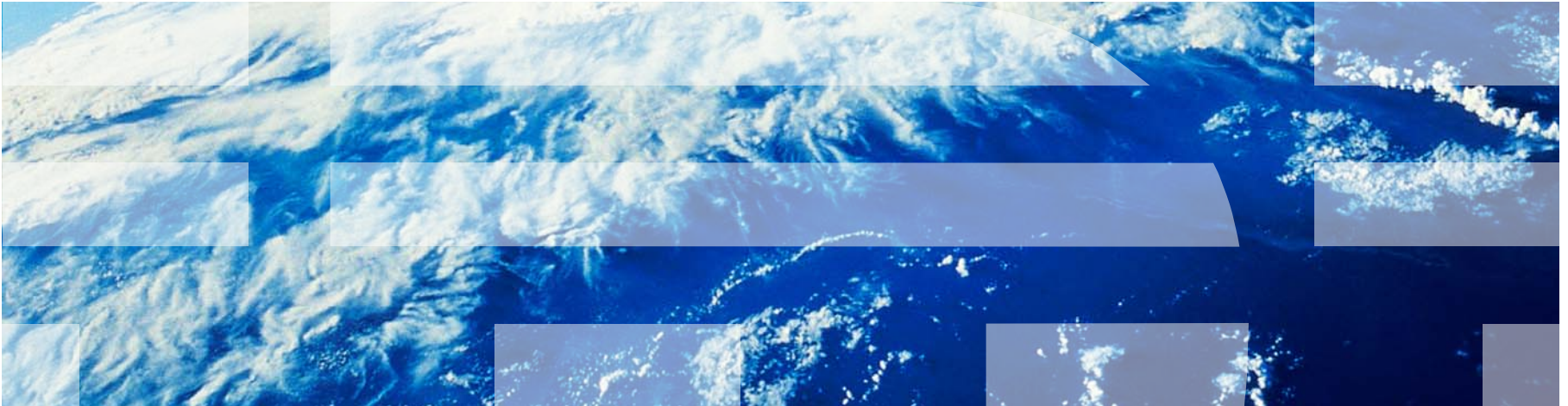


IBM Worklight V6.1.0 Getting Started

Creating your first Worklight application



Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Worklight is a trademark or registered trademark of Worklight, an IBM Company. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “[Copyright and trademark information](#)” at www.ibm.com/legal/copytrade.shtml.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- Other company products or service names may be trademarks or service marks of others.
- This document may not be reproduced in whole or in part without the prior written permission of IBM.

About IBM®

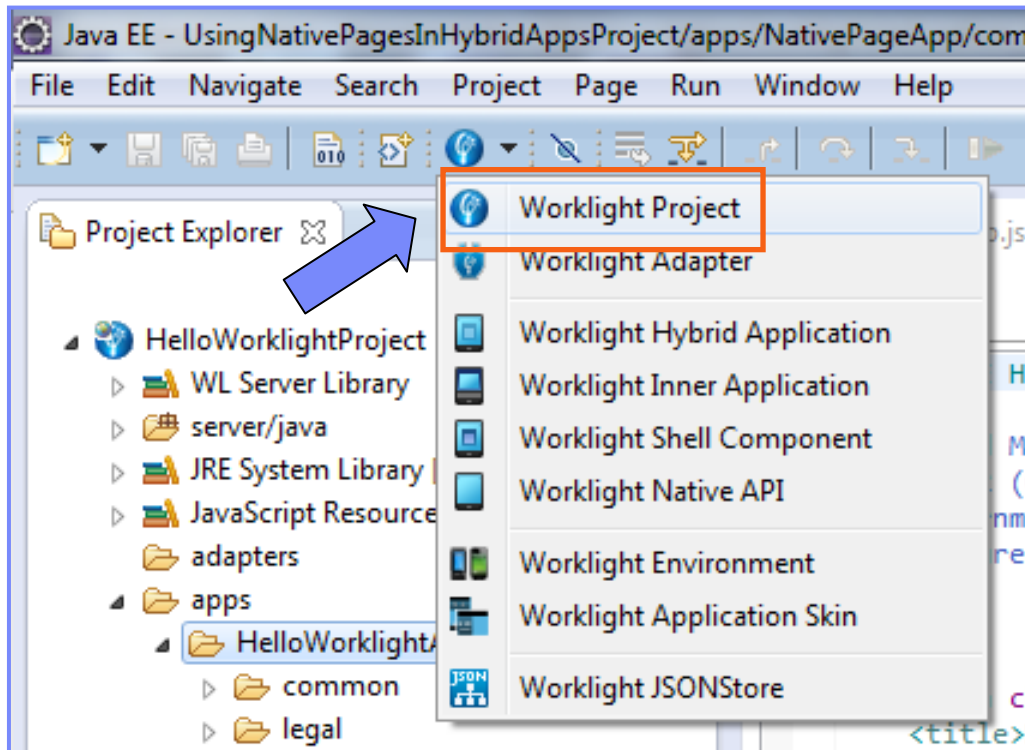
- See <http://www.ibm.com/ibm/us/en/>

Agenda

- Write your first application
- Become familiar with the application files and HTML structure
- Use Worklight Studio to deploy the application
- Worklight Studio Tips
- Exercise

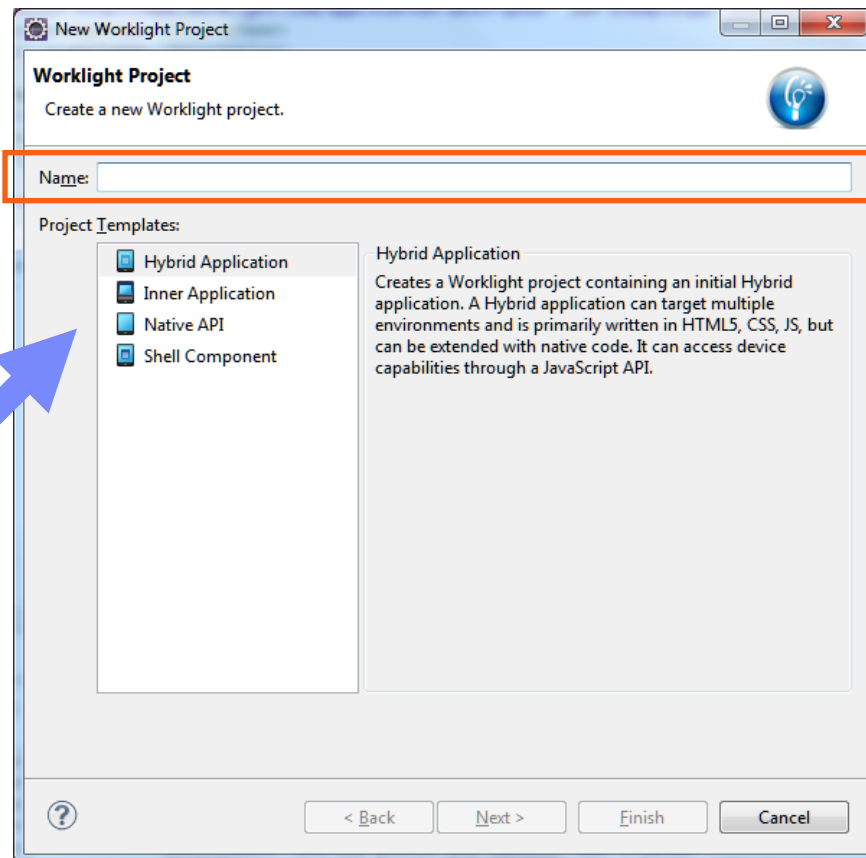
Hello Worklight Application

- Create an IBM Worklight® project.



