AT&T Developer Program

AT&T API Platform Adapter for IBM® Worklight®

Installation and Setup Guide

Revision Date **April 2013**

Legal Disclaimer

This document and the information contained herein (collectively, the "**Information**") is provided to you (both the individual receiving this document and any legal entity on behalf of which such individual is acting) ("**You**" and "**Your**") by AT&T, on behalf of itself and its affiliates ("**AT&T**") for informational purposes only. AT&T is providing the Information to You because AT&T believes the Information may be useful to You. The Information is provided to You solely on the basis that You will be responsible for making Your own assessments of the Information and are advised to verify all representations, statements and information before using or relying upon any of the Information. Although AT&T has exercised reasonable care in providing the Information to You, AT&T does not warrant the accuracy of the Information and is not responsible for any damages arising from Your use of or reliance upon the Information. You further understand and agree that AT&T in no way represents, and You in no way rely on a belief, that AT&T is providing the Information in accordance with any standard or service (routine, customary or otherwise) related to the consulting, services, hardware or software industries.

AT&T DOES NOT WARRANT THAT THE INFORMATION IS ERROR-FREE. AT&T IS PROVIDING THE INFORMATION TO YOU "AS IS" AND "WITH ALL FAULTS." AT&T DOES NOT WARRANT, BY VIRTUE OF THIS DOCUMENT, OR BY ANY COURSE OF PERFORMANCE, COURSE OF DEALING, USAGE OF TRADE OR ANY COLLATERAL DOCUMENT HEREUNDER OR OTHERWISE, AND HEREBY EXPRESSLY DISCLAIMS, ANY REPRESENTATION OR WARRANTY OF ANY KIND WITH RESPECT TO THE INFORMATION, INCLUDING, WITHOUT LIMITATION, ANY REPRESENTATION OR WARRANTY OF DESIGN, PERFORMANCE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, OR ANY REPRESENTATION OR WARRANTY THAT THE INFORMATION IS APPLICABLE TO OR INTEROPERABLE WITH ANY SYSTEM, DATA, HARDWARE OR SOFTWARE OF ANY KIND. AT&T DISCLAIMS AND IN NO EVENT SHALL BE LIABLE FOR ANY LOSSES OR DAMAGES OF ANY KIND, WHETHER DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, SPECIAL OR EXEMPLARY, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF GOODWILL, COVER, TORTIOUS CONDUCT OR OTHER PECUNIARY LOSS, ARISING OUT OF OR IN ANY WAY RELATED TO THE PROVISION, NON-PROVISION, USE OR NON-USE OF THE INFORMATION, EVEN IF AT&T HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES OR DAMAGES.

Table of Contents

Introduction

The AT&T API Platform Adapter for IBM® Worklight® provides a simplified way for Worklight developers to access the AT&T API Platformand RESTful APIs.

By significantly reducing the complexity of building applications that use the AT&T API Platform, the adapter helps developers quickly bring robust hybrid mobile applications to market.

The adapter facilitates access to the following AT&T Platform RESTful APIs:

- Advertisements
- Device Capability
- Notary
- OAuth
- Payment
- SMS
- Speech To Text
- Text To Speech

Note: To learn more about the AT&T API Platform, see the AT&T Developer Program web site.

Architectural Overview

Figure 1 shows the relationship between the mobile device where the app is running, the Worklight server, and the AT&T API platform.

