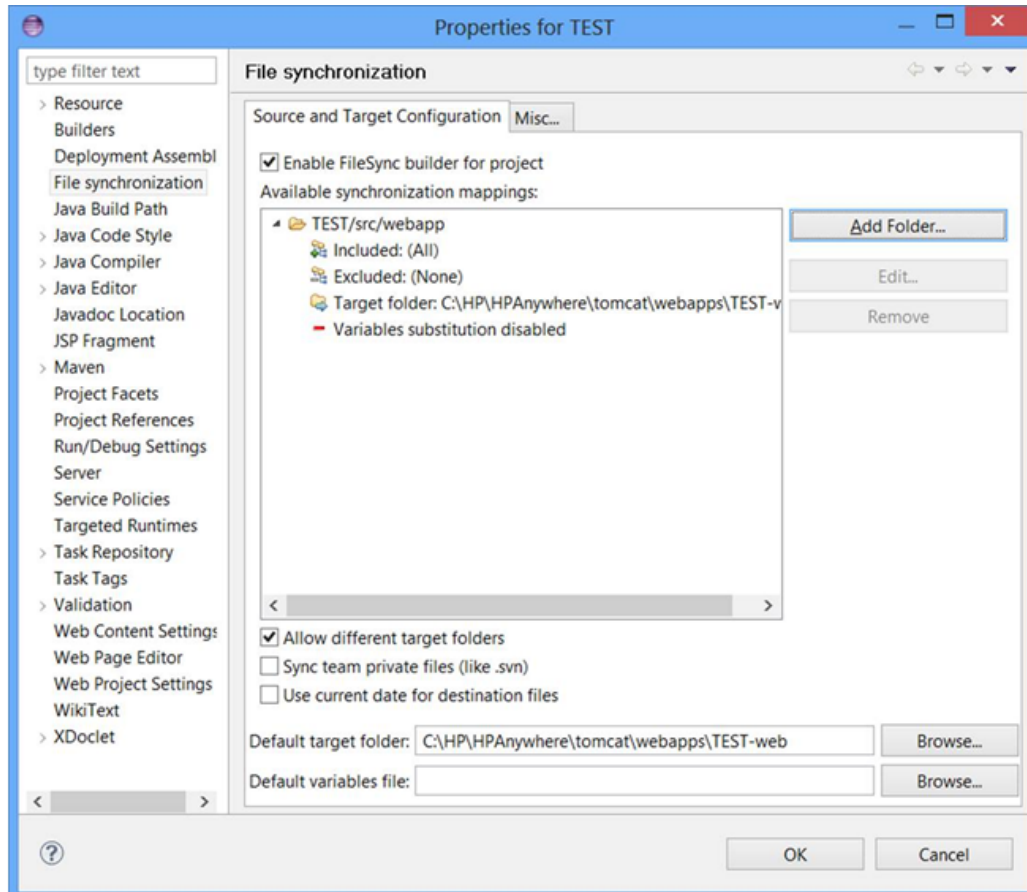
From now on, your changes are saved to the local server whenever you save your app project.

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 9.

    b. Run your app at least once on the local server. For details, see "How do I run my app on the local server?" on page 17.

    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.

Note: If you consistently receive notification that the HP Anywhere server is not started while trying to reload your app, you may need to configure the **eclipse.ini** file, as follows:

1. Close HP Anywhere IDE.

2. Open the **<HP Anywhere_installation_folder>\eclipse\eclipse.ini** file in a text editor.

3. Add the following line to the end of the file:

   -Dhttp.nonProxyHosts=localhost|127.0.0.1

4. Restart HP Anywhere IDE.

To enable/disable synchronization of your app's configuration files (/src/resources):

(Applies to the XML files for admin settings, data source provider, and user settings)

1. In the project tree:

   a. Right-click the app node and select **Properties**.

   b. In the left pane of the Properties dialog box, select:

      ○ **Configure** > **AddHP Anywhere Builder** to enable synchronization of your app's configuration files. (Selected by default for all new HP Anywhere app projects)

      ○ **Configure** > **RemoveHP Anywhere Builder** to disable automatic synchronization of your app's configuration files.

2. (Optional) After you make changes to your code, open the HP Anywhere Console to view the synchronization message. For example:
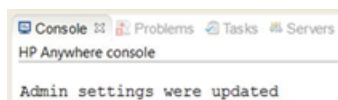
How do I remove apps?