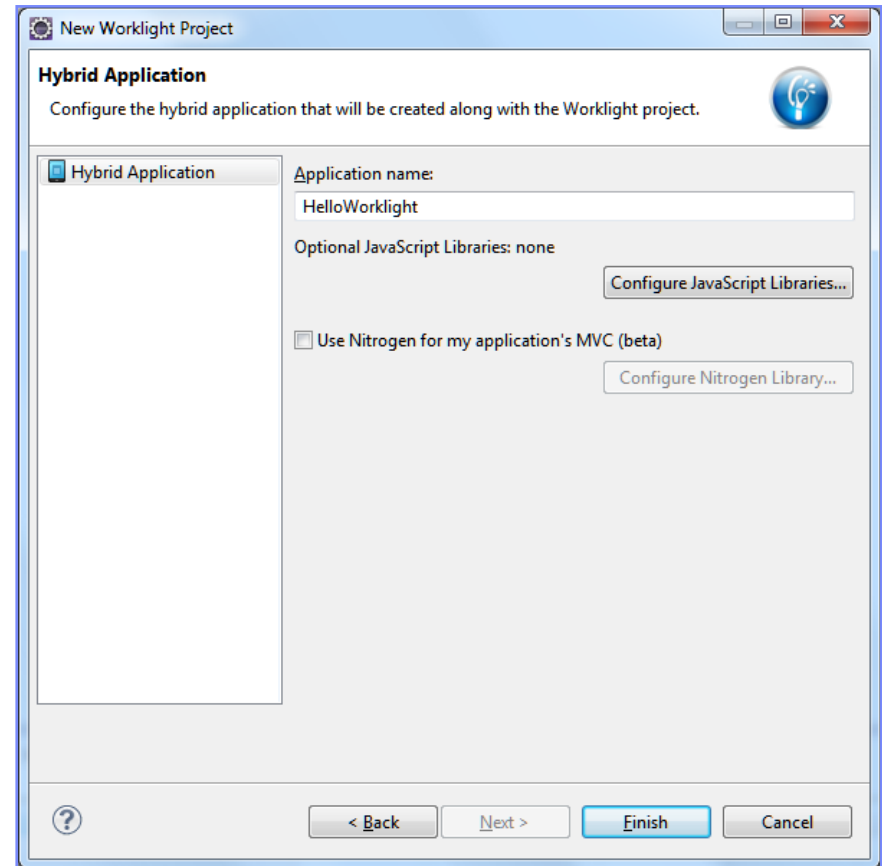
Hello Worklight Application

- Name it “HelloWorklightProject” and select a **Hybrid Application** template.



Hello Worklight Application

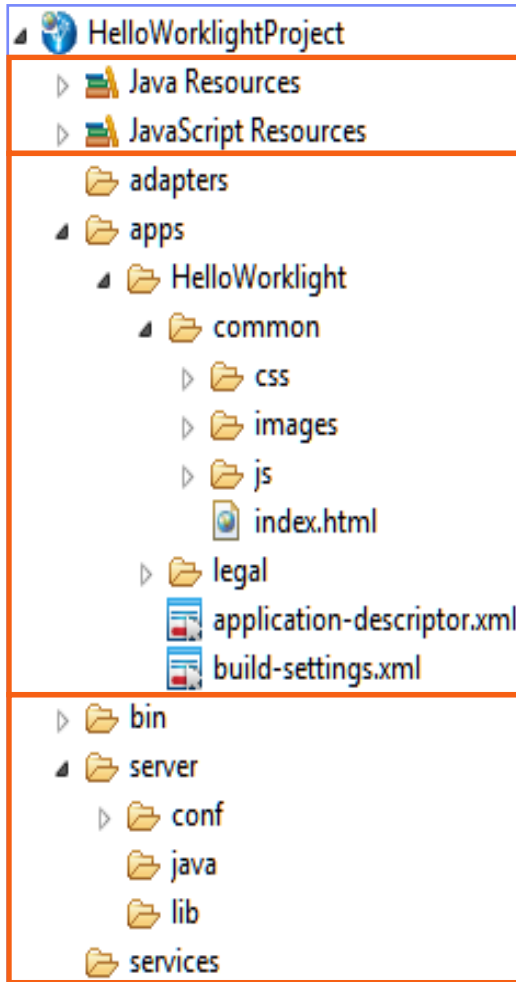
- Name your application “HelloWorklight”.
- JavaScript™ frameworks can be added to your project in this screen.
- To learn more about adding frameworks to your application, read the *Client-side development basics – Working with UI frameworks Getting Started* module.
- Click **Finish** when done.



Agenda

- Write your first application
- Become familiar with the application files and HTML structure
- Use Worklight Studio to deploy the application
- Worklight Studio Tips
- Exercise

