Java HTTP Adapter

Overview

This tutorial is a continuation of Java Adapter (../../../server-side-development/java-adapter/) and assumes previous knowledge of the concepts described there.

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 shows an example of a Java adapter that connects to an RSS feed by using a Java HttpClient.

Topics:

- RSSAdapterApplication
- RSSAdapterResource
- Results

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

```
@Path("/")

public class RSSAdapterResource {
}
```

RSSAdapterResource is where we handle the requests to your adapter.

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

HTTP Client

```
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

```
@GET
@Produces("application/json")
public void get(@Context HttpServletResponse response, @QueryParam("tag") String tag) throws Cl
ientProtocolException, 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()

```
public void execute(HttpUriRequest req, HttpServletResponse resultResponse) throws ClientProtocol
Exception, 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 flushed and closed.

If RSSResponse is not 200 0K, we write the status code and reason in the response instead.

Results

Use the testing techniques described in Java Adapter (../#testing) to test your work.

The adapter should return the RSS feed converted to JSON.

```
"mobile_development",
    "mobilefirst",
    "xamarin"
],
    "commentRss": "https:\/\developer.ibm.com\/mobilefirstplatform\/2015\/09\/01\/integrating-mq
a-into-xamarin-android-app\/feed\/",
    "comments": [
        "https:\/\developer.ibm.com\/mobilefirstplatform\/2015\/09\/01\/integrating-mqa-into-xamari
n-android-app\/#comments",
        "0"
],
    "creator": "Vidyasagar MSC",
    "description": "The post <a rel=\"nofollow\" href=\"https:\/\developer.ibm.com\/mobilefirst
```

"description": "The post Integrating MQA into Xamarin.And roid app<\/a> appeared first on IBM MobileFirst Platform<\/a>.<\ $\$ p>",

"encoded": "It all started 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 pro blem, let's define MQA.\n<h4>What is MQA?</h4>\nMQA stands for & amp;#822 0;Mobile Quality Assurance" and is part of the IBM MobileFirst Platform. >IBM MQA provides line of business professionals and develo pment teams with insightful and streamlined quality feedback and metrics from both pre-production and production, enabling them to prioritize and take action to support a dynamic mobile app strategy. ><\brace >\n\n
div style=\"width: 1058px\" class=\" wp-caption aligncenter\"><\/a><p class=\"wp-caption-text\">Features of Mobile Quality Assurance.<\p><\div>\nNote<\ /strong><\em>: To understand more about MQA, visit IBM Mobile Quality Assurance<//a> So, by now we should be good with the first part of our blog title that is MQA. So, the next question is<V p>\n<h4>What is Xamarin.Android?<\/h4>\nXamarin is a platform to create nativeA iOS, Android, M ac and Windows apps in C#. Xamarin.Android allows us to create native Android applications using th e same UI controls we would in Java, except with the flexibility and elegance of a modern language (C#).\nAs we are good with the definitions, let's address the problem.\n<stro ng>What's the problem in integrating MQA into Xamarin Android app?<//strong>\n<p >At the time of this blog post, the available MQA SDKs are iOS native SDK, Android native SDK and Ja vascript A SDK.\nSo, we have to find a workaround to address this use-case. The initial step is to download the Android MQA SDK and see what's provided. you can download it from here<\/a>. Once successfully downloaded and unzippe d, we should see a jar file namely MQA-Android-library-&lt;version number&gt;. jarA under lib folder.\n<div style=\"width: 634px\" class=\"wpcaption aligncenter\"><\/a><p class=\"wp-caption-text\ ">MQA Android SDK</div>\nAs Xamarin is C# based, What can we do with this jar file? nWe haveÄ Xamarin bindings<\strong> to our rescue, which helps using in consuming .JA Rs from C#.\nNote: Steps to consume MQA Android JAR in a $Xamarin. And roid\ app\ is\ mentioned \hat{A}\ <a\ href=\\"https://developer.xamarin.com/guides/and roid/advanc" and roid advanction of the control of the con$ ed_topics\/java_integration_overview\/binding_a_java_library_(.jar)\/\">here<\/a><\/p>\n<div style=\"widt h: 257px\" class=\"wp-caption aligncenter\">Xamarin binding project with MQA Android .JAR file</div>\nThe files of our interest her e are MQA-Android-library-2.7.4.jar<\/strong> (Version number may vary) and Metada ta xml <\/strong><\/n>\n<\li>MOA-Android-library-2 7 4 iar file will have all the MOA related classe