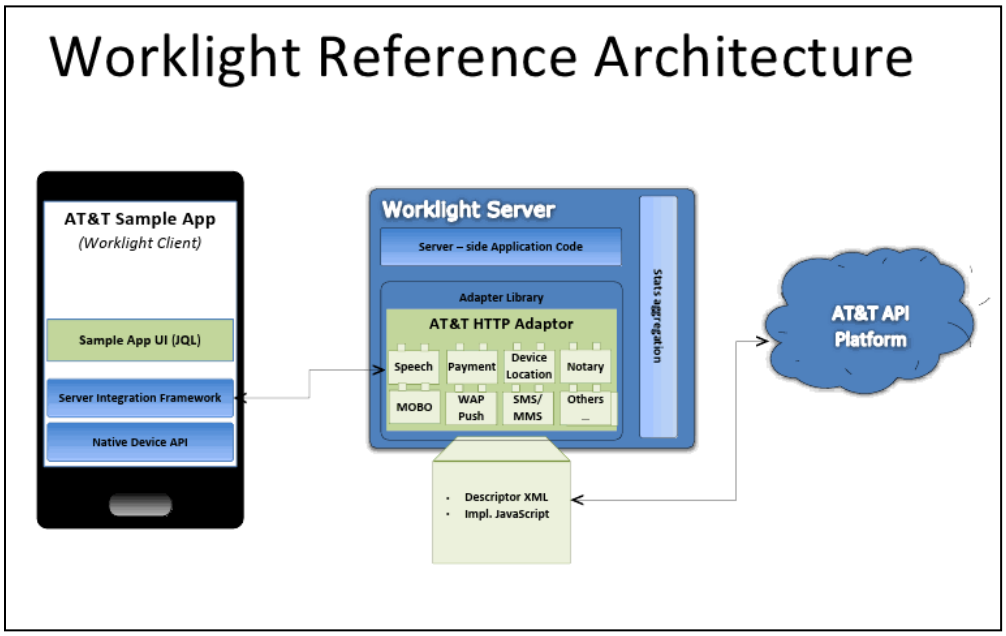

Figure 1: AT&T Worklight Architecture

Installing the Worklight Adapters Components

This section contains instructions for installing the components necessary to create apps using the adapter. The steps are:

1. Download Worklight Studio
2. Download the AT&T API Platform Adapters for IBM® Worklight®
3. Build and deploy the adapters
4. Use the adapters in your mobile app
5. Deploy your app to the Worklight server

Installing the Tools

This section contains step-by-step instructions for installing the tools necessary to create an adapter app.

To install the tools, perform the following steps.

1. Download the AT&T Worklight project code from the following location.

   https://github.com/attdevsupport/ATT_APIPlatform_Worklight

2. Download Eclipse and Worklight studio by following the instructions at the following location.

   http://www.ibm.com/developerworks/mobile/worklight/download/

3. Modify the *eclipse.ini* file by following the instructions at the following location.

   http://pic.dhe.ibm.com/infocenter/wrklight/v5r0m5/index.jsp?topic=%2Fcom.ibm.worklight.help.doc%2Fdevenv%2Ft_next_steps.html

4. Install the Android SDK or point Eclipse to the existing location when you are installing Eclipse.

5. Install the Android ADT plugin for Eclipse by following the instructions at the following location.

   http://developer.android.com/tools/sdk/eclipse-adt.html

6. If you are doing iOS development, install XCode by following the instructions at the following location. Worklight launches iPhone/iPad environments in XCode.

   https://developer.apple.com/xcode/

Importing Projects

To import projects perform the following steps.

1. Expand the *ATTWLAdapterProject.zip* and *ATTWLApplicationProject.zip* files and Import the **ATTWLAdapterProject** and **ATTWLApplicationProject** sample app projects into Eclipse.
2. From Eclipse, click **File**, **Import**.
3. Expand the **General** folder, choose **Existing Projects Into Workspace** as shown in Figure 2
4. Click **Next**.

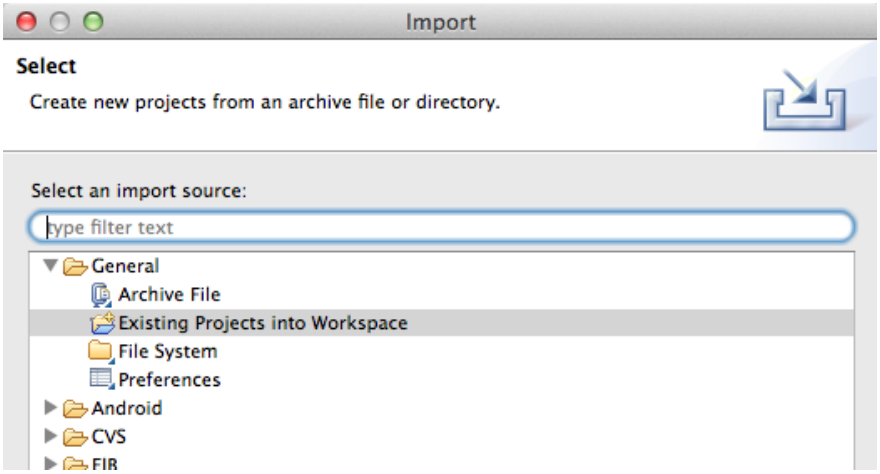

Figure 2: "Existing Projects and Workspace" location.