Worklight Project Structure



References required for application development and deployment

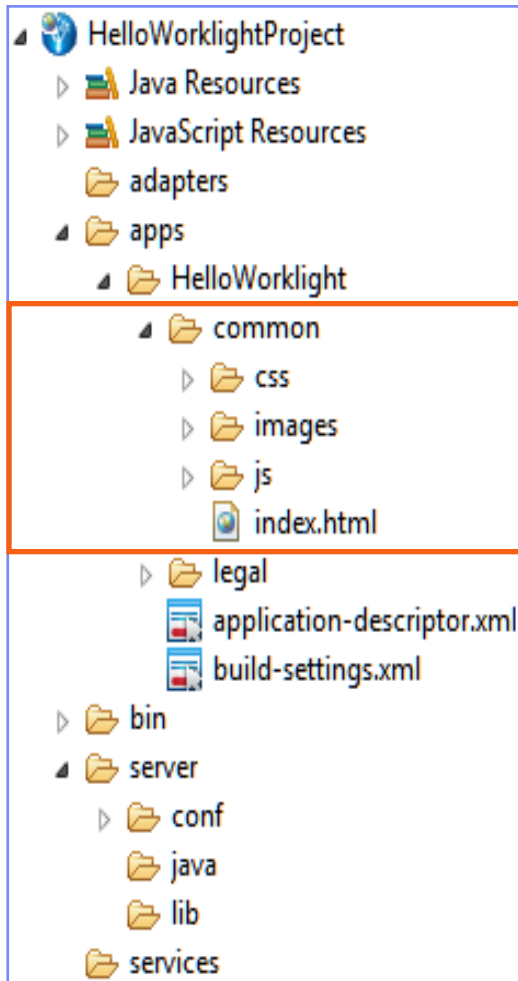


Project applications and adapters



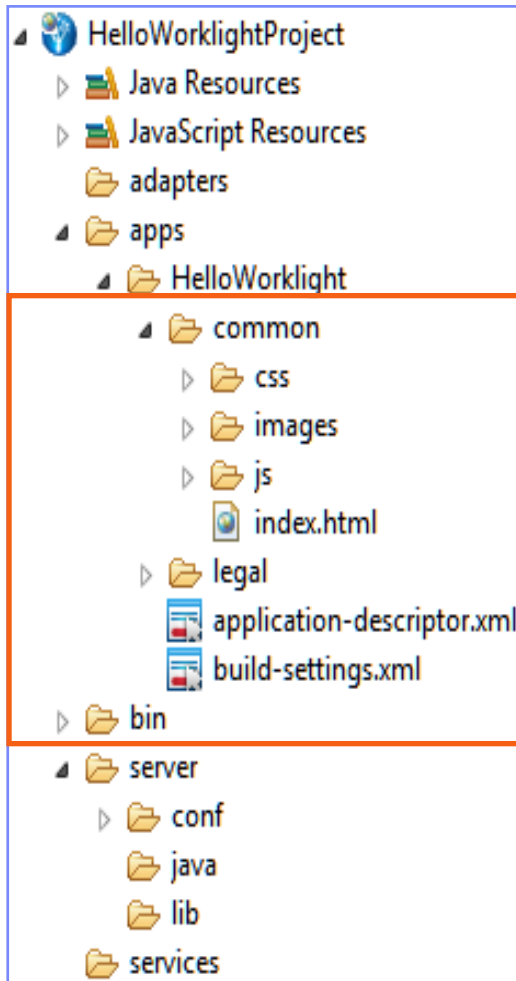
Server customization components

HelloWorklight – Project Structure



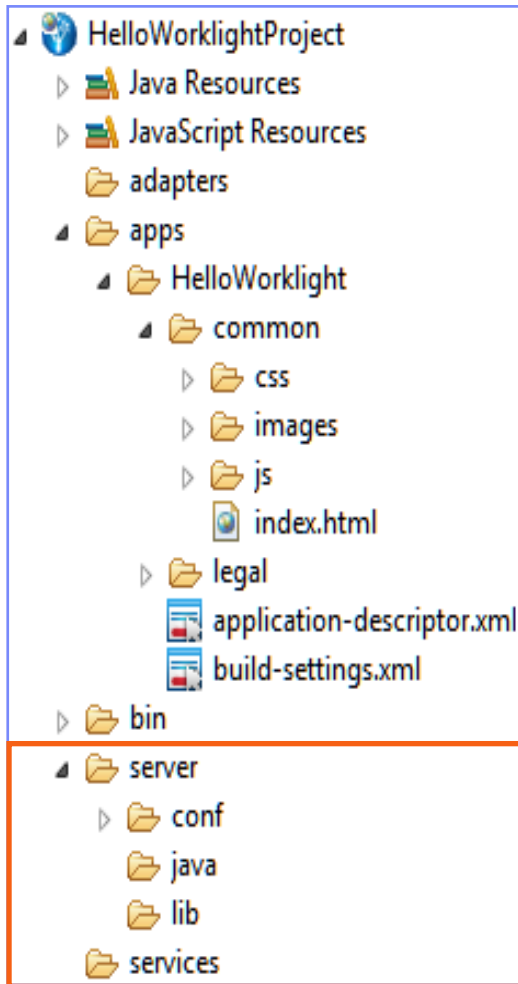
- The default environment is called **common**.
- The **common** environment contains all the resources that are shared between environments.
- Right-click **apps > New > Worklight environment** to add new environments:
 - A new environment is created.
 - The resources of the new environment have the following relationship with the common resources:
 - **images** – override the common images when both have the same name.
 - **css** – extend, override, or both, the common CSS files.
 - **js** – extend the common application instance JS object. The environment class extends the common app class.
 - **HTML** – override the common HTML code when both have the same name.

HelloWorklight – Project Structure - continued



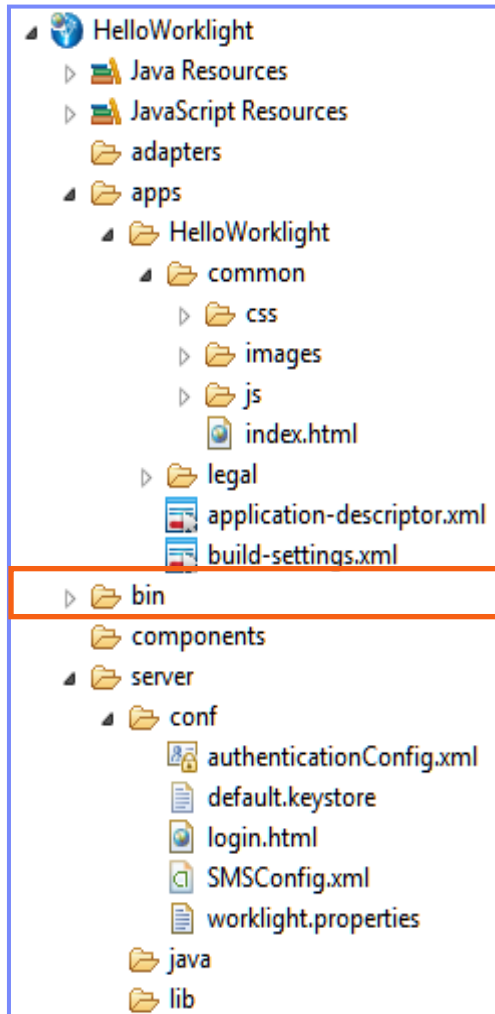
- Common Environment:
 - **Index.html**: The main HTML file.
 - **css: main.css** – main application CSS file.
 - **images**: Default Worklight images for the common environment.
 - **js**:
 - **initOptions.js**: Contains initialization options for the application.
 - **main.js**: The main JavaScript file for the application.
 - **messages.js**: JSON object that holds all app messages. Can be used as the source for translation.
- The **legal** folder holds all legal related docs.
- The **application-descriptor.xml** file contains the application metadata.
- The **build-settings.xml** file contains configuration options for minification and concatenation.
- The **bin** folder contains generated **.wlappr** and **.wladapter** files that can be used to deploy to a remote Worklight Server.

HelloWorklight – Project Structure - continued



- The **server** folder contains files that are used for server-side customization of a project:
 - The **conf** folder contains:
 - **authenticationConfig.xml**, which defines authentication realm and security tests.
 - **default.keystore**, a default SSL certificate provided by Worklight.
 - **login.html**, which presents a login form for web environments and Worklight Console.
 - **SMSConfig.xml**, which defines SMS Gateways.
 - **worklight.properties**, which defines properties used by the Worklight Server.
 - The **java** folder is used to hold Java™ classes that will be compiled and deployed to a Worklight Server after you build your application. You can put your custom Java code here.
 - The **lib** folder is used for JAR files that are deployed to the server.
- The **services** folder contains discovered back-end services.

