Java HTTP Adapter

Overview

Java adapters provide free reign over connectivity to your backend. It is therefore your responsibility to ensure best practices regarding performance and other implementation details.
This tutorial covers an example of a Java adapter that connects to an RSS feed by using a Java `HttpClient`.

Prerequisite: Make sure to read the Java Adapters (../) tutorial first.

RSSAdapterApplication

`RSSAdapterApplication` extends `MFPJAXRSApplication` and is a good place to trigger any initialization required by your application.

```
@Override
protected void init() throws Exception {
    RSSAdapterResource.init();
    logger.info("Adapter initialized!");
}
```

RSSAdapterResource

`RSSAdapterResource` is where we handle the requests to your adapter.

```
@Path("/")
public class RSSAdapterResource {
}
```

`@Path("/")` means that the resources will be available at the URL `http(s)://host:port/ProjectName/adapters/AdapterName/`.

HTTP Client

RSSAdapterResource

```
private static CloseableHttpClient client;
private static HttpHost host;

public static void init() {
    client = HttpClients.createDefault();
    host = new HttpHost("developer.ibm.com");
}
```

Because every request to your resource will create a new instance of `RSSAdapterResource`, it is important to reuse objects that may impact performance. In this example we made the Http client a `static` object and initialized it in a static `init()` method, which gets called by the `init()` of `RSSAdapterApplication` as described above.

Procedure resource

RSSAdapterResource

```
@GET
@Produces("application/json")
public void get(@Context HttpServletResponse response, @QueryParam("tag") String tag)
        throws ClientProtocolException, IOException, IllegalStateException, SAXException {
    if(tag!=null && !tag.isEmpty()){
        execute(new HttpGet("/mobilefirstplatform/tag/"+ tag +"/feed"), response);
    }
    else{
        execute(new HttpGet("/mobilefirstplatform/feed"), response);
    }

}
```

Our adapter exposes just one resource URL which allows to retrieve the RSS feed from the backend service.

- `@GET` means that this procedure only responds to `HTTP GET` requests.
- `@Produces("application/json")` specifies the Content Type of the response to send back. We chose to send the response as a `JSON` object to make it easier on the client-side.
- `@Context HttpServletResponse response` will be used to write to the response output stream. This enables us more granularity than returning a simple string.
- `@QueryParam("tag")` String tag enables the procedure to receive a parameter. The choice of `QueryParam` means the parameter is to be passed in the query (`/RSSAdapter/?tag=MobileFirst_Platform`). Other options include `@PathParam`, `@HeaderParam`, `@CookieParam`, `@FormParam`, etc.
- `throws ClientProtocolException, ...` means we are forwarding any exception back to the client. The client code is responsible for handling potential exceptions which will be received as `HTTP 500` errors. Another solution (more likely in production code) is to handle exceptions in your server Java code and decide what to send to the client based on the exact error.
- `execute(new HttpGet("/mobilefirstplatform/feed"), response)`. The actual HTTP request to the backend service is handled by another method defined later.

Depending if you pass a `tag` parameter, `execute` will retrieve a different build a different path and retrieve a different RSS file.

execute()

RSSAdapterResource

```
public void execute(HttpUriRequest req, HttpServletResponse resultResponse)
        throws ClientProtocolException, IOException,
        IllegalStateException, SAXException {
    HttpResponse RSSResponse = client.execute(host, req);
    ServletOutputStream os = resultResponse.getOutputStream();
    if (RSSResponse.getStatusLine().getStatusCode() == HttpStatus.SC_OK){
        resultResponse.addHeader("Content-Type", "application/json");
        String json = XML.toJson(RSSResponse.getEntity().getContent());
        os.write(json.getBytes(Charset.forName("UTF-8")));

    }else{
        resultResponse.setStatus(RSSResponse.getStatusLine().getStatusCode());
        RSSResponse.getEntity().getContent().close();
        os.write(RSSResponse.getStatusLine().getReasonPhrase().getBytes());
    }
    os.flush();
    os.close();
}
```

- `HttpResponse RSSResponse = client.execute(host, req)`. We use our static HTTP client to execute the HTTP request and store the response.
- `ServletOutputStream os = resultResponse.getOutputStream()`. This is the output stream to write a response to the client.
- `resultResponse.addHeader("Content-Type", "application/json")`. As mentioned before, we chose to send the response as JSON.
- `String json = XML.toJson(RSSResponse.getEntity().getContent())`. We used `org.apache.wink.json4j.utils.XML` to convert the XML RSS to a JSON string.
- `os.write(json.getBytes(Charset.forName("UTF-8")))` the resulting JSON string is written to the output stream.

The output stream is then `flush`ed and `close`d.

If `RSSResponse` is not `200 OK`, we write the status code and reason in the response instead.

Results

The adapter should return the RSS feed converted to JSON.

```
{
  "rss": {
    "channel": {
      "description": "Develop, test, manage, and secure your mobile web, native and hybrid apps",
      "generator": "http:\/\/wordpress.org\/?v=4.2.4",
      "item": [
        {
          "category": [
            "Mobile",
            "android",
            "Mobile Quality Assurance",
            "mobile_development",
            "mobilefirst",
            "xamarin"
          ],
          "commentRss": "https:\/\/developer.ibm.com\/mobilefirstplatform\/2015\/09\/01\/integrating-mqa-int
```