s and methods required for us to start an Android MQA session.<\/li>\nMetadata.xml- Allows c hanges to be made to the final API, such as changing the namespace of the generated binding. \li>\n<\ul>\nBased on the errors thrown while building the project, Metadata.xml in my case looks li ke this\n<metadata&gt;\n &lt;!--\n T his sample removes the class: android.support.v4.content.AsyncTaskLoader.LoadTask:\n <rem ove-node path=&:quot:\/api\/packaqe[@name='android.support.v4.content']\/class[@name='AsyncT askLoader.LoadTask']" \/&qt:\n \n This sample removes the method: android.support.v 4.content.CursorLoader.loadInBackground:\n <remove-node path=&quot;\/api\/package[@ name='android.support.v4.content']\/class[@name='CursorLoader']\/method[@name='loadInBackgroun d']" \/&qt;\n -->\n\n <remove-node path=&quot;\/api\/package[@na me='ext.com.google.inject.spi']\/class[@name='InjectionPoint.Factory.1']"\/>\n & lt;remove-node path="\/api\/package[@name='ext.com.google.inject.spi']\/class[@name='Inje ctionPoint.Factory.2']"\/>\n <remove-node path=&quot;\/api\/package[@ name='com.applause.android.log']\/interface[@name='LoggerInterface']"\/>\n &l t;remove-node path="\/api\/package[@name='ext.com.google.inject.internal']"\/&a mp;qt;\n <remove-node path=&quot;\/api\/package[@name='ext.com.google.inject.matcher 'l&:guot:\/&:gt:\n &:lt:remove-node path=&:guot:\/api\/package[@name='com.applaus e.android.util']\/class[@name='AbstractRequest']"\/>\n <remove-node path=& amp;quot;\/api\/package[@name='ext.com.google.inject.spi']\/class[@name='Elements.RecordingBinde r']\/method[@name='bind' and count(parameter)=1 and parameter[1][@type='ext.com.google.inject.Kev']]"\/&qt;\n\n<attr path=&quot;\/api\/package[@name='com.applause.android. messages']\class[@name='Message']\field[@name='message']\" name="manage dName">Message1<Vattr&gt;\n&lt;attr path=&quot;VapiVpacka qe[@name='com.applause.android.log']" name="managedName"& :gt;log<\/attr&gt;\n&lt;\/metadata&gt;\n\n\nOnce all the errors are fixed and your binding project builds successfully, add a new Xamarin Android project (if you haven 217;t added yet). Now, add MQA binding project reference in our Xamarin android app. Note: Both your binding project and Xamarin.Android project should be of same <stron g>target framework.A You can verify this by right clicking on your project -&gt; Options -> General.\n<div id=\"attachment_83\" style=\"width: 270px\" class=\"wp-caption aligncente r\"><\a><p class=\ "wp-caption-text\">Xamarin Android project with added reference to MQA<\/p><\/div>\nNow, let&am p;#8217;s start MQA android session in our Count. Android app. Before doing this, we should create a MQA service on IBM Bluemix. You can follow the instructions mentioned at Getting st arted with Mobile Quality Assurance- Bluemix<//a>Â or watch this video.\n<iframe class='youtube-player' type='text\/html' widt h='980' height='582' src='https:\/\/www.youtube.com\/embed\/zHRfGatcKPM?version=3&rel= 1&fs=1&showsearch=0&showinfo=1&iv load policy=1& #038;wmode=transparent' frameborder='0' allowfullscreen='true'><\id>\odors\rangle span><\p>\nStarting a Mobile Quality Assurance<\/span><\/span> session with the Android SDK entails three steps. First, build a configuration to define how Mobile Quality Assurance<\/span><\/sp an>Â works with your app. Second, start the session itself. Third, add tracking to your activities. Open < strong>MainActivity.cs<\strong> file (Android Project) and paste the code provided below<\p>\npre cl ass=\"brush: csharp; title: ; notranslate\">using System;\n\nusing Android.App;\nusing Android.Content;\ nusing Android.Runtime;\nusing Android.Views;\nusing Android.Widget;\nusing Android.OS;\n\v\MQA r eferences\nusing Com.lbm.Mga.Config:\nusing Com.lbm.Mga:\n\n\namespace Count.Android\n{\n\t[A ctivity (Label = "Count.Android", MainLauncher = true, Icon = "@drawa ble\icon")]\n\tpublic class MainActivity : Activity\n\t\tint count = 1;\n\t\t\v\Use your own ge

ta.Am...votrong>-vp>titali>titali>titali>titalion notary E.r.-t.jar nio vini havo an tho triax totatoa otacoco

 $APP_KEY=\& amp; quot; 1g59b7d884f9fdf5426162e5cb1f87a700648bce4fg0g1g379e0d3a\& amp; quot;; \\ \t the thick of the configuration of the configuration in the$

nerated APP KEY\n\t\tconst string

THAPINEY(APP KEY) VVProvides the quality assurance application APP KEY/N/I/I/I.WITHMODE(INIQA. Mode.Qa) \WSelects the quality assurance application mode\n\t\t\t\t.WithReportOnShakeEnabled(true) \ WEnables shake report trigger\n\t\t\t.WithDefaultUser("default_user@email.com") \/\Sets a default user and user selection\n\t\t\t\.Build();\n\n\t\t\\\Starting MQA Android Session\n\t\t\ MQA.StartNewSession (this, configuration);\n\t\t\V\V Set our view from the "main" I ayout resource\n\t\t\SetContentView (Resource.Layout.Main);\n\n\t\t\VV Get our button from the layout resource,\n\t\t\t\v\ and attach an event to it\n\t\t\Button button = FindViewById&It;Button> (Resource.ld.myButton);\n\t\t\n\t\t\button.Click += delegate {\n\t\t\t\button.Text = string.Format (& quot;{0} clicks!", count++);\n\t\t\};\n\t\t\}\n\n\n\n\n\c\vpre>\nNow, MQA is integrated int o Xamarin. Android app and we are good to go.
What we have implemented above is just a dr op in the Ocean of MQA, to know more about MQA and its features & amp;#8211; VisitÄ MQA Knowledge Centre<\/a><\/p>\nHappy Coding !!!\nThe post Integrating MQA into Xamarin.Android app<\/a> ap peared first on IBM MobileFi rst 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-xa
marin-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\
/#comments",
           "0"
         ],
         "creator": "ChethanKumar",
```