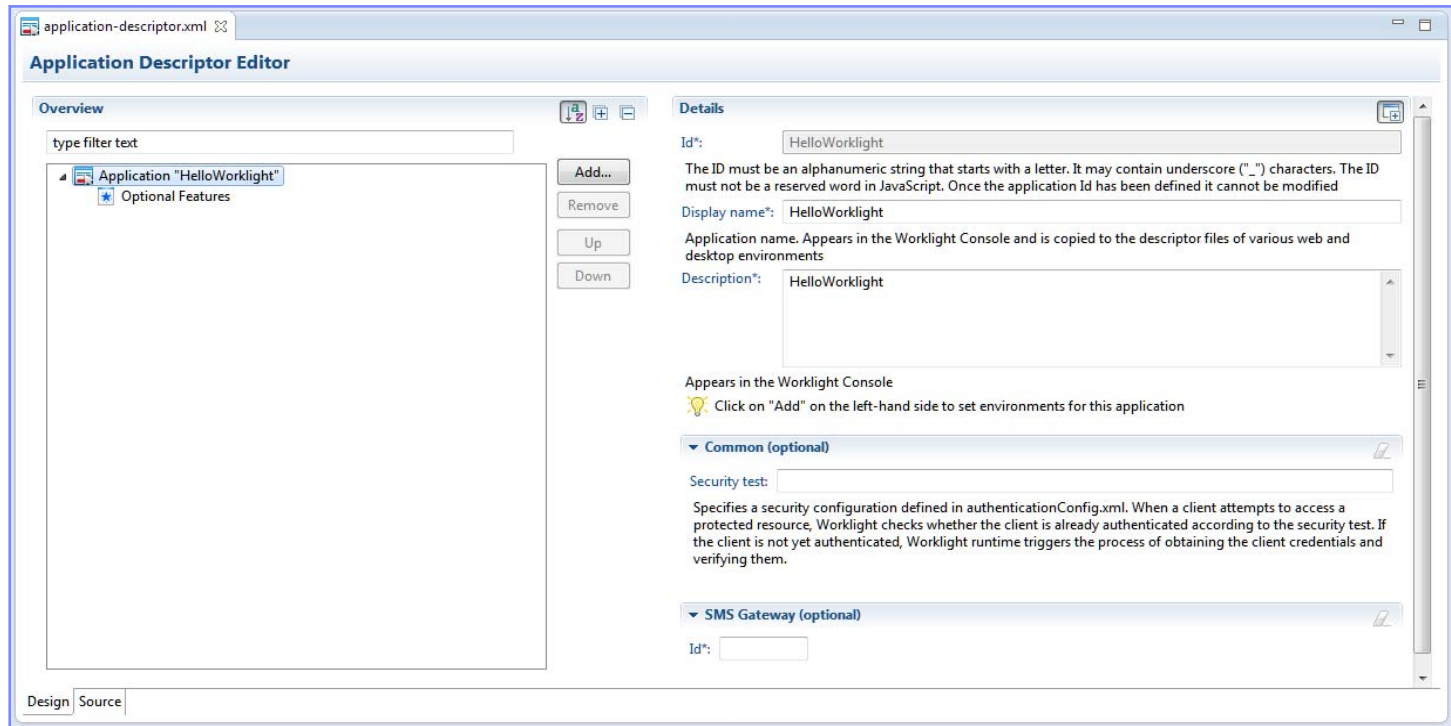
HelloWorklight – Project Structure - continued



- The **bin** folder contains project artifacts that are deployed to the Worklight Server.
- The evaluation version of Worklight Studio deploys those artifacts to the embedded Worklight Development Server automatically as a part of the build process.
 - **.wlapp** files are application bundles.
 - **.wladapter** files are adapters.
 - **.jar** and **.war** files are server customization files that contain worklight.properties, authenticationConfig.xml, and custom Java code.

HelloWorklight – application-descriptor.xml

- An application descriptor is an XML file that stores the metadata for an application.
- This file can be edited with the Design or Source editors.



HelloWorklight - application-descriptor.xml - continued

- Based on the W3C Widget Packaging and Configuration
- Contains application properties used at build time

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

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

Specify the application name, description, author details and thumbnail image to be displayed in the Worklight Console.

HelloWorklight - application-descriptor.xml - continued

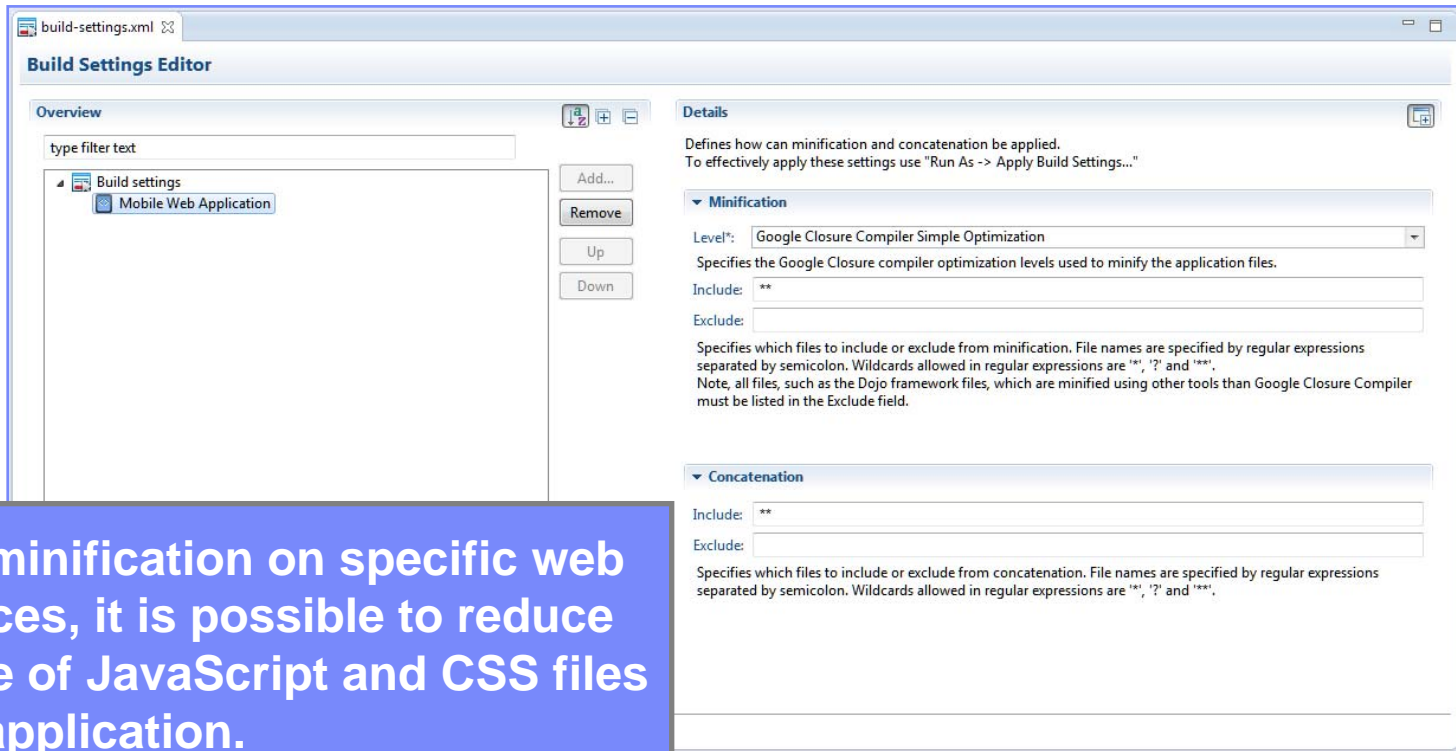
- Based on the W3C Widget Packaging and Configuration
- Contains application properties used at build time

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

Environment-specific information is inserted automatically as new environments are added to the Worklight project.