5. Choose **Select root directory**, and click **Browse**.
6. Select the *ATTWLAdapterProject* folder, as shown in Figure 3, and the *ATTWLApplicationProject* folder.
7. Click **Open**.
8. Click **Finish**.

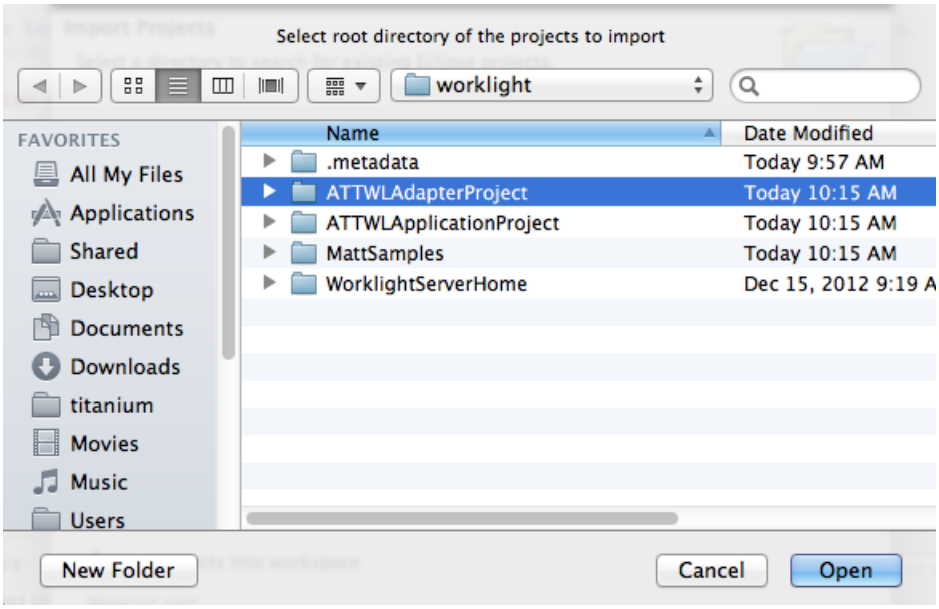

Figure 3: ATTWLAdapterProject and ATTWLApplicationProject folder location.

Registering your Application

Before you can access the AT&T Platform services, you must register your application for the services you want to access and to get an API key and secret key. These keys are necessary to call the underlying AT&T RESTful APIs that access the services.

To register an application, perform the following steps.

1. Go to https://developer.att.com and either create a new developer account or login to an existing account.
2. Select **My Apps** from the bar at the top of the page
3. Click **Setup a New Application**.

After your application is registered, you have an API key and Secret key. These keys are necessary to get your applications working with the AT&T Platform APIs.

Configuring your WorkLight Application

To configure your Worklight application, perform the following steps.

1. Open *server/conf/worklight.properties* .
2. Add the following entries.

- **apiKey**=*myApiKey*
  where myApiKey is the API key that you received when you registered your application.
- **secretKey**=*mySecretKey*
  where *mySecretKey is* the Secret key that you received when you registered your application.
- **scope**=myScope
  where *myScope is a reference to* the ATT service that is invoked by your application. All values are case-sensitive and should be separated by commas.
  - For Advertising use **ADS**
  - For Device Capabilities use **DC**
  - For Notary use **NOTARY**
  - For Payment service use **PAYMENT**
  - For SMS use **SMS**
  - For Speech To Text use **SPEECH**
  - For Text To Speech **TTS**

Building and Deploying a Worklight Adapter

To build an AT&T Worklight adapter, perform the following steps.

1. Expand the **ATTWLAdapterProject** in eclipse package explorer.
2. Expand the *adapters* folder to see the available adapters, as shown in Figure 4.

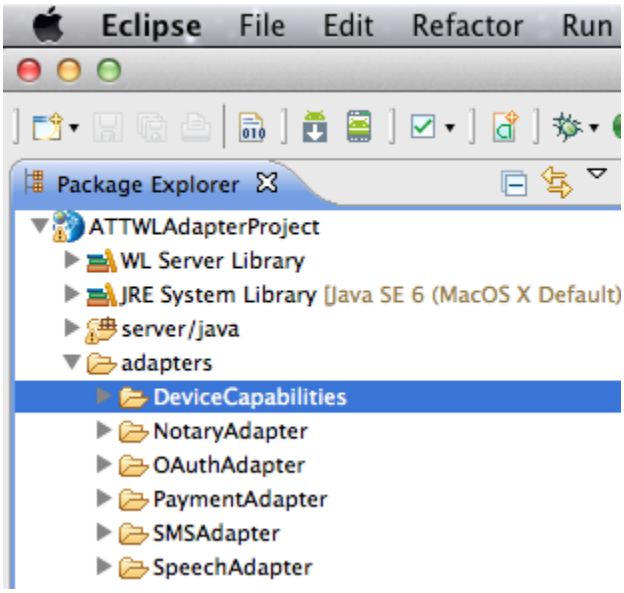

Figure 4: Available adapters.