"description": "The post Try on Bluemix and migrate to on-prem MobileFi rst Platform<\/a> appeared first on IBM MobileFirst Platform<\/a>.<\/p>",

by Custom Authenticator via AMA security service provided by bluemix.<VIi>Nn<Ii>On the server side, th e App contains a JAX-RS class to store and manipulate the data. It also contains the server side AMA s ecurity implementation.\nOn BlueMix we have application with the following configuration. on:\n\nLiberty Runtime: which used to run JAX-RS application on Bluemix nce Mobile Access service: which gives mobile application security and monitoring functionality li>Push Service for iOS 8 : which provides the capability to use iOS Push features<\/li>\n<\/ul>\n<h3> Li berty Runtime <\/h3>\n\nLiberty contains two projects with JAX-RS service (i.e Custom-oauth-j ava for Custom Authentication and LocalstoreAdapter for storing items). The service include the protect ed resource and the custom identity provider code. The liberty server is configured with TAI.\n<\/li> >Trust Association Interface (TAI) is a service provider API that enables the integration of third-party se curity services with a Liberty profile server. For more info on TAI: click here<\/a>\n<\/li>The custom identity provider authenticat es a user by sending challenges to the client. However, custom identity providers do not communicate directly with clients. They send challenges and receive responses to the challenges by means of the Ad vanced Mobile Access service. When a custom identity provider successfully authenticates the user, it provides the user identity information to Advanced Mobile Access. For more information on custom auth entication refer bluemix documentation : click here<\/a>\n<p >The custom identity provider code is defined by two http API:\nhpre class=\"brush: plain; title: ; no translate\">\startAutorization<\pre>\n and\nclass=\"brush: plain; title:; notranslate\">\handleC hallengeAnswer\nlores \notion class=\"brush: java; title: ; notranslate\"> @POST\n\t@Consumes (& amp; quot;applicationVjson")\n\t@Path("\/{tenantId}\/customAuthRealm 3\/startAuthoriz ation")\n\t@Produces(MediaType.APPLICATION_JSON)\n\tpublic JSONObject startAuthoriz ation(String payload,\n\t\t\t@PathParam("tenantId") String deviceId,\n\t\t\t@PathP aram("realmName") String realmName) throws Exception {\n\t\tJSONObject returnJson = (JSONObject) JSON.parse(CHALLENGE JSON);\n\t\treturn returnJson;\n\t\n\t@POS T\n\t@Consumes ("application\/json")\n\t@Path("\/{tenantId}\/customAu thRealm 3\/handleChallengeAnswer")\n\t@Produces(MediaType.APPLICATION JSON)\n\t public JSONObject handleChllengeAnswer(String payload,\n\t\t\@PathParam("tenantId&am p;quot;) String deviceId,\n\t\t\@PathParam("realmName") String realmName) thro ws Exception {\n\t\n\t\JSONObject userStoreJson = (JSONObject) JSON.parse(USER_STORE_JSO N);\n\t\tJSONObject failedResponseJson = (JSONObject) JSON.parse(FAILURE_JSON);\n\t\t\n\t\tif(pa yload == null || payload.isEmpty()) {\n\t\treturn failedResponseJson;\n\t\t}\n\t\tJSONObject payloadJso n = (JSONObject) JSON.parse(payload);\n\t\tJSONObject challengeAnswer = (JSONObject) payloadJs on.get("challengeAnswer");\n\t\t\n\t\tif (challengeAnswer == null) {\n\t\t\return faile dResponseJson;\n\t\t\n\t\t\n\t\t\string userName = (String) challengeAnswer.get("userName& amp;quot;);\n\t\tString password = (String) challengeAnswer.get("password");\n\t\t\ n\t\tif (userName == null || userName.isEmpty() || password == null || password.isEmpty()) {\n\t\treturn f ailedResponseJson;\n\t\t]\n\t\tn\t\tif (userStoreJson.containsKey(userName)) {\t\n\t\tJSONObject userI nfoJson = (JSONObject) userStoreJson.get(userName);\n\t\t\String userPassword = (String) userInfoJs on.get("password");\n\t\t\String userDisplayName = (String) userInfoJson.get(&a mp;quot;displayName");\n\t\t\n\t\tif (password.equals(userPassword)) {\n\t\t\t\JSONObject returnJson = new JSONObject();\n\t\t\t\JSONObject userIdentityJson = new JSONObject();\n\t\t\tuserI dentityJson.put("userName", userName);\n\t\t\t\tuserIdentityJson.put("di splayName", userDisplayName);\n\t\t\t\t\treturnJson.put("status", "success");\n\t\t\treturnJson.put("userIdentity", userIdentityJ e Localstore adapter contains few http API's to perform some basic operations like Add, Up date, Create and Delete in client application. T\n\t@Path("\/getAllItems")\n\tpublic String getAllItems() throws IOException{\n\t\ti nit();\n\t\tJsonArray jsonArray = new JsonArray();\n\t\tfor(Object key : props.keySet()){\n\t\tjsonArray.ad d(parser.parse(props.getProperty((String) key)).getAsJsonObject());\n\t\t}\n\t\treturn jsonArray.toString() ;\n\t}\n\n\t@PUT\n\t@Path("\/addItem")\n\tpublic void addItem(String itemJson) \n\ thtthrows IOException, URISyntaxException{\n\thtry{\n\thtinit();\n\thtinit newKey = props.keySet().size()+1;\n\t\tprops.put(String.valueOf(newKey), itemJson);\n\t\t\tURL url = this.getClass().getClassLoader() .getResource("data.properties"); \n\t\t\tFile file = new File(url.toURI().getPath());\n\t \t\tFileOutputStream foStream = new FileOutputStream(file);\n\t\tprops.store(foStream, "savi ng new item");\n\t\t\foStream.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("\/addAllItems")\n\tpublic String addAllItems(String itemsJson) \n\t\t\throws URISyntaxException, IOException{\n\t\ttry{\n\t\thrinit();\n\t\t\tclearAllData();\n\t\t\JsonArray jsonArr = parser.parse(itemsJson).getAsJsonArray();\n\t\t\for(int i=0;i<jso $nArr.size();i++) \ln t t t t props.put(String.valueOf(i+1), jsonArr.get(i).toString()); n/t/t t n/t t url = this.$ getClass().getClassLoader().getResource("data.properties"); \n\t\t\file fi le = new File(url.toURI().getPath());\n\t\t\tFileOutputStream foStream = new FileOutputStream(file);\n\t\t\ tprops.store(foStream, "saving new item");\n\t\t\foStream.close();\n\t\t\tr eturn ";true";\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@Path(& amp;quot;\/clearAll& amp;quot;)\n\tpublic String clearAllData() \n\t\throws MissingConfigurationOptionException, URISyntaxExceptio n, IOException{\n\t\t\tinit();\n\t\t\tprops.clear();\n\t\t\tSystem.out.println("Size: "+pro ps.size());\n\t\tURL url = this.getClass().getClassLoader().getResource("data.properties&am p;quot;); \n\t\tFile file = new File(url.toURI().getPath());\n\t\tFileOutputStream foStream = new FileOut putStream(file);\n\t\trops.store(foStream, "clearing all data");\n\t\trops.store(foStream.clos e();\n\t\treturn "cleared";\n\t}\n<\/pre>\n<\/li> Add TAI Extension in the follo wing path of server directory server/usr/vextensions