Hello Worklight – build-settings.xml

- The Build Settings is an XML file that contains configuration options for minification and concatenation of the Desktop Browser and Mobile Web environments' web resources.
- This file can be edited with the Design or Source editors.

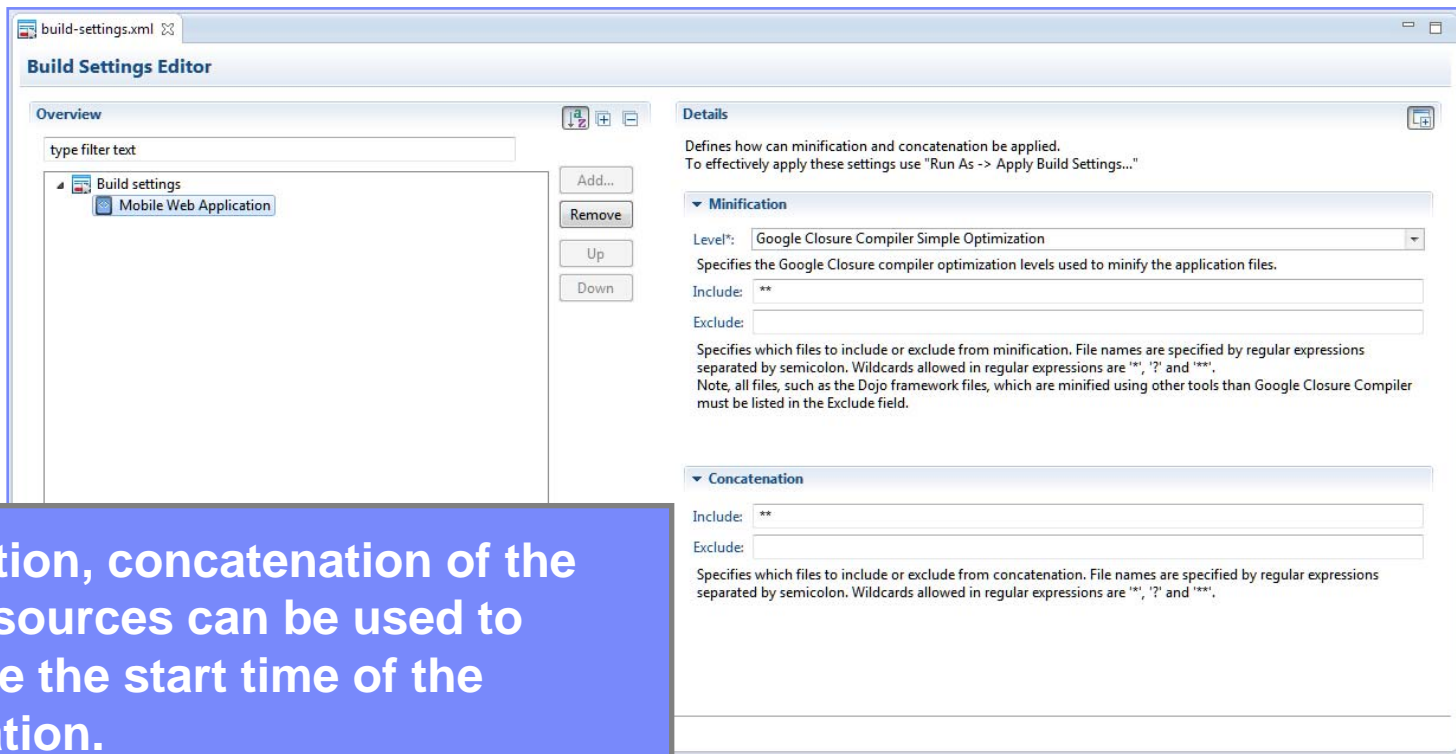


The screenshot shows the 'Build Settings Editor' window for 'build-settings.xml'. The interface is divided into an 'Overview' pane on the left and a 'Details' pane on the right. The 'Overview' pane shows a tree view with 'Build settings' expanded to show 'Mobile Web Application'. The 'Details' pane has a 'Minification' section with a dropdown menu set to 'Google Closure Compiler Simple Optimization', an 'Include' field with '**', and an empty 'Exclude' field. Below this is a 'Concatenation' section with an 'Include' field with '**' and an empty 'Exclude' field. A blue callout box is overlaid on the bottom left of the screenshot.

Using minification on specific web resources, it is possible to reduce the size of JavaScript and CSS files in the application.

HelloWorklight – build-settings.xml - continued

- The Build Settings is an XML file that contains configuration options for minification and concatenation of the Desktop Browser and Mobile Web environments' web resources.
- This file can be edited with the Design or Source editors.



In addition, concatenation of the web resources can be used to improve the start time of the application.

HelloWorklight – Single DOM model

- IBM Worklight hybrid applications use a single DOM model.
- The single DOM model means that you should never navigate between various HTML files using hyperlinks or by changing the **window.location** property.
- Instead, multi-page interfaces should be implemented by loading an external HTML file content using Ajax requests and injecting it into an existing DOM.
 - You must do so because the main application HTML file loads Worklight client-side JavaScript framework files, so after the browser navigates from one HTML file to another, the JavaScript context and loaded scripts are lost.
- Most JavaScript UI frameworks available today (for example, jQuery Mobile, Sencha Touch, Dojo Mobile) provide an extensive range of APIs to achieve the required multi-page navigation.
- This module explains the principles of a single-page application.
- Principles of multi-page applications built with a single DOM model are explained in later modules.

Index.html

- During application run time, the main HTML document cannot be replaced by another HTML document.

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

Default application HTML template complies with HTML5 standard markup, but any other DOCTYPE can be specified.

Index.html - continued

- During application run time, the main HTML document cannot be replaced by another HTML document.

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

Worklight client-side framework uses jQuery library for internal functions. By default, the \$ char is assigned to the internal jQuery in the main JavaScript file.

Remove this line if your application does not require jQuery, or if you want to use your own version of jQuery.

Index.html - continued

- During application run time, the main HTML document cannot be replaced by another HTML document.

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

Worklight client framework initialization is bound to onload event in the `initOptions.js` file. For more information about the initialization options, see the IBM Worklight user documentation.

Index.html - continued

- During application run time, the main HTML document cannot be replaced by another HTML document.

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

The blue arrow indicates where to insert your HTML code.

initOptions.js

- The **initOptions.js** file contains Worklight framework initialization settings.
- It is also responsible for initializing the Worklight framework once the **<body>** element has finished loading.
- By default, the Worklight application starts in offline mode.

```
var wlInitOptions = {
  connectOnStartup : false,

  logger : {enabled: true, level: 'debug', stringify: true, pretty: false,
    tag: {level: false, pkg: true}, whitelist: [], blacklist: []},

  analytics : {
    enabled: false
  }
};

if (window.addEventListener) {
  window.addEventListener('load', function() { WL.Client.init(wlInitOptions); }, false);
} else if (window.attachEvent) {
  window.attachEvent('onload', function() { WL.Client.init(wlInitOptions); });
}
```

- Initialization options are documented in the file itself, as well as in the IBM Worklight user documentation.