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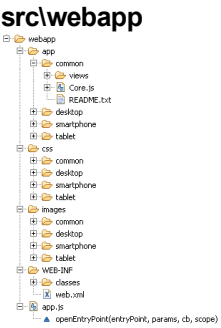
By default, each app project has the following package structure:

```
HelloWorldProject
   src/java
      com.hp.miniapp.sample
      com.hp.miniapp.sample.provider
      com.hp.miniapp.sample.rest
   src/resources
      l10n
         HelloWorldProject_en.properties
      log
         HelloWorldProject.log4j.properties
      descriptor.xml
      HelloWorldProject-admin-settings.xml
      HelloWorldProject-user-settings.xml
   Referenced Libraries
   JRE System Library [JavaSE-1.6]
   src
      webapp
         app
            common
            desktop
            smartphone
            tablet
         css
         images
         WEB-INF
         app.js
   target
      archive-tmp
      HelloWorldProject-web
      surefire
      war
      HelloWorldProject-cp.zip
      HelloWorldProject-web.war
   assembly-zip.xml
   pom.xml
   test-page.html
```

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	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.

src/resources
 	
resources
  l10n
    HelloWorld_en.properties
  log
    log4j.properties
    logging.properties
  descriptor.xml
  HelloWorld-admin-settings.xml
  HelloWorld-ds-provider.xml
  HelloWorld-user-settings.xml
	The server-side configuration files (for example, settings, data source, and localization files).
	
	l10n. Localization folder that includes ***.properties** files for each language in which the app can be displayed.
	
	log.Log files folder that includes **log4j.properties** and **logging.properties**.
	
	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 13.
	
	<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 15.
	
	<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 15.
	
	<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 15.

Referenced Libraries	Third-party libraries that are available for the app (project dependencies).
JRE System Library	The JRE libraries that are available for the app.
src	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.

src\webapp	The client-side resources and settings for the app.

webapp
  app
    common
      views
      Core.js
      README.txt
    desktop
    smartphone
    tablet
  css
    common
    desktop
    smartphone
    tablet
  images
    common
    desktop
    smartphone
    tablet
  WEB-INF
    classes
    web.xml
  app.js
    openEntryPoint(entryPoint, params, cb, scope)
	app. The JavaScript client code for the app.
	
	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.
	
	images. The referenced images used by your app.
	
	Each of the **app**, **css**, and **images** folders contains the following subfolders:
	
	• **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.
	
	The **descriptor.xml** file (described above)
	
	WEB-INF. The directory for the J2EE container, which contains the standard web.xml file and resources for spring, log4j, and third-party logging.
	
	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.

target	The target files that are created when your app runs. This folder is empty when your app project is created. 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.
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 37.

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 37, 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 37, 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 37, 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 37, 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 37, 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 37, 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 37, 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 37, 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. If you try to create an app project without first setting the required proxy, the following error message is displayed:
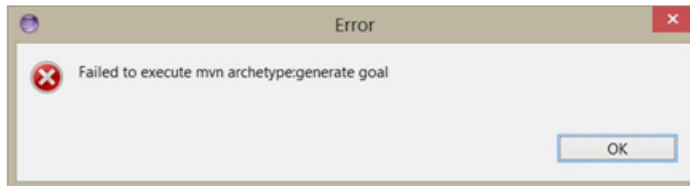
To configure a Maven proxy:

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>
```

How do I update HP Anywhere IDE with the latest HP Anywhere Plugin

The HP Anywhere Plugin is updated regularly. You can download and install the latest HP Anywhere Plugin at your convenience. You can also configure HP Anywhere IDE to automatically receive and download updates to the HP Anywhere Plugin.

To manually install the latest HP Anywhere Plugin and receive automatic update notifications in the future:

1. In HP Anywhere IDE, select **Help > Install New Software**. The Available Software wizard opens.

2. In the **Work with** field, select **HP Anywhere Eclipse Update Site**. The text area in the center of the wizard screen dynamically displays HP-related check boxes.

3. Select the **HP** check box.
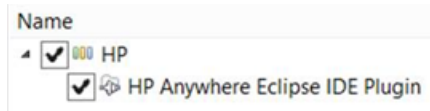
4. Click **Next** and follow the wizard's instructions.

5. When prompted, restart HP Anywhere IDE.

This process installs the latest version of the plugin and configures HP Anywhere IDE to automatically notify you when future updates become available.

To receive/configure automatic notifications when a new plugin update is available:

1. Select **Window > Preferences > Install/Update > Automatic Updates**.

2. In the Automatic Updates pane, select the **Automatically find new updates and notify me** check box.

3. Configure the schedule and download options for receiving notifications when updates to the plugin are available.

4. Click **OK**.

From now on, when updates to the plugin are released, you will receive notifications, enabling you to install the latest version of the plugin at your convenience.

To disable automatic notifications when a new plugin update is a available:

1. Select **Window > Preferences > Install/Update > Automatic Updates**.

2. In the Automatic Updates pane, clear the **Automatically find new updates and notify me** check box.

3. Click **OK**.

You will no longer be notified when new updates are available.

How do I run an app that was created using an earlier version of HP Anywhere IDE?

After you upgrade HP Anywhere IDE, your app may not run seamlessly on the newer version because it was not compiled against the latest HP Anywhere server version:

To upgrade your app to the current version:

1. In the HP Anywhere IDE Package Explorer, navigate to: **<app main node>/pom.xml** and double-click the file to open it.

2. In the Design or Source tab, modify the **btoa.build.version** value to the latest version, for example: **10.02.100**

3. Run the app to recompile it.

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>
```