o-xamarin-android-app\/feed\/",
            "comments": [
              "https:\/\/developer.ibm.com\/mobilefirstplatform\/2015\/09\/01\/integrating-mqa-into-xamarin-android-app\/#comments",
              "0"
            ],
            "creator": "Vidyasagar MSC",
            "description": "<p>The post <a rel=\"nofollow\" href=\"https:\/\/developer.ibm.com\/mobilefirstplatfo
rm\/2015\/09\/01\/integrating-mqa-into-xamarin-android-app\/\">Integrating MQA into Xamarin.Android app<\
/a> appeared first on <a rel=\"nofollow\" href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\">IBM Mobile
First Platform<\/a>.<\/p>",
            "encoded": "<p>It all startedÂ when I received an email seeking help on using MQA or to be more
precise integrating MQA into Xamarin based android app. Before jumping into addressing the problem, let&#
8217;s define MQA.<\/p>\n<h4>What is MQA?<\/h4>\n<p>MQA stands for &#8220;Mobile Quality Assuranc
e&#8221; and is part of the IBM MobileFirst Platform.<\/p>\n<blockquote><p><em><span style=\"line-height
: 1.5\">IBM MQA provides line of business professionals and development teams with insightful and streamli
ned quality feedback and metrics from both pre-production and production, enabling them to prioritize and ta
ke action to support a dynamic mobile app strategy.<\/span><\/em><\/p><\/blockquote>\n<p>The Features o
f MQA are<\/p>\n<div style=\"width: 1058px\" class=\"wp-caption aligncenter\"><a href=\"http:\/\/vidyasagarm
sc.com\/wp-content\/uploads\/2015\/09\/MQA1.png\"><img class=\"size-full wp-image-65\" src=\"http:\/\/vidya
sagarmsc.com\/wp-content\/uploads\/2015\/09\/MQA1.png\" alt=\"Features of Mobile Quality Assurance.\" wi
dth=\"1048\" height=\"350\" \/><\/a><p class=\"wp-caption-text\">Features of Mobile Quality Assurance.<\/p>
<\/div>\n<p><em><strong>Note<\/strong><\/em>: To understand more about MQA, visitÂ <a href=\"http:\/\/w
ww-03.ibm.com\/software\/products\/en\/ibm-mobilefirst-platform-quality-assurance\">IBM Mobile Quality Ass
urance<\/a><\/p>\n<p>So, by now we should be good with the first part of our blog title that is MQA. So, the
next question is<\/p>\n<h4>What is Xamarin.Android?<\/h4>\n<p>Xamarin is a platform to create nativeÂ iO
S, Android, Mac and Windows apps in C#.Â Xamarin.Android allows us to create native Android applications
using the same UI controls we would in Java, except with the flexibility and elegance of a modern language (
C#).<\/p>\n<p>As we are good with the definitions, let&#8217;s address the problem.<\/p>\n<p><strong>Wh
at&#8217;s the problem in integrating MQA into Xamarin Android app?<\/strong><\/p>\n<p>At the time of thi
s blog post, the available MQA SDKs are iOS native SDK, Android native SDK and Javascript Â SDK.<\/p>\n
<p>So, we have to find a workaround to address this use-case. The initial step is to download the Android M
QA SDK and see what&#8217;s provided. you can download it from <a href=\"http:\/\/www-01.ibm.com\/sup
port\/knowledgecenter\/#!\/SSJML5_6.0.0\/com.ibm.mqa.uau.saas.doc\/topics\/c_AndroidSDKsForDownload
.html\">here<\/a>. Once successfully downloaded and unzipped, we should see a jar file namely <strong><e
m>MQA-Android-library-&lt;version number&gt;.jar<\/em>Â <\/strong>under lib folder<strong>.<\/strong><\/
p>\n<div style=\"width: 634px\" class=\"wp-caption aligncenter\"><a href=\"http:\/\/vidyasagarmsc.com\/wp-c
ontent\/uploads\/2015\/09\/MQA2.png\"><img class=\"size-full wp-image-70\" src=\"http:\/\/vidyasagarmsc.co
m\/wp-content\/uploads\/2015\/09\/MQA2.png\" alt=\"MQA Android SDK \" width=\"624\" height=\"440\" \/><\/
a><p class=\"wp-caption-text\">MQA Android SDK<\/p><\/div>\n<p>As Xamarin is C# based, What can we
do with this jar file?<\/p>\n<p>We haveÂ <strong>Xamarin bindings<\/strong> to our rescue, which helps usi
ng in consuming .JARs from C#.<\/p>\n<p><strong><em>Note<\/em>:<\/strong> Steps to consume MQA An
droid JAR in a Xamarin.Android app is mentionedÂ <a href=\"https:\/\/developer.xamarin.com\/guides\/androi
d\/advanced_topics\/java_integration_overview\/binding_a_java_library_(.jar)\/\">here<\/a><\/p>\n<div style=
\"width: 257px\" class=\"wp-caption aligncenter\"><a href=\"http:\/\/vidyasagarmsc.com\/wp-content\/uploads\
/2015\/09\/MQA31.png\"><img class=\"wp-image-72 size-full\" src=\"http:\/\/vidyasagarmsc.com\/wp-content\/
uploads\/2015\/09\/MQA31.png\" alt=\"\" width=\"247\" height=\"303\" \/><\/a><p class=\"wp-caption-text\">X
amarin binding project with MQA Android .JAR file<\/p><\/div>\n<p>The files of our interest here are <strong
>MQA-Android-library-2.7.4.jar<\/strong> (Version number may vary) and <strong>Metadata.xml.<\/strong><
\/p>\n<ul>\n<li>MQA-Android-library-2.7.4.jar file will have all the MQA related classes and methods
required for us to start an Android MQA session.<\/li>\n<li>Metadata.xml- <em>Allows changes to be made t
o the final API, such as changing the namespace of the generated binding.<\/em><\/li>\n<\/ul>\n<p>Based o
n the errors thrown while building the project, Metadata.xml in my case looks like this<\/p>\n<pre class=\"bru
sh: xml; title: ; notranslate\">&lt;metadata&gt;\n &lt;!--\n This sample removes the class: android.support.v4.
content.AsyncTaskLoader.LoadTask:\n &lt;remove-node path=&quot;\/api\/package[@name='android.suppo
rt.v4.content']\/class[@name='AsyncTaskLoader.LoadTask']&quot; \/&gt;\n \n This sample removes the met
hod: android.support.v4.content.CursorLoader.loadInBackground:\n &lt;remove-node path=&quot;\/api\/pack