Main.js

- When creating an application, a **main.js** file is created and holds its JavaScript portion.
- It contains a **wlCommonInit()** function that is invoked automatically after Worklight framework initialization finishes.
- You can add your application initialization code there.
- This function is used in environment-specific JavaScript files to have a common initialization start point. There are more details in subsequent Getting Started modules.

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


Main.js - continued

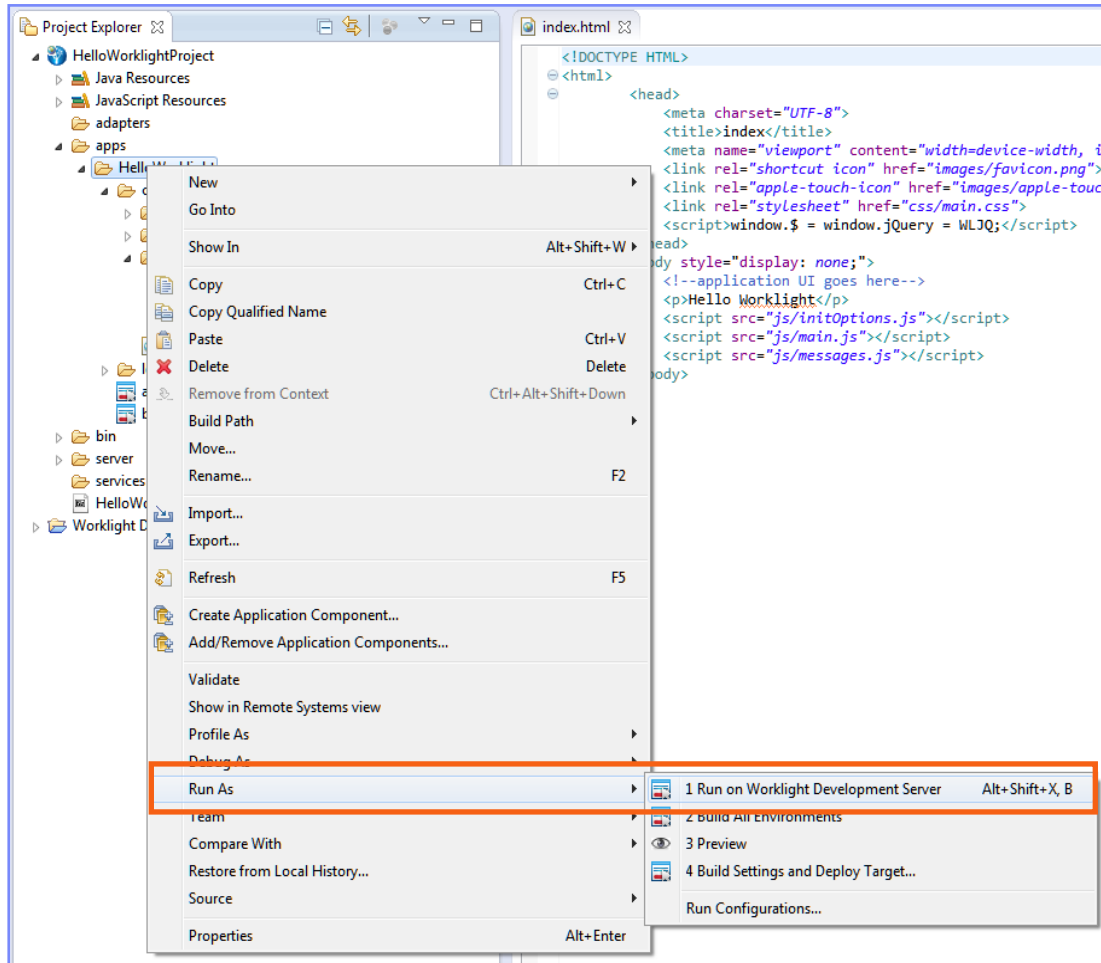
- As described previously, the Worklight application starts in offline mode by default.
- To begin communicating with Worklight Server, you need to follow the instructions in the default **wlCommonInit()** function.

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

Agenda

- Write your first application
- Become familiar with the application files & HTML structure
- Use Worklight Studio to deploy the application
- Worklight Studio Tips
- Exercise

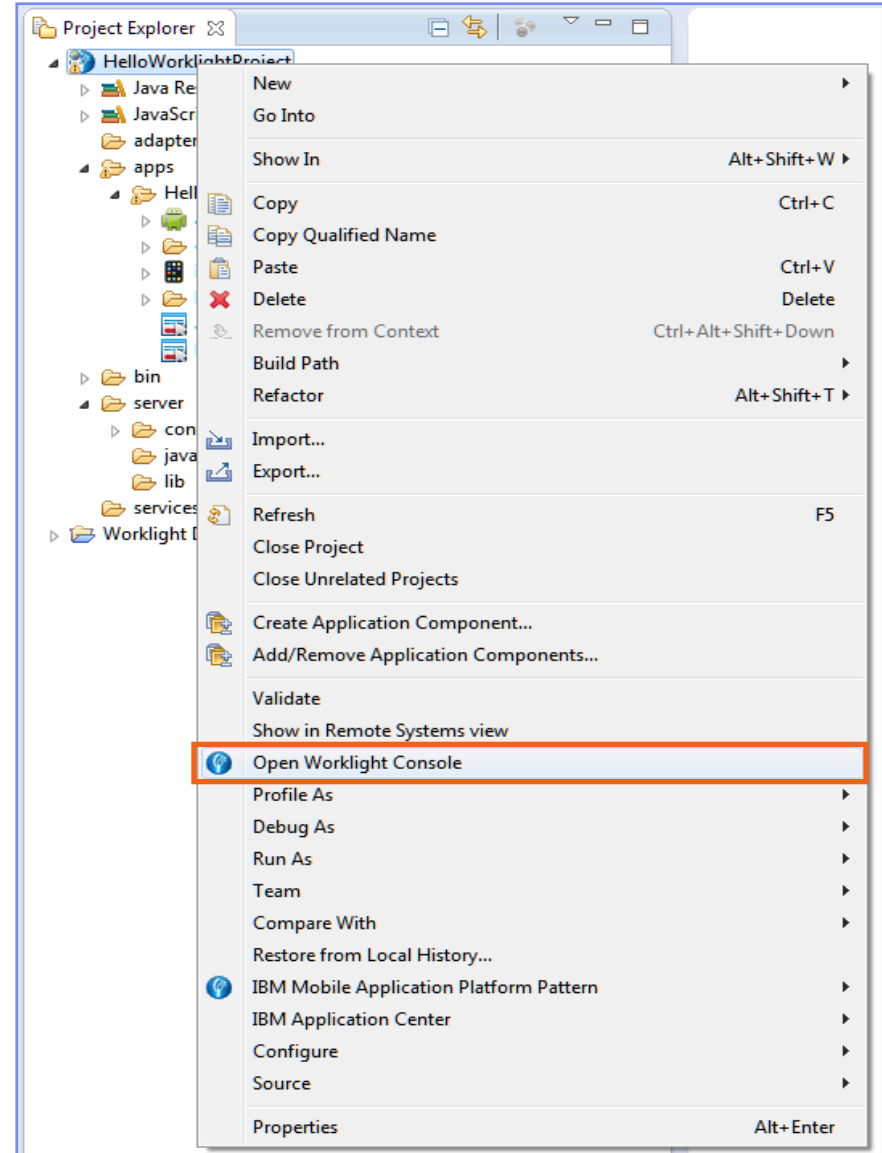
Building an Application



- Select an app to build, and then right-click the app name.
- Click **Run As > Run on Worklight Development Server**.
- After the build completes, the application is available for preview in the **catalog** tab of the Worklight Console.
- By default the port number is **10080**, and the context root is the project name.