server/vextensions Extension Link : Download the extension n.zip from here<\/a>\n<\/li>\nAdd TAI Security constraint in web.xml file for both the projects.\n<pre class=\"brush: xml; title: ; notranslate\"><security-constraint&gt;\n\ \t&lt;web-resourcecollection>\n \t &|t;web-resource-name>LocalstoreApplication&|t;\/web-re \t <url-pattern&gt;\/apps\/*&lt;\/url-pattern&gt;\/n \t& source-name>\n amp;lt;\/web-resource-collection>\n \t<auth-constraint&gt;\n name&:gt:TAIUserRole&:lt:\/role-name&:gt:\n \t&:lt:\/auth-constraint&:gt:\n&am p;lt:\security-constraint&qt:\n<:security-role id="SecurityRole TAIUserRole& <role-name&gt;TAIUserRole&lt;\/role-name&gt;\n&lt;\/s ecurity-role><\/pre>\n<\li>\nAdd OAuthTai feature in server.xml\n<pre class=\"brush: plain; title:; notranslate\"><feature&gt;usr:OAuthTai-1.0&lt;\/feature&gt;<\/pre>\n<\/li> Protect the Url's using TAI by adding following code in server.xml\n :xml; title: ; notranslate\"> <usr_OAuthTAI id=&quot;myOAuthTAI&quot; realmName= "imfRealm">\n\t\t<securityConstraint httpMethods=&quot;GET, POST" securedURLs=" \/ LocalstoreAdapter\/*" \/ > \n\t\t<se curityConstraint httpMethods="GET, POST" securedURLs=" \(\screen{VCUSTON} \) custom-o auth-java\/*"\/>\n\t<\/usr_OAuthTAl&gt;\n\n &lt;webApplication id= "custom-oauth-java" location="custom-oauth-java.war" nam e="custom-oauth-java"&qt;\n <application-bnd&qt;\n\t\t&lt ;security-role name="TAIUserRole"&qt;\n\t\t\<special-subject type=&a mp;quot;ALL_AUTHENTICATED_USERS"V>\n\t\t<Vsecurity-role&gt;\n\t& amp;lt;\/application-bnd> \n\t<\/webApplication&gt; \n\t &lt;webApplication id ="LocalstoreAdapter" location="LocalstoreAdapter.war" nam e="LocalstoreAdapter">\n <application-bnd&gt;\n\t\t& :ht;security-role name="TAIUserRole">\n\t\t\amp;lt;special-subject type= "ALL AUTHENTICATED USERS"\/\&qt;\n\t\t<\/security-role&qt;\n \t&|t;\/application-bnd> \n\t&|t;\/webApplication><\/pre>\n<\/li>\nSpecify th e IMF Auth Url inside Server.env file in liberty.\nre class=\"brush: xml; title: ; notranslate\">imfService Url=https:\/vimf-authserver.ng.bluemix.net\/imf-authserver<\/pre>\n<\/li>\nCreate a server package which contains above two applications using following command.\npre class=\"brush: plain; title: ; notr anslate\">.\server package \${server_name} --include=usr<\/pre>\n<\/li> Push the newly created s erver package to bluemix using following command.\ncf p ush \${app_name} -p \${path_to_server_package_zip}\n<\/li>\n<h3>Advance Mobile Acce ss service<\\h3>\n\nBind the pushed application to Advance Mobile Access Service.\n<a hr ef=\"https:\/\developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-S hot-2015-07-17-at-3.28.04-pm.png\"><\/a >\n<\/li>\nRegister your client application in AMA dashboard. For more info refer documentation : <

