Communicating with HTTP back-end systems

AKHIL S
akhil.new10@gmail.com

04 Mar 2016

Worklight Adapter

> Adapters are the server-side code of applications that are deployed on and serviced by IBM Worklight Foundation.

> Adapters connect to enterprise applications (otherwise referred to as back-end systems), deliver data to and from mobile applications, and perform any necessary server-side logic on this data.

> With IBM Worklight Foundation, you can create and configure adapters manually.

benifits...

> **Transparency**: Data retrieved from back-end applications is exposed in a uniform manner, so that application developers can access data uniformly, regardless of its source, format, and protocol.

> **Security**: Worklight adapters use flexible authentication facilities to create connections with back-end systems. Adapters offer control over the identity of the user with whom the connection is made.

Adapter framework

> The adapter framework mediates between the mobile apps and the back-end services. A typical flow is depicted in the following diagram.
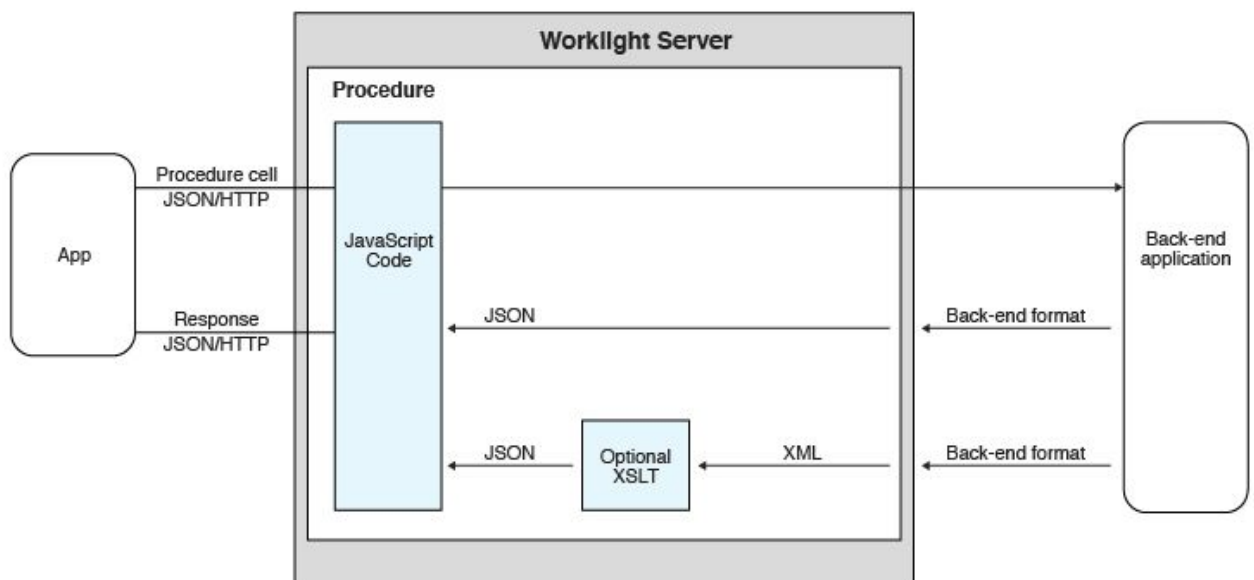
An adapter exposes a set of services, called procedures. Mobile apps invoke procedures by issuing Ajax requests.

The procedure retrieves information from the back-end application.

The back-end application then returns data in some format.

The JavaScript implementation of the procedure receives the JSON data, performs any additional processing, and returns it to the calling app.

Types...

> The HTTP adapter

> The SQL adapter

> The Cast Iron adapter

> The JMS adapter

HTTP adapter

> Can read structured HTTP sources, for example RSS feeds.

> Allows sending HTTP request and retrieves data from the response headers and body.

> Is easily customizable with simple server-side JavaScript.

> Enables optional server-side filtering.

> Retrieved data can be in XML, HTML, JSON, or plain text formats.

Working ???

It uses XML to define the adapter properties and procedures.

It uses JavaScript to create procedures.

It uses XSL to filter received records and fields.(optional)

Creating the adapter

● In Worklight Studio, create a Worklight Adapter.

○ Choose the HTTP Adapter type.

○ A standard HTTP adapter structure is created:

contains...

> HTTPAdapter.xml

> HTTPAdapter-impl.js

> filtered.xsl

XML file: > Set the protocol to HTTP or HTTPS.

> Set the HTTP domain to domain part of HTTP URL.

> Set the TCP Port.

> Declare the required procedures.

```xml
<wl:adapter name="Notesadapter"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:wl="http://www.ibm.com/mfp/integration"
        xmlns:http="http://www.ibm.com/mfp/integration/http">

        <displayName>Notesadapter</displayName>
        <description>Notesadapter</description>
        <connectivity>
                <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
                        <protocol>http</protocol>
                        <domain>www.engadget.com</domain>
                        <port>80</port>
                        <connectionTimeoutInMilliseconds>30000</connectionTimeoutInMilliseconds>
                        <socketTimeoutInMilliseconds>30000</socketTimeoutInMilliseconds>
                        <maxConcurrentConnectionsPerNode>50</maxConcurrentConnectionsPerNode>
                        <!-- Following properties used by adapter's key manager for choosing specific
certificate from key store
                        <sslCertificateAlias></sslCertificateAlias>
                        <sslCertificatePassword></sslCertificatePassword>
                        -->
                </connectionPolicy>
        </connectivity>
        <procedure name="getStories"/>
    <procedure name="getStoriesFiltered"/>
</wl:adapter>
```

JS file: Procedures are implemented in the adapter JavaScript file.