age[@name='android.support.v4.content']\/class[@name='CursorLoader']\/method[@name='loadInBackgrou
nd']&quot; \/&gt;\n --&gt;\n\n &lt;remove-node path=&quot;\/api\/package[@name='ext.com.google.inject.spi
']\/class[@name='InjectionPoint.Factory.1']&quot;\/&gt;\n &lt;remove-node path=&quot;\/api\/package[@nam
e='ext.com.google.inject.spi']\/class[@name='InjectionPoint.Factory.2']&quot;\/&gt;\n &lt;remove-node path=
&quot;\/api\/package[@name='com.applause.android.log']\/interface[@name='LoggerInterface']&quot;\/&gt;\
n &lt;remove-node path=&quot;\/api\/package[@name='ext.com.google.inject.internal']&quot;\/&gt;\n &lt;re
move-node path=&quot;\/api\/package[@name='ext.com.google.inject.matcher']&quot;\/&gt;\n &lt;remove-no
de path=&quot;\/api\/package[@name='com.applause.android.util']\/class[@name='AbstractRequest']&quot;\
/&gt;\n &lt;remove-node path=&quot;\/api\/package[@name='ext.com.google.inject.spi']\/class[@name='Ele
ments.RecordingBinder']\/method[@name='bind' and count(parameter)=1 and parameter[1][@type='ext.com.
google.inject.Key']]&quot;\/&gt;\n\n&lt;attr path=&quot;\/api\/package[@name='com.applause.android.messa
ges']\/class[@name='Message']\/field[@name='message']&quot; name=&quot;managedName&quot;&gt;Mes
sage1&lt;\/attr&gt;\n&lt;attr path=&quot;\/api\/package[@name='com.applause.android.log']&quot; name=&q
uot;managedName&quot;&gt;log&lt;\/attr&gt;\n&lt;\/metadata&gt;\n\n<\/pre>\n<p>Once all the errors are fixe
d and your binding project builds successfully, add a new Xamarin Android project (if you haven&#8217;t ad
ded yet). Now, add MQA binding project reference in our Xamarin android app. <em><strong>Note:<\/strong
><\/em> Both your binding project and Xamarin.Android project should be of same <strong>target framework
.Â <\/strong>You can verify this by right clicking on your project -&gt; Options -&gt; General.<\/p>\n<div id=\"
attachment_83\" style=\"width: 270px\" class=\"wp-caption aligncenter\"><a href=\"http:\/\/vidyasagarmsc.co
m\/wp-content\/uploads\/2015\/09\/MQA5.png\"><img class=\"size-full wp-image-83\" src=\"http:\/\/vidyasaga
rmsc.com\/wp-content\/uploads\/2015\/09\/MQA5.png\" alt=\"Xamarin Android project with added reference t
o MQA\" width=\"260\" height=\"652\" \/><\/a><p class=\"wp-caption-text\">Xamarin Android project with add
ed reference to MQA<\/p><\/div>\n<p>Now, let&#8217;s start MQA android session in our Count.Android ap
p. Before doing this, we should create a MQA service on IBM Bluemix. You can follow the instructions menti
oned atÂ <a href=\"https:\/\/www.ng.bluemix.net\/docs\/#services\/MobileQualityAssurance\/index.html#Mobil
eQualityAssurance\">Getting started with Mobile Quality Assurance- Bluemix<\/a>Â or watch this video.<\/p
>\n<p><span class='embed-youtube' style='text-align:center; display: block;'><iframe class='youtube-player'
type='text\/html' width='980' height='582' src='https:\/\/www.youtube.com\/embed\/zHRfGatcKPM?version=3
&#038;rel=1&#038;fs=1&#038;showsearch=0&#038;showinfo=1&#038;iv_load_policy=1&#038;wmode=tran
sparent' frameborder='0' allowfullscreen='true'><\/iframe><\/span><\/p>\n<p>Starting aÂ <span class=\"ph\"
><span id=\"d6087e24\" class=\"ph\">Mobile Quality Assurance<\/span><\/span>Â session with the Android
SDK entails three steps. First, build a configuration to define howÂ <span class=\"ph\"><span id=\"d6087e24
-d6083e11a1310\" class=\"ph\">Mobile Quality Assurance<\/span><\/span>Â works with your app. Second,
start the session itself. Third, add tracking to your activities. Open <strong>MainActivity.cs<\/strong> file (An
droid Project) and paste the code provided below<\/p>\n<pre class=\"brush: csharp; title: ; notranslate\">usin
g System;\n\nusing Android.App;\nusing Android.Content;\nusing Android.Runtime;\nusing Android.Views;\n
using Android.Widget;\nusing Android.OS;\n\/\/MQA references\nusing Com.Ibm.Mqa.Config;\nusing Com.Ib
m.Mqa;\n\n\nnamespace Count.Android\n{\n\t[Activity (Label = &quot;Count.Android&quot;, MainLauncher =
true, Icon = &quot;@drawable\/icon&quot;)]\n\tpublic class MainActivity : Activity\n\t{\n\t\tint count = 1;\n\t\t\/\/
Use your own generated APP KEY\n\t\tconst string APP_KEY=&quot;1g59b7d884f9fdf5426162e5cb1f87a70
0648bce4fg0g1g379e0d3a&quot;;\n\t\tprotected override void OnCreate (Bundle bundle)\n\t\t{\n\t\t\tbase.On
Create (bundle);\n\t\t\t\/\/MQA Android session configuration \n\t\t\tConfiguration configuration = new Configu
ration.Builder(this)\n\t\t\t\t.WithAPIKey(APP_KEY) \/\/Provides the quality assurance application APP_KEY\n
\t\t\t\t.WithMode(MQA.Mode.Qa) \/\/Selects the quality assurance application mode\n\t\t\t\t.WithReportOnSh
akeEnabled(true) \/\/Enables shake report trigger\n\t\t\t\t.WithDefaultUser(&quot;default_user@email.com&q
uot;) \/\/Sets a default user and user selection\n\t\t\t\t.Build();\n\n\t\t\t\/\/Starting MQA Android Session\n\t\t\tM
QA.StartNewSession (this, configuration);\n\t\t\t\/\/ Set our view from the &quot;main&quot; layout resource\n
\t\t\tSetContentView (Resource.Layout.Main);\n\n\t\t\t\/\/ Get our button from the layout resource,\n\t\t\t\/\/ an
d attach an event to it\n\t\t\tButton button = FindViewById&lt;Button&gt; (Resource.Id.myButton);\n\t\t\t\n\t\t\tb
utton.Click += delegate {\n\t\t\t\tbutton.Text = string.Format (&quot;{0} clicks!&quot;, count++);\n\t\t\t};\n\t\t}\n\
t}\n}\n\n\n\n<\/pre>\n<p>Now, MQA is integrated into Xamarin.Android app and we are good to go.<\/p>\n<p
>What we have implemented above is just a drop in the Ocean of MQA, to know more about MQA and its fe
atures &#8211; VisitÂ <a href=\"http:\/\/www-01.ibm.com\/support\/knowledgecenter\/?lang=en#!\/SSJML5_6
.0.0\/com.ibm.mqa.uau.saas.doc\/mqa600saas_welcome.html\" target=\"_blank\">MQA Knowledge Centre<\
/a><\/p>\n<p>Happy Coding !!!<\/p>\n<p>The post <a rel=\"nofollow\" href=\"https:\/\/developer.ibm.com\/mo
bilefirstplatform\/2015\/09\/01\/integrating-mqa-into-xamarin-android-app\/\">Integrating MQA into Xamarin.A
ndroid app<\/a> appeared first on <a rel=\"nofollow\" href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\"

```
ndroid app<\/a> appeared first on <a rel=\"nofollow\" href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\
>IBM MobileFirst Platform<\/a>.<\/p>",
          "guid": {
            "content": "https:\/\/developer.ibm.com\/mobilefirstplatform\/?p=16964",
            "isPermaLink": "false"
          },
          "link": "https:\/\/developer.ibm.com\/mobilefirstplatform\/2015\/09\/01\/integrating-mqa-into-xamarin
-android-app\/",
          "pubDate": "Tue, 01 Sep 2015 20:27:07 +0000",
          "title": "Integrating MQA into Xamarin.Android app"
        },
        {
          "category": [
            "Uncategorized",
            "MobileFirst_Platform"
          ],
          "commentRss": "https:\/\/developer.ibm.com\/mobilefirstplatform\/2015\/08\/19\/try-on-bluemix-and
-buy-mfp\/feed\/",
          "comments": [
            "https:\/\/developer.ibm.com\/mobilefirstplatform\/2015\/08\/19\/try-on-bluemix-and-buy-mfp\/#co
mments",
            "0"
          ],
          "creator": "ChethanKumar",
          "description": "<p>The post <a rel=\"nofollow\" href=\"https:\/\/developer.ibm.com\/mobilefirstplatfo
rm\/2015\/08\/19\/try-on-bluemix-and-buy-mfp\/\">Try on Bluemix and migrate to on-prem MobileFirst Platfor
m<\/a> appeared first on <a rel=\"nofollow\" href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\">IBM Mo
bileFirst Platform<\/a>.<\/p>",
          "encoded": "<p>Contributed By : Chethan Kumar SN (chethankumar.sn@in.ibm.com) and Vittal P
ai (vittalpai@in.ibm.com)<\/p>\n<p>With the release of MobileFirst Platform v7.1, one can now migrate any e
xisting iOS app built for MobileServices on Bluemix to MobileFirst Platform with just a handful of simple steps
.<\/p>\n<p>To elucidate the process, lets look at how to migrate a simple Bluemix iOS app.<\/p>\n<p>To mi
grate an existing iOS app built for MobileServices on Bluemix to run on MobileFirst Platform, follow the steps
below.<\/p>\n<ul>\n<li><a href=\"#migrateexisting\">Existing Bluemix Server Application<\/a><\/li>\n<li><a h
ref=\"#migrateblu\">Existing Bluemix Client Application<\/a><\/li>\n<li><a href=\"#configureclient\">Migration
of Client Application<\/a><\/li>\n<li><a href=\"#migratemfp\">Migration of JAX-RS Application to JAVA Adapt
er<\/a><\/li>\n<li><a href=\"#configoauth\">Configuring Custom-OAuth<\/a><\/li>\n<li><a href=\"#configurep
ush\">Configuring Push Capability<\/a><\/li>\n<li><a href=\"#sample\">Sample and Source Code<\/a><\/li>\
n<\/ul>\n<h2 id=\"migrateexisting\">Existing Bluemix Server Application<\/h2>\n<p>The Bluemix app has the
following functionality:<\/p>\n<ul>\n<li>On the client side, the application stores a list of items and provides a
way to add more items to the list. Each item can able to store Name, Store, Price and image of the product.
The App&#8217;s are protected by Custom Authenticator via AMA security service provided by bluemix.<\/li>
\n<li>On the server side, the App contains a JAX-RS class to store and manipulate the data. It also contains
the server side AMA security implementation.<\/li>\n<\/ul>\n<p>On BlueMix we have application with the foll
owing configuration:<\/p>\n<ul>\n<li>Liberty Runtime : which used to run JAX-RS application on Bluemix<\/li
>\n<li>Advance Mobile Access service : which gives mobile application security and monitoring functionality
<\/li>\n<li>Push Service for iOS 8 : which provides the capability to use iOS Push features<\/li>\n<\/ul>\n<h3
> Liberty Runtime <\/h3>\n<ul>\n<li>Liberty contains two projects with JAX-RS service (i.e Custom-oauth-ja
va for Custom Authentication and LocalstoreAdapter for storing items). The service include the protected res
ource and the custom identity provider code. The liberty server is configured with TAI.\n<\/li>\n<li>Trust Asso
ciation Interface (TAI) is a service provider API that enables the integration of third-party security services wit
h a Liberty profile server. For more info on TAI : <a href=\"http:\/\/www-01.ibm.com\/support\/knowledgecente
r\/was_beta_liberty\/com.ibm.websphere.wlp.nd.multiplatform.doc\/ae\/twlp_dev_custom_tai.html\" target=\"_
blank\">click here<\/a>\n<\/li>\n<li>The custom identity provider authenticates a user by sending challenges
to the client. However, custom identity providers do not communicate directly with clients. They send challen
ges and receive responses to the challenges by means of the Advanced Mobile Access service. When a cus
tom identity provider successfully authenticates the user, it provides the user identity information to Advance
[illegible]
```