a href=\"https:\//www.ng.bluemix.net\/docs\/services\/mobileaccess\/index.html\" target=\"_blank\">click here<\/a>\n<\a>\n<\li>AMA provides Facebook, Google, or a custom identity provider to authenticate acc ess to protected resources. Add Custom identity provider feature as it can be migrated to MFPF and sp ecify the corresponding jax-rs custom authentication application url and realm name.

\/>\n<imq src=\"https:\/\/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\" wi dth=\"955\" height=\"375\" class=\"alignnone size-full wp-image-14890\" \/><\a>\n<\/li>\nAdd the fol lowing code inside didFinishLaunchingWithOptions function in AppDelegate of client application which will register the realm and initialize connection with Bluemix Application.\n<pre class=\"brush: plain; title: ; notranslate\"> IMFClient.sharedInstance().registerAuthenticationDelegate(customAuthDelegate, forRe alm: "customAuthRealm_3")\nIMFClient.sharedInstance().initializeWithBackendR oute("https:\//parkstore.mybluemix.net", backendGUID: "5e3ad88ddd48-469d-b46f-2c4ad66b5345")\n<\li>\nThe following is the sample code to inv oke the Rest url's in client application.\nvar r $equest: IMFResourceRequest = IMFResourceRequest(path: \& amp; quot; https: \\ \lor \lor parkstore.mybluemix.n$ et/LocalstoreAdapter/apps/5e3ad88d-dd48-469d-b46f-2c4ad66b5345/localstore/getAllItems&qu ot:, method: &:guot:GET&:guot;)\n request.sendWithCompletionHandler { (wlResponse:IM FResponse!, err:NSError!) -&qt; Void in\n<\li>\n<\vul>\n<h3>Push Service for iOS 8<\h3>\ nNBind the application with Push Service for iOS 8
br\/>\n<\/a>\n<\/li> Configure Apple Push Notificat ion service (APNs) which requires Apple Developer Account and Generate pl2 certificates. Documentati on link : click here<\/a>\n<\/li>\nUpload the generated pl2 certificate in Push service dashbo ard\n<\ /a>\n<\li>Add the following code inside didFinishLaunchingWithOptions function in AppDelegate of client application which will register notifications in client app.\nnpre class=\"brush: plain; title: ; notra nslate\"> let notificationTypes: UIUserNotificationType = UIUserNotificationType.Badge | UIUserNotifica tionType.Alert | UIUserNotificationType.Sound\n let notificationSettings: UIUserNotificationSettings = UIUserNotificationSettings(forTypes: notificationTypes, categories: nil)\n terUserNotificationSettings(notificationSettings)\n application.registerForRemoteNotifications()</pr e>\n<\/li>\nhhholi>Add the following code inside didRegisterForRemoteNotificationsWithDeviceToken func tion in AppDelegate of client application which will register pushclient and subscribe to tag in client app.\ nnpre class=\"brush: plain; title: ; notranslate\">IMFPushClient.sharedInstance().registerDeviceToken(d if error != nil {\n eviceToken, completionHandler: { (response, error) -> Void in\n println("Error during device registration \\(error.description)"\\n }\n else {\n println("Response during device registration json: \\(response.responseJs on.description)")\n var tags = ["parkstore"]\n ushClient.sharedInstance().subscribeToTags(tags, completionHandler: { (response:IMFResponse!, err: NSError!) -& amp;gt; Void in\n if err != nil {\n println("There was an error while subscribing to tag")\n }else{\n println(" Successfully subscribe to tag parkstore&:guot:)\n $<\pre>\n<$ }\n })\n \li>\n<|i>Add the following function inside Appdelegate which triggers when push notification arrived in client app.\npre class=\"brush: plain; title: ; notranslate\">func application(application: UIApplication, di dReceiveRemoteNotification userInfo: [NSObject : AnyObject]) {\n println("Got remote N otification. Data: \\(userInfo.description\)"\\n let info = userInfo as NSDictionary\n