Worklight Console

- To run the console, right-click the project name and select **Open Worklight Console**.
- To view Worklight Console in an external browser window, from the top menu bar in Eclipse go to **Window > Preferences > General > Web Browser** and select the “Use external web browser” radio button. There you can also select your preferred browser.

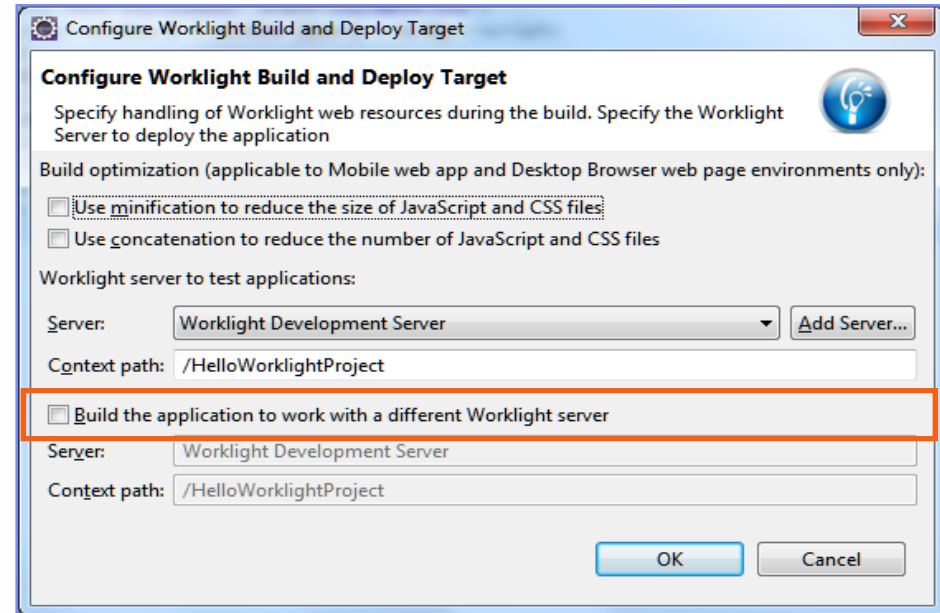


Worklight Console - continued

- Since IBM Worklight V6.0.0.0, the Worklight Console URL has changed:
 - The port is changed from **8080** to **10080**.
 - The URL also now contains the project context root.
 - For example: <http://localhost:10080/MyProject/console>
- The various features of Worklight Console will be gradually discovered and explained as you progress through the Getting Started modules.

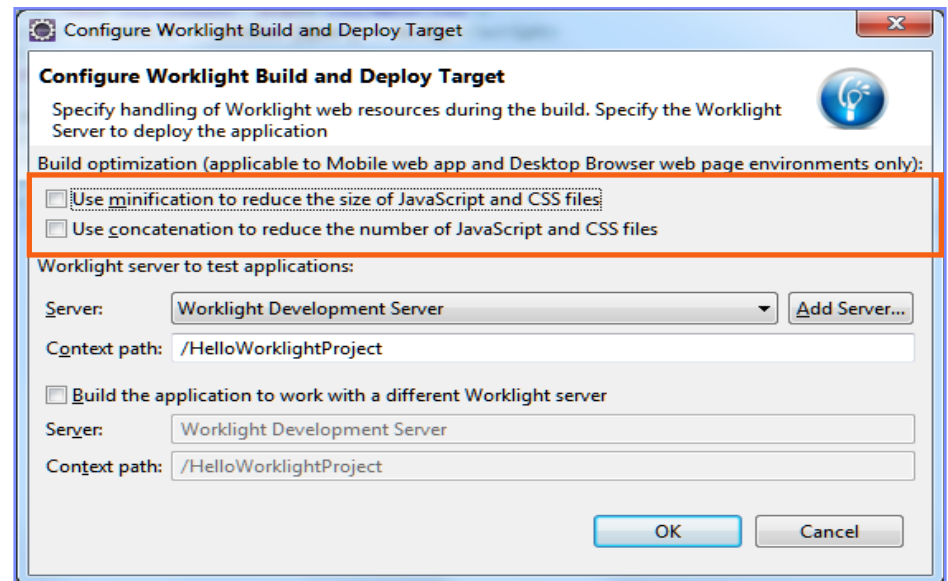
Build Settings and Deploy Target

- To change the target server, right-click the application name and select **Run As > Build Settings and Deploy Target**.
- Select the “Build the application to work with a different Worklight server” check box.
- Update the server and context path and click **OK**.
- Next, you must now re-build the application.
- **Note:** Only Apache Tomcat and WebSphere® Application Server Liberty Profile are supported.



Build Settings and Deploy Target - continued

- If minification and/or concatenation are required for the application, be sure to also enable this build setting prior to building the application.
- To do so, right-click the application name and select **Run As > Build Settings and Deploy Target**.
- Choose one or all of the highlighted options
- Click **OK**.



Agenda

- Write your first application
- Become familiar with the application files & HTML structure
- Use Worklight Studio to deploy the application
- **Worklight Studio Tips**
- Exercise

Worklight Studio Tips

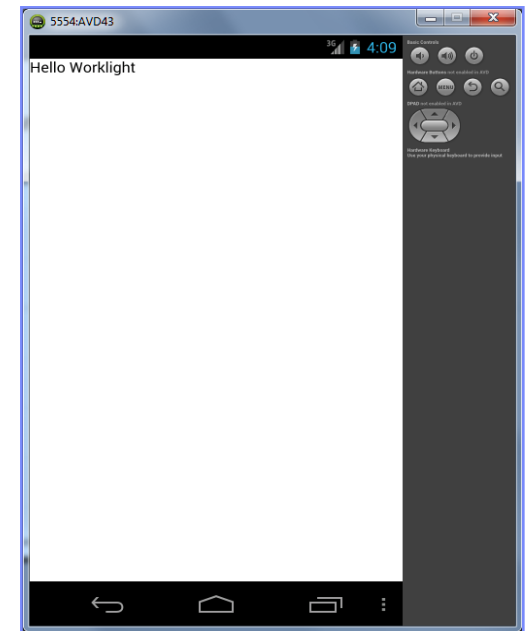
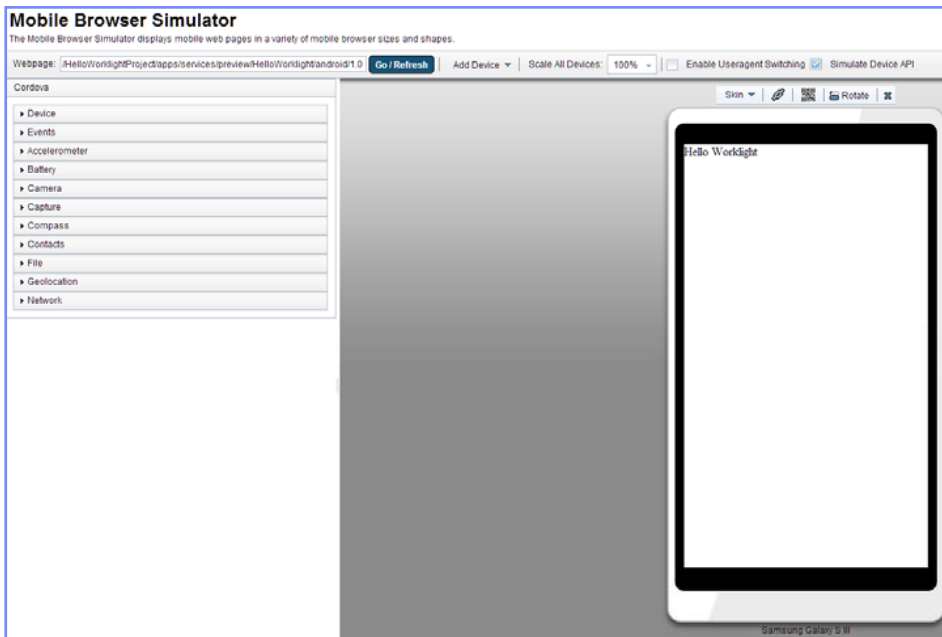
- Since Worklight Studio V6.1.0, it is no longer required to re-build and deploy the application during development to view the changes that have been made.
- Instead, simply save the file you are working on.
- Preview the application, or if already previewed, simply refresh the tab you are viewing, and the changes done in Worklight Studio are immediately accessible for further examination and debugging.