d Mobile Access. For more information on custom authentication refer bluemix documentation : <a href=\"http
s:\/\/www.ng.bluemix.net\/docs\/services\/mobileaccess\/security\/id_provs\/index-gentopic2.html#custom_id
_prov\" target=\"_blank\">click here<\/a>\n<p>The custom identity provider code is defined by two http API:<
\/p>\n<pre class=\"brush: plain; title: ; notranslate\">\/startAutorization<\/pre>\n<p> and\n<pre class=\"brush:
plain; title: ; notranslate\">\/handleChallengeAnswer<\/pre>\n<pre class=\"brush: java; title: ; notranslate\">
@POST\n\t@Consumes (&quot;application\/json&quot;)\n\t@Path(&quot;\/{tenantId}\/customAuthRealm_3\/
startAuthorization&quot;)\n\t@Produces(MediaType.APPLICATION_JSON)\n\tpublic JSONObject startAutho
rization(String payload,\n\t\t\t@PathParam(&quot;tenantId&quot;) String deviceId,\n\t\t\t@PathParam(&quot;r
ealmName&quot;) String realmName) throws Exception {\n\t\tJSONObject returnJson = (JSONObject) JSON
.parse(CHALLENGE_JSON);\n\t\treturn returnJson;\n\t}\n\t\n\t@POST\n\t@Consumes (&quot;application\/js
on&quot;)\n\t@Path(&quot;\/{tenantId}\/customAuthRealm_3\/handleChallengeAnswer&quot;)\n\t@Produces
(MediaType.APPLICATION_JSON)\n\tpublic JSONObject handleChllengeAnswer(String payload,\n\t\t\t@Pa
thParam(&quot;tenantId&quot;) String deviceId,\n\t\t\t@PathParam(&quot;realmName&quot;) String realmN
ame) throws Exception {\n\t\t\n\t\tJSONObject userStoreJson = (JSONObject) JSON.parse(USER_STORE_
JSON);\n\t\tJSONObject failedResponseJson = (JSONObject) JSON.parse(FAILURE_JSON);\n\t\t\n\t\tif(payl
oad == null || payload.isEmpty()) {\n\t\t\treturn failedResponseJson;\n\t\t}\n\t\tJSONObject payloadJson = (JS
ONObject) JSON.parse(payload);\n\t\tJSONObject challengeAnswer = (JSONObject) payloadJson.get(&quot
;challengeAnswer&quot;);\n\t\t\n\t\tif (challengeAnswer == null) {\n\t\t\treturn failedResponseJson;\n\t\t}\n\t\t\
n\t\tString userName = (String) challengeAnswer.get(&quot;userName&quot;);\n\t\tString password = (String)
challengeAnswer.get(&quot;password&quot;);\n\t\t\n\t\tif (userName == null || userName.isEmpty() || passwo
rd == null || password.isEmpty()) {\n\t\t\treturn failedResponseJson;\n\t\t}\n\t\t\n\t\tif (userStoreJson.containsK
ey(userName)) {\t\n\t\t\tJSONObject userInfoJson = (JSONObject) userStoreJson.get(userName);\n\t\t\tStrin
g userPassword = (String) userInfoJson.get(&quot;password&quot;);\n\t\t\tString userDisplayName = (String)
userInfoJson.get(&quot;displayName&quot;);\n\t\t\t\n\t\t\tif (password.equals(userPassword)) {\n\t\t\t\tJSONO
bject returnJson = new JSONObject();\n\t\t\t\tJSONObject userIdentityJson = new JSONObject();\n\t\t\t\tuserI
dentityJson.put(&quot;userName&quot;, userName);\n\t\t\t\tuserIdentityJson.put(&quot;displayName&quot;,
userDisplayName);\n\t\t\t\t\n\t\t\t\treturnJson.put(&quot;status&quot;, &quot;success&quot;);\n\t\t\t\treturnJso
n.put(&quot;userIdentity&quot;, userIdentityJson);\n\t\t\t\treturn returnJson;\n\t\t\t}\t\t\t\n\t\t}\n\t\t\n\t\treturn fail
edResponseJson;\n\t}\n<\/pre>\n<p>The Localstore adapter contains few http API&#8217;s to perform some
basic operations like Add, Update, Create and Delete in client application.<\/p>\n<pre class=\"brush: java; titl
e: ; notranslate\"> @GET\n\t@Path(&quot;\/getAllItems&quot;)\n\tpublic String getAllItems() throws IOExcept
ion{\n\t\tinit();\n\t\tJsonArray jsonArray = new JsonArray();\n\t\tfor(Object key : props.keySet()){\n\t\t\tjsonArra
y.add(parser.parse(props.getProperty((String) key)).getAsJsonObject());\n\t\t}\n\t\treturn jsonArray.toString();
\n\t}\n\n\t@PUT\n\t@Path(&quot;\/addItem&quot;)\n\tpublic void addItem(String itemJson) \n\t\t\tthrows IOEx
ception, URISyntaxException{\n\t\ttry{\n\t\t\tinit();\n\t\t\tint newKey = props.keySet().size()+1;\n\t\t\tprops.put(
String.valueOf(newKey), itemJson);\n\t\t\tURL url = this.getClass().getClassLoader().getResource(&quot;dat
a.properties&quot;); \n\t\t\tFile file = new File(url.toURI().getPath());\n\t\t\tFileOutputStream foStream = new F
ileOutputStream(file);\n\t\t\tprops.store(foStream, &quot;saving new item&quot;);\n\t\t\tfoStream.close();\n\n\t\
t}catch(IOException ioe){\n\t\t\tioe.printStackTrace();\n\t\t}\n\n\t}\n\n\t@POST\n\t@Path(&quot;\/addAllItems&
quot;)\n\tpublic String addAllItems(String itemsJson) \n\t\t\tthrows URISyntaxException, IOException{\n\t\ttry
{\n\t\t\tinit();\n\t\t\tclearAllData();\n\t\t\tJsonArray jsonArr = parser.parse(itemsJson).getAsJsonArray();\n\t\t\tfo
r(int i=0;i&amp;lt;jsonArr.size();i++){\n\t\t\t\tprops.put(String.valueOf(i+1), jsonArr.get(i).toString());\n\t\t\t}\n\t\t\
tURL url = this.getClass().getClassLoader().getResource(&amp;quot;data.properties&amp;quot;); \n\t\t\tFile f
ile = new File(url.toURI().getPath());\n\t\t\tFileOutputStream foStream = new FileOutputStream(file);\n\t\t\tpro
ps.store(foStream, &amp;quot;saving new item&amp;quot;);\n\t\t\tfoStream.close();\n\t\t\treturn &amp;quot;tr
ue&amp;quot;;\n\t\t}catch(IOException ioe){\n\t\t\tioe.printStackTrace();\n\t\t}\n\t\treturn &quot;false&quot;;\n\t
}\n\n\t@DELETE\n\t@Path(&quot;\/clearAll&quot;)\n\tpublic String clearAllData() \n\t\t\tthrows MissingConfig
urationOptionException, URISyntaxException, IOException{\n\t\t\tinit();\n\t\t\tprops.clear();\n\t\t\tSystem.out.p
rintln(&quot;Size : &quot;+props.size());\n\t\t\tURL url = this.getClass().getClassLoader().getResource(&quot;
data.properties&quot;); \n\t\t\tFile file = new File(url.toURI().getPath());\n\t\t\tFileOutputStream foStream = ne
w FileOutputStream(file);\n\t\t\tprops.store(foStream, &quot;clearing all data&quot;);\n\t\t\tfoStream.close();\n
\t\t\treturn &quot;cleared&quot;;\n\t}\n<\/pre>\n<\/li>\n<li> Add TAI Extension in the following path of server d
irectory server\/usr\/extensions<br \/>\nTAI Extension Link : Download the extension.zip from <a href=\"https:
\/\/hub.jazz.net\/project\/chethan\/parkstore-bluemix-server\/overview\" target=\"_blank\">here<\/a>\n<\/li>\n<l
i> Add TAI Security constraint in web.xml file for both the projects.\n<pre class=\"brush: xml; title: ; notranslat
e\">&lt;security-constraint&gt;\n   \t&lt;web-resource-collection&gt;\n   \t   &lt;web-resource-name&gt;Local
storeApplication&lt;\/web-resource-name&gt;\n   \t   &lt;url-pattern&gt;\/apps\/*&lt;\/url-pattern&gt;\n   \t&lt;\/