> The service URL is used for procedure invocation.

> The procedure name in the JavaScript file must be the same   .   as in the XML file.

> The required invocation parameters are method, path, and returnedContentType.

> To invoke a HTTP request, use the WL.Server.invokeHttp method.

> Provide an input parameters object.

> specify: – The HTTP method: GET or POST.

> specify: – The returned content type: XML, JSON, HTML, or plain.

 > specify: – The service path.

 > XSL transformation: It can be used to filter received data.

```
function getStories() {

        var input = {
                method : 'get',
                returnedContentType : 'xml',
                path : "rss.xml"
            };

            return WL.Server.invokeHttp(input);
        }

function getStoriesFiltered() {

        var input = {
                method : 'get',
                returnedContentType : 'xml',
                path : "rss.xml",
                transformation : {
                        type : 'xslFile',
                        xslFile : 'filtered.xsl'
                }
            };

            return WL.Server.invokeHttp(input);
        }
```

#XSL file: > It is used to filter received records and fields.

```
<xsl:stylesheet version="1.0"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:h="http://www.w3.org/1999/xhtml"
                xmlns:dc="http://purl.org/dc/elements/1.1/" >
    <xsl:output method="text"/>

    <xsl:template match="/">
        {
                'Items': [
                        <xsl:for-each select="//item">
                        {
                                'title': '<xsl:value-of select="title"/>',
                                'creator': '<xsl:value-of select="dc:creator"/>',
                                'link': '<xsl:value-of select="link"/>',
                                'pubDate': '<xsl:value-of select="pubDate"/>'
                        },
                        </xsl:for-each>
                ]
        }
    </xsl:template>
</xsl:stylesheet>
```

Invoking adapter procedures from client applications...

> The first step to invoke an adapter procedure is to create an **invocationData object:**

It consists of a JSON block of properties.

> **adapter** (mandatory) A string that contains the name of the adapter as specified in the adapter element.

> **procedure** (mandatory) Procedure name as defined in the XML file.

> **parameters** (mandatory) An array of parameters that are passed to the back-end JavaScript procedure. Leave empty if no parameters are required.

> The second step is to **define failure and success behaviors** in the options object.

WL.Client.invokeProcedure({..});

It consists

> **onSuccess:** The function to be invoked on successful completion of the asynchronous call

> **onFailure:** The function to be invoked on failure.(Includes both server-side errors and client-side)

```javascript
var invocationData={
                            adapter : 'Notesadapter',
                            procedure : 'getStories',
                            procedure : 'getStoriesFiltered'

                    };
  invocationOptions={
                            onSuccess : loadFeedsSuccess,
                            onFailure: loadFeedsFailure,
                    };
WL.Client.invokeProcedure(invocationData,invocationOptions);
    function loadFeedsSuccess(result){

    WL.Logger.debug("Feed retrive success");
    if (result.invocationResult.Items.length>0) {

            displayFeeds(result.invocationResult.Items);
    }
    else{

            loadFeedsFailure();}
    }
    function loadFeedsFailure(result){
            WL.Logger.debug("Feed retrive failed");
    }
```

Invocation result:

Application is now online.

response [/NotesandPasswords/apps/services/api/NotesandPasswords/common/query] success: /*-secure-
{"Items":[{"pubDate":"Fri, 04 Mar 2016 01:42:00 -0500","title":"Sony's Future Lab will ask users for input on early
prototypes","link":"http:\/\/www.engadget.com\/2016\/03\/04\/sony-future-lab-program\/","creator":"Mariella Moon"},{"pubDate":"Fri,
Mar 2016 00:36:00 -0500","title":"Kendrick Lamar releases a new album in response to tweets","link":"http:\/\/www.engadget.com\/2016
\/03\/04\/kendrick-lamar-releases-a-new-album-in-response-to-tweets\/","creator":"Richard Lawler"},{"pubDate":"Thu, 03 Mar 2016
22:08:00 -0500","title":"The Hubble has seen farther back in time than ever before","link":"http:\/\/www.engadget.com\/2016\/03\/03
\/the-hubble-has-seen-farther-back-in-time-than-ever-before\/","creator":"Andrew Tarantola"},{"pubDate":"Thu, 03 Mar 2016 21:59:00
-0500","title":"AT&T is bringing internet-style targeted advertising to TV","link":"http:\/\/www.engadget.com\/2016\/03\/03\/atandt-
is-bringing-internet-style-targeted-advertising-to-tv\/","creator":"Timothy J. Seppala"},{"pubDate":"Thu, 03 Mar 2016 21:29:00
-0500","title":"Instagram blocks users from linking to Telegram and Snapchat","link":"http:\/\/www.engadget.com\/2016\/03\/03
\/instagram-blocks-telegram-and-snapchat-deeplinks\/","creator":"Mariella Moon"},{"pubDate":"Thu, 03 Mar 2016 20:39:00
-0500","title":"DARPA unveils its next VTOL aircraft concept","link":"http:\/\/www.engadget.com\/2016\/03\/03\/darpa-unveils-its-nex
vtol-aircraft-concept\/","creator":"Andrew Tarantola"},{"pubDate":"Thu, 03 Mar 2016 19:54:00 -0500","title":"Nintendo Direct roundup
'Mario Maker,' 'Splatoon' and more","link":"http:\/\/www.engadget.com\/2016\/03\/03\/nintendo-direct-roundup-mario-maker-splatoon-

Feed retrive success