Agenda

- Write your first application
- Become familiar with the application files & HTML structure
- Use Worklight Studio to deploy the application
- Worklight Studio Tips
- Exercise

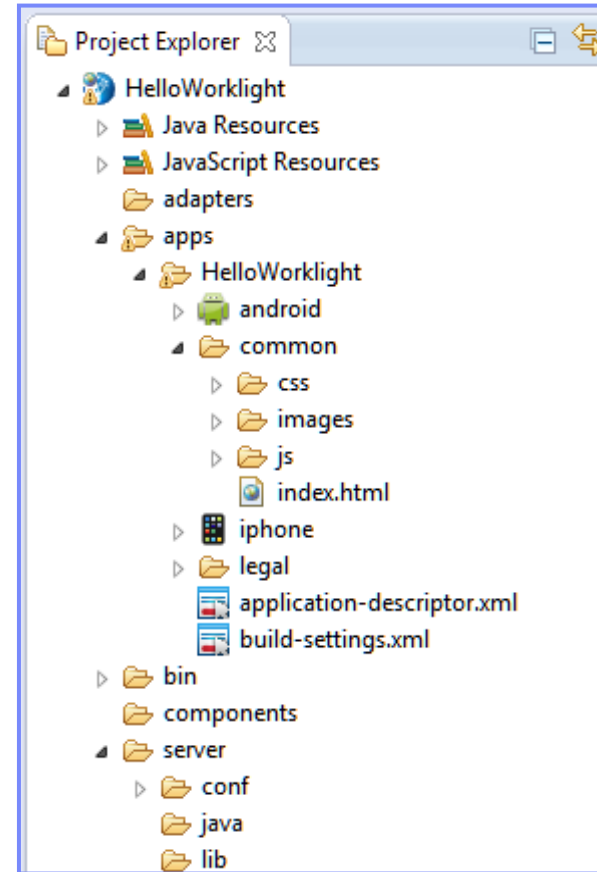
Exercise

- HelloWorld Application
 - Use the wizard to create a **HelloWorklightProject** project.
 - Use the wizard to create an application; name it **HelloWorklight**.
 - Use Worklight Studio to deploy your application.
 - Use Worklight Console to preview your application.



Exercise solution

- The sample for this module can be found on the Getting Started page of the IBM Worklight documentation website at <http://www.ibm.com/mobile-docs>.
- To import the solution to Eclipse, right-click in a blank area in the Project Explorer view and choose **Import > Import > General > Existing Projects into Workspace** and either select the archive (zipped file) or root directory (if unzipped).



Check yourself questions

- What is the default application structure?
 - A single HTML file, and a number of CSS and JS files.
 - A number of HTML and JS files, and a single CSS file.
 - A number of HTML, JS, and CSS files.
 - A single HTML, single CSS, and a single JS file.
- Which of the following server-side components cannot be added as part of a Worklight Project?
 - Source code of custom server-side Java classes.
 - Properties that are used by the server.
 - JAR files that contain server-side Java code.
 - JMeter application to test server performance.
- The correct place to specify the URL that the application uses is:
 - authenticationConfig.xml.
 - worklight.properties.
 - application-descriptor.xml.
 - This URL is set separately per platform in a dedicated settings file.

Check yourself questions

- What is the default application structure?
 - A single HTML file, and a number of CSS and JS files.
 - A number of HTML and JS files, and a single CSS file.
 - A number of HTML, JS, and CSS files.
 - A single HTML, single CSS, and a single JS file.
- Which of the following server-side components cannot be added as part of a Worklight Project?
 - Source code of custom server-side Java classes.
 - Properties that are used by the Server.
 - JAR files that contain server-side Java code.
 - JMeter application to test Server performance.
- The correct place to specify the URL that the application uses is:
 - authenticationConfig.xml.
 - worklight.properties.
 - application-descriptor.xml.
 - This URL is set separately per platform in a dedicated settings file.

Notices

- Permission for the use of these publications is granted subject to these terms and conditions.
- This information was developed for products and services offered in the U.S.A.
- IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
- IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
 - IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.
- For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:
 - Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan
- **The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.
- This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.
- Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
- IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
- Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:
 - IBM Corporation
Dept F6, Bldg 1
294 Route 100
Somers NY 10589-3216
USA

- Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.
- The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.
- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

- This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
- Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:
 - © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Privacy Policy Considerations

- IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.
- Depending upon the configurations deployed, this Software Offering may use session cookies that collect session information (generated by the application server). These cookies contain no personally identifiable information and are required for session management. Additionally, persistent cookies may be randomly generated to recognize and manage anonymous users. These cookies also contain no personally identifiable information and are required.
- If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent. For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the sections entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Support and comments

- For the entire IBM Worklight documentation set, training material and online forums where you can post questions, see the IBM website at:
 - <http://www.ibm.com/mobile-docs>
- **Support**
 - Software Subscription and Support (also referred to as Software Maintenance) is included with licenses purchased through Passport Advantage and Passport Advantage Express. For additional information about the International Passport Advantage Agreement and the IBM International Passport Advantage Express Agreement, visit the Passport Advantage website at:
 - <http://www.ibm.com/software/passportadvantage>
 - If you have a Software Subscription and Support in effect, IBM provides you assistance for your routine, short duration installation and usage (how-to) questions, and code-related questions. For additional details, consult your IBM Software Support Handbook at:
 - <http://www.ibm.com/support/handbook>
- **Comments**
 - We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.
 - For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.
 - When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state.
 - Thank you for your support.
 - Submit your comments in the IBM Worklight Developer Edition support community at:
 - <https://www.ibm.com/developerworks/mobile/worklight/connect.html>
 - If you would like a response from IBM, please provide the following information:
 - Name
 - Address
 - Company or Organization
 - Phone No.
 - Email address

Thank You