web-resource-collection&gt;\n	\t&lt;auth-constraint&gt;\n		&lt;role-name&gt;TAIUserRole&lt;\/role-na
me&gt;\n	\t&lt;\/auth-constraint&gt;\n&lt;\/security-constraint&gt;\n&lt;security-role id=&quot;SecurityRole_
TAIUserRole&quot; &gt;\n	&lt;role-name&gt;TAIUserRole&lt;\/role-name&gt;\n&lt;\/security-role&gt;<\/pre
>\n<\/li>\n<li> Add OAuthTai feature in server.xml\n<pre class=\"brush: plain; title: ; notranslate\">&lt;feature
&gt;usr:OAuthTai-1.0&lt;\/feature&gt;<\/pre>\n<\/li>\n<li> Protect the Url&#8217;s using TAI by adding follow
ing code in server.xml\n<pre class=\"brush: xml; title: ; notranslate\"> &lt;usr_OAuthTAI id=&quot;myOAuthT
AI&quot; realmName=&quot;imfRealm&quot;&gt;\n\t\t&lt;securityConstraint httpMethods=&quot;GET, POST
&quot; securedURLs=&quot;\/LocalstoreAdapter\/*&quot;\/&gt;\n\t\t&lt;securityConstraint httpMethods=&quot
;GET, POST&quot; securedURLs=&quot;\/custom-oauth-java\/*&quot;\/&gt;\n\t&lt;\/usr_OAuthTAI&gt; \n\n
&lt;webApplication id=&quot;custom-oauth-java&quot; location=&quot;custom-oauth-java.war&quot; name=&
quot;custom-oauth-java&quot;&gt;\n	&lt;application-bnd&gt;\n\t\t&lt;security-role name=&quot;TAIUserRol
e&quot;&gt;\n\t\t\t&lt;special-subject type=&quot;ALL_AUTHENTICATED_USERS&quot;\/&gt;\n\t\t&lt;\/securi
ty-role&gt;\n\t&lt;\/application-bnd&gt; \n\t&lt;\/webApplication&gt;	\n\t	&lt;webApplication id=&quot;Localsto
reAdapter&quot; location=&quot;LocalstoreAdapter.war&quot; name=&quot;LocalstoreAdapter&quot;&gt;\n
&lt;application-bnd&gt;\n\t\t&lt;security-role name=&quot;TAIUserRole&quot;&gt;\n\t\t\t&lt;special-subject typ
e=&quot;ALL_AUTHENTICATED_USERS&quot;\/&gt;\n\t\t&lt;\/security-role&gt;\n\t&lt;\/application-bnd&gt; \
n\t&lt;\/webApplication&gt;<\/pre>\n<\/li>\n<li> Specify the IMF Auth Url inside Server.env file in liberty.\n<pre
class=\"brush: xml; title: ; notranslate\">imfServiceUrl=https:\/\/imf-authserver.ng.bluemix.net\/imf-authserver
<\/pre>\n<\/li>\n<li> Create a server package which contains above two applications using following comma
nd.\n<pre class=\"brush: plain; title: ; notranslate\">.\/server package ${server_name} --include=usr<\/pre>\n
<\/li>\n<li> Push the newly created server package to bluemix using following command.\n<pre class=\"brus
h: plain; title: ; notranslate\">cf push ${app_name} -p ${path_to_server_package_zip}<\/pre>\n<\/li>\n<\/ul>\n
<h3>Advance Mobile Access service<\/h3>\n<ul>\n<li> Bind the pushed application to Advance Mobile Acce
ss Service.\n<p><a href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/20
15\/07\/Screen-Shot-2015-07-17-at-3.28.04-pm.png\"><img src=\"https:\/\/developer.ibm.com\/mobilefirstplat
form\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-17-at-3.28.04-pm-1024x346.png\" alt=\
"Advance Mobile Access\" width=\"980\" height=\"331\" class=\"alignnone size-large wp-image-14882\" \/><\/
a>\n<\/li>\n<li> Register your client application in AMA dashboard. For more info refer documentation : <a hr
ef=\"https:\/\/www.ng.bluemix.net\/docs\/services\/mobileaccess\/index.html\" target=\"_blank\">click here<\/a
>\n<p><a href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/S
creen-Shot-2015-07-17-at-3.42.32-pm.png\"><img src=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-
content\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-17-at-3.42.32-pm.png\" alt=\"AMA Client Registr
ation\" width=\"935\" height=\"452\" class=\"alignnone size-full wp-image-14883\" \/><\/a>\n<\/li>\n<li> AMA p
rovides Facebook, Google, or a custom identity provider to authenticate access to protected resources. Add
Custom identity provider feature as it can be migrated to MFPF and specify the corresponding jax-rs custom
authentication application url and realm name.<br \/>\n<a href=\"https:\/\/developer.ibm.com\/mobilefirstplatf
orm\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-17-at-4.03.21-pm.png\"><img src=\"http
s:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-
17-at-4.03.21-pm.png\" alt=\"Custom Auth AMA\" width=\"955\" height=\"375\" class=\"alignnone size-full
wp-image-14890\" \/><\/a>\n<\/li>\n<li> Add the following code inside didFinishLaunchingWithOptions functio
n in AppDelegate of client application which will register the realm and initialize connection with Bluemix App
lication.\n<pre class=\"brush: plain; title: ; notranslate\"> IMFClient.sharedInstance().registerAuthenticationD
elegate(customAuthDelegate, forRealm: &quot;customAuthRealm_3&quot;)\nIMFClient.sharedInstance().init
ializeWithBackendRoute(&quot;https:\/\/parkstore.mybluemix.net&quot;, backendGUID: &quot;5e3ad88d-
dd48-469d-b46f-2c4ad66b5345&quot;)<\/pre>\n<\/li>\n<li> The following is the sample code to invoke the R
est url&#8217;s in client application.\n<pre class=\"brush: plain; title: ; notranslate\">var request: IMFResour
ceRequest = IMFResourceRequest(path: &quot;https:\/\/parkstore.mybluemix.net\/LocalstoreAdapter\/apps\/
5e3ad88d-dd48-469d-b46f-2c4ad66b5345\/localstore\/getAllItems&quot;, method: &quot;GET&quot;)\n
request.sendWithCompletionHandler { (wlResponse:IMFResponse!, err:NSError!) -&gt; Void in<\/pre>\n<\/li>
\n<\/ul>\n<h3>Push Service for iOS 8<\/h3>\n<ul>\n<li> Bind the application with Push Service for iOS 8<br
\/>\n<a href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Scre
en-Shot-2015-07-17-at-4.07.01-pm.png\"><img src=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-co
ntent\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-17-at-4.07.01-pm-1024x367.png\" alt=\"Push AMA\
" width=\"980\" height=\"351\" class=\"alignnone size-large wp-image-14891\" \/><\/a>\n<\/li>\n<li> Configure
Apple Push Notification service (APNs) which requires Apple Developer Account and Generate pl2 certificat
es. Documentation link : <a href=\"https:\/\/www.ng.bluemix.net\/docs\/services\/mobilepush\/index.html#certi
ficates\" target=\"_blank\">click here<\/a>\n<\/li>\n<li> Upload the generated pl2 certificate in Push service d