data - info objectForKey/&amn:quot:ane&amn:quot:\2 objectForKey/&amn:quot:alert&amn:quot:\ ael N

```
uata = \frac{1}{2} uata 
                              let userData = data.objectForKey("body") as! String\n
ertView = UIAlertView(title: & amp;quot; WishList! & amp;quot;, message: & amp;quot; \\
(userData)", delegate: nil, cancelButtonTitle: "OK")\n
                                                                                                                                                   alertView.sho
               \n\<\pre>\n<\li>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>\n<\ul>
p>Add the following Code snippets to the existing Bluemix Client Application and name the application
with same name which you have registered in Advance Mobile Access Dashboard.
d the following code inside didFinishLaunchingWithOptions function in AppDelegate of client application
which will register the realm and initialize connection with Bluemix Application.\n<pre class=\"brush: plai
n; title:; notranslate\"> IMFClient.sharedInstance().registerAuthenticationDelegate(customAuthDelegate
, forRealm: "customAuthRealm_3")\nIMFClient.sharedInstance().initializeWithBac
kendRoute("https:\/\parkstore.mybluemix.net", backendGUID: "5e3ad8
8d-dd48-469d-b46f-2c4ad66b5345&amp:quot:)\n<\li>\li>\n<\li>The following is the sample code to i
nvoke the Rest url&amp:#8217:s in client application.\npre class=\"brush: plain; title: ; notranslate\">va
r request: IMFResourceRequest = IMFResourceRequest(path: & amp;quot;https:\/\parkstore.mybluemix
.net/LocalstoreAdapter/apps/5e3ad88d-dd48-469d-b46f-2c4ad66b5345/localstore/getAllItems&
                                                                                   request.sendWithCompletionHandler { (wlResponse:
quot:, method: &amp:quot:GET&amp:quot:)\n
shLaunchingWithOptions function in AppDelegate of client application which will register notifications in
client app.\npre class=\"brush: plain; title: ; notranslate\"> let notificationTypes: UIUserNotificationType
e = UIUserNotificationType.Badge | UIUserNotificationType.Alert | UIUserNotificationType.Sound\n
let notificationSettings: UIUserNotificationSettings = UIUserNotificationSettings(forTypes: notificationTy
pes, categories: nil)\n
                                                       application.registerUserNotificationSettings(notificationSettings)\n
application.registerForRemoteNotifications()\n<\li>\nAdd the following code inside didRegist
erForRemoteNotificationsWithDeviceToken function in AppDelegate of client application which will regi
ster pushclient and subscribe to tag in client app.\npre class=\"brush: plain; title: ; notranslate\">IMFPu
shClient.sharedInstance().registerDeviceToken(deviceToken, completionHandler: { (response, error) -&
amp:gt: Void in\n
                                        if error != nil {\n
                                                                                  println(&amp:guot:Error during device registration \\
(error.description)")\n
                                                                 }\n
                                                                                  else {\n
                                                                                                              println("Response duri
ng device registration json: \\((response.responseJson.description)\\(\)"\\\(\)
                                                                                                                                                    var tags = [
"parkstore"]\n
                                                                          IMFPushClient.sharedInstance().subscribeToTags(tags,
completionHandler: { (response:IMFResponse!, err:NSError!) -&qt; Void in\n
                                                                                                                                                          if err !=
nil {\n
                                 println("There was an error while subscribing to tag")\n
}else{\n
                                     println("Successfully subscribe to tag parkstore")\n
}\n
                     })\n
                                      }<\/pre>\n<\/li>Add the following function inside Appdelegate which trigg
ers when push notification arrived in client app.\npre class=\"brush: plain; title: ; notranslate\">func app
lication(application: UIApplication, didReceiveRemoteNotification userInfo: [NSObject : AnyObject]) {\n
println("Got remote Notification. Data: \\(userInfo.description)"\\\n
                                                                                                                                                      let info = us
erInfo as NSDictionary\n
                                                let data = info.objectForKey("aps")?.objectForKey(&
                                                                                    let userData = data.obiectForKev(&amp:guot:bodv&
amp;quot;alert") as! NSDictionary\n
amp:quot:) as! String\n
                                              let alertView = UIAlertView(title: & amp;quot; WishList! & amp;quot;, message
: "\\(userData)", delegate: nil, cancelButtonTitle: "OK"\\n
alertView.show()\n
                                      \n\<\/pre>\n<\/li>The following are the screenshots of client 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\/uploads\/sites\/
32\/2015\/07\/IMG 0020-169x300.jpg\" alt=\"IMG 0020\" width=\"169\" height=\"300\" class=\"alignnone
size-medium wp-image-14917\" \/><\a><a href=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-c
ontent/uploads/sites/32/2015/07/IMG 00211.jpg\"><img src=\"https://developer.ibm.com/mobilefir
stplatform/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:\/\/devel
oper.ibm.com/mobilefirstplatform/wp-content/uploads/sites/32/2015/07/IMG 0025.jpg\"><img src=\
"https://developer.ibm.com/mobilefirstplatform/wp-content/uploads/sites/32/2015/07/IMG 0025-1
69x300.jpg\" alt=\"IMG 0025\" width=\"169\" height=\"300\" class=\"alignnone size-medium wp-image-1
4920\" \/><\a><a href=\"https:\/\developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\
/2015\/07\/IMG 0024.jpg\"><img src=\"https:\/\/developer.ibm.com\/mobilefirstplatform\/wp-content\/uplo
ads/sites/32/2015/07/IMG 0024-169x300.jpg\" alt=\"IMG 0024\" width=\"169\" height=\"300\" class=\
"alignnone size-medium wp-image-14919\" \/><\a><a href=\"https:\/\developer.ibm.com\/mobilefirstplat
```