3. Right click on an adapter folder and click **Run As -> Deploy Worklight Adapter**, as shown in Figure 5. Repeat this for each adapter folder. This builds the *.adapter* file in the */bin* folder.

   These steps deploy the Worklight adapter project. If your sample is in a separate Worklight project, the adapters need to be redeployed using the console as described in .

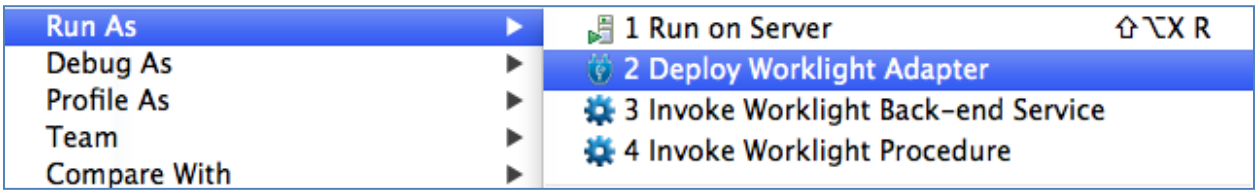

Figure 5: Building a Worklight Adapter.

Start Worklight Server

To start the Worklight server use the following procedure.

1. Navigate to the */apps/ATTWLKitchenSink/common/js/sms.js* file.
2. Specify your own shortcode in the **shortCode** variable that is declared at the top of the file.

3. Right click on the **ATTWLApplicationProject** project and select **Start Worklight Server**, as shown in Figure 6.

Figure 6: Starting the Worklight server

Deploying Adapters for The Sample Application

To deploy adapters for the sample application, perform the following steps.

1. Open the Worklight Console at the following location:

   http://localhost:8080/console

   when using IBM Worklight Developer Studio.

   The IBM Worklight Console opens in a Catalog page that enables you to work with Applications and Adapters, as shown in Figure 7.

2. Figure 7: The IBM Worklight Catalog.
3. Click **Choose File** and select an adapter from the *bin* folder of the **ATTWLAdapterProject**.
4. Click **Submit**.

   Worklight displays a message that indicates whether the deployment action succeeded or failed. **Repeat this procedure for each adapter**

   The deployed adapter is added to the catalog. Refresh the page to see it:

Figure 9: Details of deployed adapter.

Building and Deploying the Sample App

To build and deploy the sample app, perform the following steps.

1. Open the ATTWLApplicationProject folder in Project Explorer.
2. Expand the *apps* folder.
3. Select **ATTWLKitchenSink**, as shown in Figure 10.

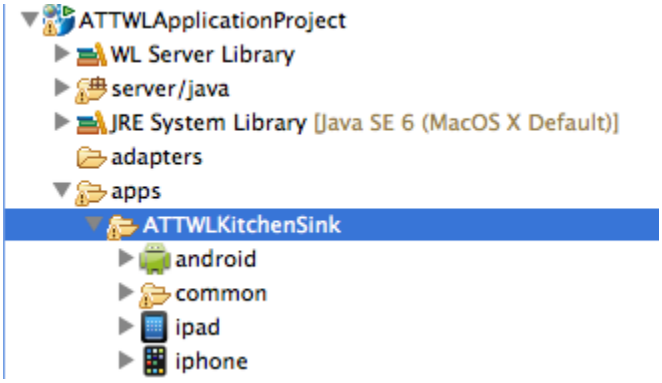

Figure 10: Selecting the ATTWLKitchenSink app.

4. Right click on **ATTWLKitchenSink** and select **Build all and deploy**, as shown in Figure 11.

Figure 11: Building and deploying applications.

5. The details of the deployed application are added to the catalog and can be accessed at the URL http://localhost:8080/console, as shown in Figure 12.