ashboard\n<p><a href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/201
5\/07\/Screen-Shot-2015-07-12-at-6.47.14-pm.png\"><img src=\"https:\/\/developer.ibm.com\/mobilefirstplatf
orm\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-12-at-6.47.14-pm-1024x377.png\" alt=\"
Push Service\" width=\"980\" height=\"361\" class=\"alignnone size-large wp-image-14816\" \/><\/a>\n<\/li>\n
<li>Add the following code inside didFinishLaunchingWithOptions function in AppDelegate of client applicatio
n which will register notifications in client app.\n<pre class=\"brush: plain; title: ; notranslate\"> let notification
Types: UIUserNotificationType = UIUserNotificationType.Badge | UIUserNotificationType.Alert | UIUserNotifi
cationType.Sound\n   let notificationSettings: UIUserNotificationSettings = UIUserNotificationSettings(for
Types: notificationTypes, categories: nil)\n   \n   application.registerUserNotificationSettings(notificatio
nSettings)\n   application.registerForRemoteNotifications()<\/pre>\n<\/li>\n<li>Add the following code insi
de didRegisterForRemoteNotificationsWithDeviceToken function in AppDelegate of client application which
will register pushclient and subscribe to tag in client app.\n<pre class=\"brush: plain; title: ; notranslate\">IMF
PushClient.sharedInstance().registerDeviceToken(deviceToken, completionHandler: { (response, error) -&gt;
Void in\n   if error != nil {\n   println(&quot;Error during device registration \\(error.description)
&quot;)\n   }\n   else {\n   println(&quot;Response during device registration json: \\(res
ponse.responseJson.description)&quot;)\n   var tags = [&quot;parkstore&quot;]\n   IMFPus
hClient.sharedInstance().subscribeToTags(tags, completionHandler: { (response:IMFResponse!, err:NSError
!) -&gt; Void in\n   if err != nil {\n   println(&quot;There was an error while subscribi
ng to tag&quot;)\n   }else{\n   println(&quot;Successfully subscribe to tag parkstore
&quot;)\n   }\n   })\n   }<\/pre>\n<\/li>\n<li>Add the following function inside Appde
legate which triggers when push notification arrived in client app.\n<pre class=\"brush: plain; title: ; notranslat
e\">func application(application: UIApplication, didReceiveRemoteNotification userInfo: [NSObject : AnyObje
ct]) {\n   println(&quot;Got remote Notification. Data : \\(userInfo.description)&quot;)\n   let info = userI
nfo as NSDictionary\n   let data = info.objectForKey(&quot;aps&quot;)?.objectForKey(&quot;alert&quot;)
as! NSDictionary\n   let userData = data.objectForKey(&quot;body&quot;) as! String\n   let alertView =
UIAlertView(title: &quot;WishList!&quot;, message: &quot;\\(userData)&quot;, delegate: nil, cancelButtonTitle
: &quot;OK&quot;)\n   alertView.show()\n   }\n}<\/pre>\n<\/li>\n<\/ul>\n<h2 id=\"migrateblu\">Existing Bl
uemix Client Application<\/h2>\n<p>Add the following Code snippets to the existing Bluemix Client Applicati
on and name the application with same name which you have registered in Advance Mobile Access Dashbo
ard.<\/p>\n<ul>\n<li> Add the following code inside didFinishLaunchingWithOptions function in AppDelegate
of client application which will register the realm and initialize connection with Bluemix Application.\n<pre cla
ss=\"brush: plain; title: ; notranslate\"> IMFClient.sharedInstance().registerAuthenticationDelegate(customAu
thDelegate, forRealm: &quot;customAuthRealm_3&quot;)\nIMFClient.sharedInstance().initializeWithBackend
Route(&quot;https:\/\/parkstore.mybluemix.net&quot;, backendGUID: &quot;5e3ad88d-dd48-469d-b46f-2c4a
d66b5345&quot;)<\/pre>\n<\/li>\n<li> The following is the sample code to invoke the Rest url&#8217;s in clie
nt application.\n<pre class=\"brush: plain; title: ; notranslate\">var request: IMFResourceRequest = IMFReso
urceRequest(path: &quot;https:\/\/parkstore.mybluemix.net\/LocalstoreAdapter\/apps\/5e3ad88d-dd48-469d-
b46f-2c4ad66b5345\/localstore\/getAllItems&quot;, method: &quot;GET&quot;)\n   request.sendWithCom
pletionHandler { (wlResponse:IMFResponse!, err:NSError!) -&gt; Void in<\/pre>\n<\/li>\n<li>Add the followin
g code inside didFinishLaunchingWithOptions function in AppDelegate of client application which will registe
r notifications in client app.\n<pre class=\"brush: plain; title: ; notranslate\"> let notificationTypes: UIUserNoti
ficationType = UIUserNotificationType.Badge | UIUserNotificationType.Alert | UIUserNotificationType.Sound\
n   let notificationSettings: UIUserNotificationSettings = UIUserNotificationSettings(forTypes: notificationT
ypes, categories: nil)\n   \n   application.registerUserNotificationSettings(notificationSettings)\n   a
pplication.registerForRemoteNotifications()<\/pre>\n<\/li>\n<li>Add the following code inside
didRegisterForRemoteNotificationsWithDeviceToken function in AppDelegate of client application which will
register pushclient and subscribe to tag in client app.\n<pre class=\"brush: plain; title: ; notranslate\">IMFPus
hClient.sharedInstance().registerDeviceToken(deviceToken, completionHandler: { (response, error) -&gt; Voi
d in\n   if error != nil {\n   println(&quot;Error during device registration \\(error.description)&qu
ot;)\n   }\n   else {\n   println(&quot;Response during device registration json: \\(respons
e.responseJson.description)&quot;)\n   var tags = [&quot;parkstore&quot;]\n   IMFPushCli
ent.sharedInstance().subscribeToTags(tags, completionHandler: { (response:IMFResponse!, err:NSError!) -
&gt; Void in\n   if err != nil {\n   println(&quot;There was an error while subscribing
to tag&quot;)\n   }else{\n   println(&quot;Successfully subscribe to tag parkstore&q
uot;)\n   }\n   })\n   }<\/pre>\n<\/li>\n<li>Add the following function inside Appdeleg
ate which triggers when push notification arrived in client app.\n<pre class=\"brush: plain; title: ; notranslate\"
>func application(application: UIApplication, didReceiveRemoteNotification userInfo: [NSObject : AnyObject]

>func application(application: UIApplication, didReceiveRemoteNotification userInfo: [NSObject : AnyObject]
) {\n   println(&quot;Got remote Notification. Data : \\(userInfo.description)&quot;)\n   let info = userInfo
as NSDictionary\n   let data = info.objectForKey(&quot;aps&quot;)?.objectForKey(&quot;alert&quot;) as!
NSDictionary\n   let userData = data.objectForKey(&quot;body&quot;) as! String\n   let alertView = UI
AlertView(title: &quot;WishList!&quot;, message: &quot;\\(userData)&quot;, delegate: nil, cancelButtonTitle: &
quot;OK&quot;)\n   alertView.show()\n   }\n}<\/pre>\n<\/li>\n<li>The following are the screenshots of cli
ent application.<br \/>\n<a href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\
/32\/2015\/07\/IMG_0020.jpg\"><img src=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uplo
ads\/sites\/32\/2015\/07\/IMG_0020-169x300.jpg\" alt=\"IMG_0020\" width=\"169\" height=\"300\" class=\"alig
nnone size-medium wp-image-14917\" \/><\/a><a href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-
content\/uploads\/sites\/32\/2015\/07\/IMG_00211.jpg\"><img src=\"https:\/\/developer.ibm.com\/mobilefirstpl
atform\/wp-content\/uploads\/sites\/32\/2015\/07\/IMG_00211-169x300.jpg\" alt=\"IMG_0021\" width=\"169\"
height=\"300\" class=\"alignnone size-medium wp-image-14918\" \/><\/a><a href=\"https:\/\/developer.ibm.co
m\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/IMG_0025.jpg\"><img src=\"https:\/\/develop
er.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/IMG_0025-169x300.jpg\" alt=\"IMG
_0025\" width=\"169\" height=\"300\" class=\"alignnone size-medium wp-image-14920\" \/><\/a><a href=\"htt
ps:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/IMG_0024.jpg\"><im
g src=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/IMG_0024-
169x300.jpg\" alt=\"IMG_0024\" width=\"169\" height=\"300\" class=\"alignnone size-medium wp-image-1491
9\" \/><\/a><a href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/0
7\/IMG_0026.jpg\"><img src=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/3
2\/2015\/07\/IMG_0026-169x300.jpg\" alt=\"IMG_0026\" width=\"169\" height=\"300\" class=\"alignnone size-
medium wp-image-14921\" \/><\/a>\n<\/li>\n<\/ul>\n<h2>Migration to On-Prem<\/h2>\n<h3 id=\"configureclie
nt\">Migration of Client Application<\/h3>\n<p>Migration of Client Application includes following two steps<\/
p>\n<li>Configuring Cocoapods<\/li>\n<li>Client App Migration<\/li>\n<h3 id=\"cocoapods\">Configuring Coc
oapods<\/h3>\n<p>If CocoaPods has not been installed on a specific computer:<\/p>\n<ul>\n<li>Follow the
&#8220;Getting Started&#8221; guide for CocoaPods installation: http:\/\/guides.cocoapods.org\/using\/getti
ng-started.html<\/li>\n<li>Open &#8220;Terminal&#8221; at the installation location and run the &#8220;pod
init&#8221; command<\/li>\n<\/ul>\n<p>The following steps assume that the client application is working wit
h CocoPods. If not, follow this &#8220;Using CocoaPods&#8221; documentation : <a href=\"http:\/\/guides.c
ocoapods.org\/using\/using-cocoapods.html\" target=\"_blank\">click here<\/a><\/p>\n<p>In both cases, the i
nstructions below explain how to edit the &#8220;Podfile&#8221; file.<\/p>\n<ol>\n<li>Open the &#8220;Pod
file&#8221; file located in the root of your XCode project in a favourite text editor.<\/li>\n<li>Comment out or
remove the existing content.<\/li>\n<li>Add the following lines:\n<pre class=\"brush: plain; title: ; notranslate\"
>source 'https:\/\/github.rtp.raleigh.ibm.com\/imflocalsdks\/imf-client-sdk-specs.git'\npod 'IMFCompatibility'<\/
pre>\n<\/li>\n<li>Open &#8220;Terminal&#8221; at the location of &#8220;Podfile&#8221;.<\/li>\n<li>Verify t
hat the XCode project is closed.<\/li>\n<li>Run the &#8220;pod install&#8221; command.<\/li>\n<\/ol>\n<p>
Open the [MyProject].xcworkspace file in XCode. This file is located side by side with [MyProject].xcodeproj.
<br \/>\nAn usual CocoaPods-based project is managed as a workspace containing the application (the exe
cutable) and the library (all project dependencies brought by the CocoaPods manager).<\/p>\n<p>In Xcode&
#8217;s Build Settings, search for &#8220;Other Linker Flags&#8221; and insert ${inherited} (if -ObjC is defi
ned in this field, you can just delete it, since it is configured in the CocoaPod project).<\/p>\n<h3>Client App
Migration<\/h3>\n<ol>\n<li>Search for bluemix dependency imports like\n<pre class=\"brush: plain; title: ; not
ranslate\">#import &lt;IMFCore\/IMFCore.h&gt;\n#import &lt;IMFPush\/IMFPush.h&gt;<\/pre>\n<p>Replace t
he above imports with <\/p>\n<pre class=\"brush: plain; title: ; notranslate\">#import &lt;IMFCompatibility\/IM
FCompatibility.h&gt;<\/pre>\n<\/li>\n<li>Look for a call to the &#8220;initializeWithBackendRoute&#8221; m
ethod and replace the route URL with your on-premise server URL. For example:\n<pre class=\"brush: plain;
title: ; notranslate\">IMFClient.sharedInstance().initializeWithBackendRoute(&quot;https:\/\/parkstore.myblue
mix.net&quot;, backendGUID: &quot;5e3ad88d-dd48-469d-b46f-2c4ad66b5345&quot;<\/pre>\n<p>should b
e replaced with your on-premise MFP server URL<\/p>\n<pre class=\"brush: plain; title: ; notranslate\">IMFCl
ient.sharedInstance().initializeWithBackendRoute(&quot;http:\/\/localhost:9080\/ParkStoreMFP&quot;, backe
ndGUID: &quot;5e3ad88d-dd48-469d-b46f-2c4ad66b5345&quot;<\/pre>\n<p>Note, that backendGUID para
meter is ignored and can be empty. Look for all instantiations of IMFResourceRequest class and update it<\/l
i>\n<li>Look for all instantiations of IMFResourceRequest class and update the request URL with absolute or
relative path to the resource. For example:\n<pre class=\"brush: plain; title: ; notranslate\">var request: IMFR
esourceRequest = IMFResourceRequest(path: &quot;https:\/\/parkstore.mybluemix.net\/LocalstoreAdapter\/
apps\/5e3ad88d-dd48-469d-b46f-2c4ad66b5345\/localstore\/getAllItems&quot;, method: &quot;GET&quot;)<
\/pre>\n<p>should be replaced with <\/p>\n<pre class=\"brush: plain; title: ; notranslate\">var request: IMFRe