torm/wp-content/uploads/sites/32/2015/07/IMG 0026.jpg\"><\a>\n<\/li>\n<\/ul>\n <h2>Migration to On-Prem</h2>\n<h3 id=\"configureclient\">Migration of Client Application</h3>\n Migration of Client Application includes following two steps Client App Migration<\/li>\n<h3 id=\"cocoapods\">Configuring Cocoapods<\/h3>\nIf CocoaPods has not been installed on a specific computer:\n\nFollow the "Getting Started&am p;#8221; guide for CocoaPods installation: http://guides.cocoapods.org/using/getting-started.html >\nOpen "Terminal" at the installation location and run the "po d init" command<\/li>\nThe following steps assume that the client application is working with CocoPods. If not, follow this & amp;#8220; Using CocoaPods& amp;#8221; documentation: click here<\/a> \nIn both cases, the instructions below explain how to edit the "Podfile̶ 1; file.\n\nOpen the "Podfile" file located in the root of your XCod e project in a favourite text editor.Comment out or remove the existing content.Add the following lines:\nline .com/imflocalsdks/imf-client-sdk-specs.git'\npod 'IMFCompatibility'<\/pre>\n<\/li>\nOpen & amp;#82 20;Terminal" at the location of "Podfile".<\/li>\nVerify that the XCode project is closed.NRun the "pod install" command.N >\nOpen the [MyProject].xcworkspace file in XCode. This file is located side by side with [MyProject] .xcodeproj.

-\nAn usual CocoaPods-based project is managed as a workspace containing the appl ication (the executable) and the library (all project dependencies brought by the CocoaPods manager). \nIn Xcode's Build Settings, search for "Other Linker FlagsR 21; and insert \${inherited} (if -ObjC is defined in this field, you can just delete it, since it is configured in t he CocoaPod project).
N<h3>Client App Migration
NNNN dependency imports like\nre class=\"brush: plain; title: ; notranslate\">#import <IMFCore\/IMF Core.h&qt;\n#import <IMFPush\IMFPush.h&qt;<\/pre>\nReplace the above import s with \nclass=\"brush: plain; title: ; notranslate\">#import <IMFCompatibility\IMFCom patibility.h&qt;\n<\li>Look for a call to the &#8220;initializeWithBackendRoute&a mp;#8221; method and replace the route URL with your on-premise server URL. For example:\nor cl ass=\"brush: plain; title:; notranslate\">IMFClient.sharedInstance().initializeWithBackendRoute(&q uot;https:\//parkstore.mybluemix.net", backendGUID: "5e3ad88d-dd48-469d-b46f-2c4ad66b5345"\nshould be replaced with your on-premise MFP server URL nnpre class=\"brush: plain; title: ; notranslate\">IMFClient.sharedInstance().initializeWithBackendRoute("http:\/\localhost:10080\/ParkStoreMFP", backendGUID: "5e3ad88d-dd 48-469d-b46f-2c4ad66b5345"\nNote, that backendGUID parameter is ignored an d can be empty. Look for all instantiations of IMFResourceRequest class and update it or all instantiations of IMFResourceRequest class and update the request URL with absolute or relative path to the resource. For example:\npre class=\"brush: plain; title: ; notranslate\">var request: IMFRes ourceRequest = IMFResourceRequest(path: & amp;quot;https:\//parkstore.mybluemix.net/LocalstoreAd apter\/apps\/5e3ad88d-dd48-469d-b46f-2c4ad66b5345\/localstore\/getAllItems", method: &a mp:guot:GET&:guot:)\nshould be replaced with\npre class=\"brush: plain: title:: n otranslate\">var request: IMFResourceRequest = IMFResourceRequest(path: "http:\//localh ost:10080\ParkStoreMFP\adapters\LocalstoreAdapter\localstore\getAllItems", method: &a mp;quot;GET")\n<\li>Add the following code inside didRegisterForRemoteNotifi cationsWithDeviceToken function in Appdelegate of Client application.\npre class=\"brush: plain; title: ; notranslate\"> WLPush.sharedInstance().tokenFromClient = deviceToken.description<\pre>\n<\/li> >All on-premise applications require the "worklight.plist" file to be present in th e application resources. In the <code>IBMMobileFirstPlatformFoundationNativeSDK<\/code> pod we su pply a file named sample.worklight.plist<\/strong>.\n\nLocate the &#8220;sample. worklight.plist" file in the â€~IBMMobileFirstPlatformFoundationNativeSDK' pod. Copy this file to the parent (application) project and rename it to "worklight.plist 221;..<\/li>\nEdit the &#8220;worklight.plist&#8221; file by setting the &#8220;applicat ion id" key to the name of your application deployed to the on-premise MFPF server <\ul>\n<\\li>\n<\\ol>\n<\Adapter<\\h3>\n \nTo migrate JAX-RS application to on-prem (MobileFirst Foundation) server we need to do the following steps for server:\n Create MobileFirst Project & amp;#8211:& amp;at; Create native API