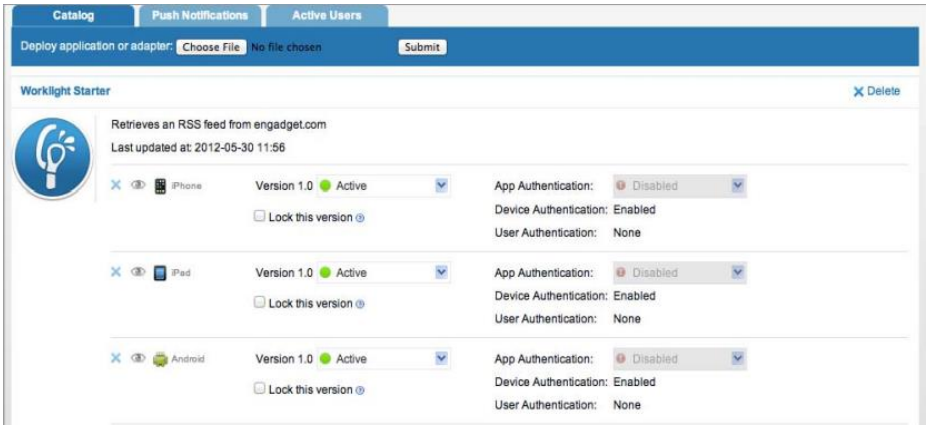

Figure 12: Details of the deployed application.

6. For the android environment, a new android project is created automatically after the application is deployed. The name of the android project created is:

   *project-name**application-name*Android

   Where *project*-name is the name of the project and *application*-name is the name of the app. For the sample application, the name of the project is:

   ATTWLApplicationProjectATTWLKitchenSinkAndroid

   This project can be executed as a normal android application. For example, **RunAs**, **Android Application**.

7. For the iPhone or iPad environment, right click on the *iphone* or *ipad* folder, and select **Run As -> Xcode project**, as shown in Figure 13. This deploys your project to the Xcode environment where it can be executed.

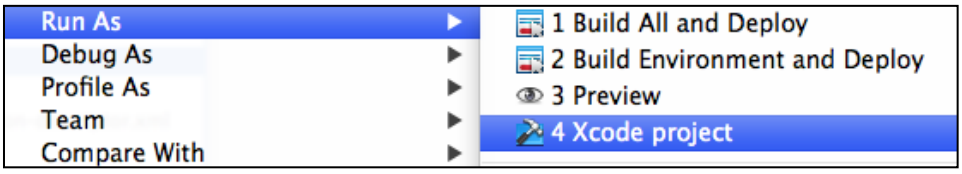

Figure 13: iPhone or iPad environment.

Using the Adapters to Create a new Worklight Application

To create a new Worklight application, follow the instructions at:

http://www.ibm.com/developerworks/mobile/library/mo-aim1206-working-with-worklight-1/index.html

1. In your application, invoke the ATT adapters from Java Script. Worklight applications can invoke adapter procedures to communicate with any data source without being subjected to same origin constraints.
2. Invoke an adapter procedure to create an invocationData object, as shown in the following code snippet.

```
var invocationData = {
     adapter : 'SMSAdapter',
     procedure : 'sendSMS',
     parameters : [{'body' : { "outboundSMSMessage": {"Message" :
"Hello All",  "Address" : "555-555-1212"}},
'contentType' : 'application/json', 'accept' : 'application/json',
'accessToken':'Bearer ' +  window.localStorage.accessToken}]
    };
```

   This object consists of a JSON block of properties:

   - **adapter** – A string that contains the name of the adapter as specified in the **<wl:adapter>** element of the adapter xml file.
   - **procedure** – Procedure name as defined in the adapter *xml* file.
   - **parameters** – An array of parameters passed on to the remote procedure.

3. Define the failure and success behavior in an options object.

```
var options = {
     onSuccess : yourSuccessCallback,
     onFailure : yourFailureCallback,
     InvacationContext {}
    };
```

   This object consists of a JSON block of properties:

   - **onSuccess** - The function to be invoked on successful completion of the asynchronous call.
   - **onFailure** – The function to be invoked on failure.
   - **invocationContext** – Optional parameter. An object that is returned to the success and failure handlers.

4. Invoke the procedure using the **invocationData** object and options object.

```
WL.Client.invoke.Procedure(invocationData, options)
```

   To learn more about installing Worklight adapters, see the guide at:

   http://public.dhe.ibm.com/ibmdl/export/pub/software/mobile-solutions/worklight/docs/wl_5_0_admin.pdf