\/pre>\n<p>should be replaced with<\/p>\n<pre class=\"brush: plain; title: ; notranslate\">var request: IMFRe
sourceRequest = IMFResourceRequest(path: &quot;http:\/\/localhost:9080\/ParkStoreMFP\/adapters\/Localst
oreAdapter\/localstore\/getAllItems&quot;, method: &quot;GET&quot;)<\/pre>\n<\/li>\n<li>Add the following c
ode inside didRegisterForRemoteNotificationsWithDeviceToken function in Appdelegate of Client application
.\n<pre class=\"brush: plain; title: ; notranslate\"> WLPush.sharedInstance().tokenFromClient = deviceToken.
description<\/pre>\n<\/li>\n<li>All on-premise applications require the &#8220;worklight.plist&#8221; file to b
e present in the application resources. In the <code>IBMMobileFirstPlatformFoundationNativeSDK<\/code>
pod we supply a file named <strong>sample.worklight.plist<\/strong>.\n<ul>\n<li>Locate the &#8220;sample.
worklight.plist&#8221; file in the â€˜IBMMobileFirstPlatformFoundationNativeSDKâ€™ pod.<\/li>\n<li>Copy t
his file to the parent (application) project and rename it to &#8220;worklight.plist&#8221;.<\/li>\n<li>Edit the
&#8220;worklight.plist&#8221; file by setting the &#8220;application id&#8221; key to the name of your appli
cation deployed to the on-premise MFPF server<\/li>\n<\/ul>\n<\/li>\n<\/ol>\n<h3 id=\"migratemfp\">Migration
of JAX-RS Application to JAVA Adapter<\/h3>\n<ol>\n<li>To migrate JAX-RS application to on-prem (Mobile
First Foundation) server we need to do the following steps for server:\n<p>   Create MobileFirst Project &#8
211;&gt; Create native API app for iOS<br \/>\n   â€‹â€‹<br \/>\n<a href=\"https:\/\/developer.ibm.com\/m
obilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-12-at-6.50.04-pm.png\"><i
mg src=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-S
hot-2015-07-12-at-6.50.04-pm.png\" alt=\"Screen Shot 2015-07-12 at 6.50.04 pm\" width=\"595\" height=\"59
6\" class=\"alignnone size-full wp-image-14817\" \/><\/a><\/p>\n<p><a href=\"https:\/\/developer.ibm.com\/m
obilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-12-at-6.51.13-pm.png\"><i
mg src=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-S
hot-2015-07-12-at-6.51.13-pm.png\" alt=\"Screen Shot 2015-07-12 at 6.51.13 pm\" width=\"598\" height=\"59
0\" class=\"alignnone size-full wp-image-14818\" \/><\/a><\/p>\n<p><a href=\"https:\/\/developer.ibm.com\/m
obilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-12-at-6.52.28-pm.png\"><i
mg src=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-S
hot-2015-07-12-at-6.52.28-pm.png\" alt=\"Screen Shot 2015-07-12 at 6.52.28 pm\" width=\"717\" height=\"42
4\" class=\"alignnone size-full wp-image-14819\" \/><\/a><\/li>\n<li>Add two adapters for Custom Authenticati
on and Localstore and migrate the JAX-RS code as shown in the following example.<\/li>\n<\/ol>\n<p>Copy
the JAX-RS BlueMix code and paste it in the newly created Localstore Java adapter JAX-RS file.<\/p>\n<p>
Add and remove the following changes in your adapter code.<\/p>\n<ul>\n<li> remove <code>\/{tenantId}\/<\
/code><\/li>\n<li> remove the <code>@PathParam -&gt; PathParam(\"tenantId\") String deviceId<\/code> an
d <code>@PathParam(\"realmName\") String realmName<\/code><\/li>\n<li> Add scope to the all http api re
source <code>@OAuthSecurity (scope=\"customAuthRealm_3\")<\/code><\/li>\n<\/ul>\n<p>The code looks l
ike the following<\/p>\n<pre class=\"brush: plain; title: ; notranslate\">\n\t@GET\n\t@OAuthSecurity (scope=
&quot;customAuthRealm_3&quot;)\n\t@Path(&quot;\/getAllItems&quot;)\n\tpublic String getAllItems() throws
MissingConfigurationOptionException{\n\t\tinit();\n\t\tJsonArray jsonArray = new JsonArray();\n\t\tfor(Object k
ey : props.keySet()){\n\t\t\tjsonArray.add(parser.parse(props.getProperty((String) key)).getAsJsonObject());\n
\t\t}\n\t\treturn jsonArray.toString();\n\t}\n\n\t@PUT\n\t@OAuthSecurity (scope=&quot;customAuthRealm_3&
quot;)\n\t@Path(&quot;\/addItem&quot;)\n\tpublic void addItem(String itemJson) \n\t\t\tthrows MissingConfigu
rationOptionException, URISyntaxException, IOException{\n\t\ttry{\n\t\t\tinit();\n\t\t\tint newKey = props.keySe
t().size()+1;\n\t\t\tprops.put(String.valueOf(newKey), itemJson);\n\t\t\tURL url = this.getClass().getClassLoad
er().getResource(&quot;data.properties&quot;); \n\t\t\tFile file = new File(url.toURI().getPath());\n\t\t\tFileOutp
utStream foStream = new FileOutputStream(file);\n\t\t\tprops.store(foStream, &quot;saving new item&quot;);\
n\t\t\tfoStream.close();\n\n\t\t}catch(IOException ioe){\n\t\t\tioe.printStackTrace();\n\t\t}\n\n\t}\n\n\t@POST\n\t
@OAuthSecurity (scope=&quot;customAuthRealm_3&quot;)\n\t@Path(&quot;\/addAllItems&quot;)\n\tpublic
String addAllItems(String itemsJson) \n\t\t\tthrows MissingConfigurationOptionException, URISyntaxExceptio
n, IOException{\n\t\ttry{\n\t\t\tinit();\n\t\t\tclearAllData();\n\t\t\tJsonArray jsonArr = parser.parse(itemsJson).get
AsJsonArray();\n\t\t\tfor(int i=0;i&amp;lt;jsonArr.size();i++){\n\t\t\t\tprops.put(String.valueOf(i+1), jsonArr.get(i)
.toString());\n\t\t\t}\n\t\t\tURL url = this.getClass().getClassLoader().getResource(&amp;quot;data.properties&
amp;quot;); \n\t\t\tFile file = new File(url.toURI().getPath());\n\t\t\tFileOutputStream foStream = new FileOutpu
tStream(file);\n\t\t\tprops.store(foStream, &amp;quot;saving new item&amp;quot;);\n\t\t\tfoStream.close();\n\t\
t\treturn &amp;quot;true&amp;quot;;\n\t\t}catch(IOException ioe){\n\t\t\tioe.printStackTrace();\n\t\t}\n\t\treturn
&amp;quot;false&amp;quot;;\n\t}\n\n\t@DELETE\n\t@OAuthSecurity(enabled=false)\n\t@Path(&quot;\/clear
All&quot;)\n\tpublic String clearAllData() \n\t\t\tthrows MissingConfigurationOptionException, URISyntaxExce
ption, IOException{\n\t\tinit();\n\t\tprops.clear();\n\t\tSystem.out.println(&quot;Size : &quot;+props.size());\
n\t\tURL url = this.getClass().getClassLoader().getResource(&quot;data.properties&quot;); \n\t\tFile file = n
ew File(url.toURI().getPath());\n\t\tFileOutputStream foStream = new FileOutputStream(file);\n\t\tprops.stor

e(foStream, &quot;clearing all data&quot;);\n\t\t\tfoStream.close();\n\t\t\treturn &quot;cleared&quot;;\n\t}\n<\/p
re>\n<h3 id=\"configoauth\">Configuring Custom-OAuth<\/h3>\n<ul>\n<li>Add realm with same name you h
ad on BlueMix and login module to the authenticationConfig.xml.\n<pre class=\"brush: xml; title: ; notranslate\
">&lt;realm name=&quot;customAuthRealm_3&quot; loginModule=&quot;customAuthLoginModule_3&quot;&
gt;\n&lt;className&gt;com.worklight.core.auth.ext.CustomIdentityAuthenticator&lt;\/className&gt;\t\n&lt;pa
rameter name=&quot;providerUrl&quot; value=&quot;http:\/\/localhost:9080\/ParkStoreMFP\/adapters\/Custo
mauth&quot;\/&gt;\n&lt;\/realm&gt;\n\n&lt;loginModule name=&quot;customAuthLoginModule_3&quot; expira
tionInSeconds=&quot;3600&quot;&gt;\n&lt;className&gt;com.worklight.core.auth.ext.CustomIdentityLogin
Module&lt;\/className&gt;\n&lt;\/loginModule&gt;<\/pre>\n<\/li>\n<li>Add Custom-oauth Realm in userIdenti
tyRealms in Application Descriptor file of iOS Native API\n<pre class=\"brush: xml; title: ; notranslate\">&lt;us
erIdentityRealms&gt;customAuthRealm_3&lt;\/userIdentityRealms&gt;<\/pre>\n<\/li>\n<\/ul>\n<h3 id=\"config
urepush\">Configuring Push Capability<\/h3>\n<ul>\n<li> Add apns p12 certificate which is generated from A
pple Developer Account under iOS Native API Folder\n<p><a href=\"https:\/\/developer.ibm.com\/mobilefirst
platform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-12-at-6.58.03-pm.png\"><img src=\
"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-Shot-2015
-07-12-at-6.58.03-pm.png\" alt=\"Screen Shot 2015-07-12 at 6.58.03 pm\" width=\"286\" height=\"171\" class
=\"alignnone size-full wp-image-14820\" \/><\/a>\n<\/li>\n<li> Add Push configuration in Application Descripto
r file of iOS Native API and include the password of added apns certificate.\n<pre class=\"brush: xml; title: ; n
otranslate\">&lt;pushSender password=&quot;password&quot;\/&gt;\n&lt;tags&gt;\n &lt;tag&gt;\n   &lt;name
&gt;parkstore&lt;\/name&gt;\n &lt;\/tag&gt;\n&lt;\/tags&gt;<\/pre>\n<\/li>\n<li> Create HTTP Push Adapter wit
h following function code which will send the user push notification to the devices which is subscribed to tag
&#8220;parkstore&#8221;.\n<pre class=\"brush: xml; title: ; notranslate\">function sendTagNotification(notific
ationText) {\n   var notificationOptions = {};\n   notificationOptions.message = {};\n   notificationOptions.targ
et = {};\n\n   notificationOptions.message.alert = notificationText;\n   notificationOptions.target.tagNames = [
&quot;parkstore&quot;];\n\n   WL.Server.sendMessage(&quot;ParkStoreMFP&quot;, notificationOptions);\n\
n   return {\n   result : &quot;Notification sent to users subscribed to the tag parkstore.&quot;\n   };\n}<\/p
re>\n<\/li>\n<\/ul>\n<p>By performing above steps one can easily run iOS app built for Bluemix on MobileFir
st Platform and following are the links to samples.<\/p>\n<h3 id=\"sample\">Sample and Source Code<\/h3>\
n<p>Bluemix Server : <a href=\"https:\/\/hub.jazz.net\/git\/chethan\/parkstore-bluemix-server\">Parkstore blu
emix server<\/a><br \/>\nBluemix Client : <a href=\"https:\/\/hub.jazz.net\/git\/chethan\/parkstore-bluemix\">P
arkstore bluemix<\/a><br \/>\nMFP Server   : <a href=\"https:\/\/hub.jazz.net\/git\/chethan\/parkstore-mfp-se
rver\">Parkstore mfp server<\/a><br \/>\nMFP Client   : <a href=\"https:\/\/hub.jazz.net\/git\/chethan\/parksto
re-mfp\">Parkstore mfp<\/a><\/p>\n<p>The post <a rel=\"nofollow\" href=\"https:\/\/developer.ibm.com\/mobil
efirstplatform\/2015\/08\/19\/try-on-bluemix-and-buy-mfp\/\">Try on Bluemix and migrate to on-prem MobileFi
rst Platform<\/a> appeared first on <a rel=\"nofollow\" href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\"
>IBM MobileFirst Platform<\/a>.<\/p>",
            "guid": {
              "content": "https:\/\/developer.ibm.com\/mobilefirstplatform\/?p=14769",
              "isPermaLink": "false"
            },
            "link": "https:\/\/developer.ibm.com\/mobilefirstplatform\/2015\/08\/19\/try-on-bluemix-and-buy-mfp\/
",
            "pubDate": "Wed, 19 Aug 2015 10:36:51 +0000",
            "title": "Try on Bluemix and migrate to on-prem MobileFirst Platform"
          }
        ],
        "language": "en-US",
        "lastBuildDate": "Tue, 08 Sep 2015 09:22:53 +0000",
        "link": [
          {
            "href": "https:\/\/developer.ibm.com\/mobilefirstplatform\/feed\/",
            "rel": "self",
            "type": "application\/rss+xml"
          },
          "https:\/\/developer.ibm.com\/mobilefirstplatform"
        ],
        "title": "IBM MobileFirst Platform",

```
        "updateFrequency": "1",
        "updatePeriod": "hourly"
      },
      "version": "2.0"
   }
}
```

Sample application

Click to download (https://github.com/MobileFirst-Platform-Developer-Center/JavaAdapters) the MobileFirst project.