app for iOS<br \/>\n â€<â€<
br \/>\n<\a><\p>\n <img src=\"https:\//developer.ibm.com/mobilefirstplatform/wp-content/uploads/sites//32/2015//07/Sc</pre> reen-Shot-2015-07-12-at-6.51.13-pm.png\" alt=\"Screen Shot 2015-07-12 at 6.51.13 pm\" width=\"598\" height=\"590\" class=\"alignnone size-full wp-image-14818\" \/><\/a><\/p>\n<\/a><\/li>Add t wo adapters for Custom Authentication and Localstore and migrate the JAX-RS code as shown in the f ollowing example.\n<\ol>\n<\p>Copy the JAX-RS BlueMix code and paste it in the newly created L ocalstore Java adapter JAX-RS file.
Add and remove the following changes in your adapter c ode.\nremove <code>\{tenantId}\</code><\li>\nremove the <code>@PathParam -> PathParam(\"tenantId\") String deviceId<\code> and <code>@PathParam(\"realmName\") St ring realmName<\/code><\/li>\nAdd scope to the all http api resource <code>@OAuthSecurity (sco pe=\"customAuthRealm 3\")<\/code><\/li>\nThe code looks like the following<\/p>\n\n\p>\n ss=\"brush: plain; title: ; notranslate\">\n\t@GET\n\t@OAuthSecurity (scope="customAuthRe alm_3")\n\t@Path("\/getAllItems")\n\tpublic String getAllItems() throws MissingConfigurationOptionException{\n\t\tinit();\n\t\tJsonArray | sonArray = new JsonArray();\n\t\tfor(Obj ect key: props.keySet()){\n\t\t\sonArray.add(parser.parse(props.getProperty((String) key)).getAsJsonO bject());\n\t\treturn jsonArray.toString();\n\t}\n\n\t@PUT\n\t@OAuthSecurity (scope="cus tomAuthRealm_3")\n\t@Path("\/addItem")\n\tpublic void addItem(String itemJson) \n\t\tthrows MissingConfigurationOptionException, URISyntaxException, IOException{\n\t\tr y{\n\t\tinit();\n\t\tint newKey = props.keySet().size()+1;\n\t\t\props.put(String.valueOf(newKey), itemJs on);\n\t\t\tURL url = this.getClass().getClassLoader().getResource("data.properties&quo t;); \n\t\t\File file = new File(url.toURI().getPath());\n\t\tFileOutputStream foStream = new FileOutputStr eam(file);\n\t\tprops.store(foStream, & p;quot;saving new item");\n\t\tfoStream.close();\n\ n/t/t}catch(IOException ioe){\n\t\t\tioe.printStackTrace();\n\t\t}\n\n\t@POST\n\t@OAuthSecurity (sc ope="customAuthRealm 3")\n\t@Path("\/addAllItems")\n\tpu blic String addAllItems(String itemsJson) \n\t\throws MissingConfigurationOptionException, URISynta xException, IOException{\n\t\ttry{\n\t\t\tinit();\n\t\t\tclearAllData();\n\t\t\JsonArray jsonArr = parser.parse(itemsJson).getAsJsonArray();\n\t\tfor(int i=0;i<jsonArr.size();i++){\n\t\t\tprop s.put(String.valueOf(i+1), jsonArr.get(i).toString());\n\t\t\t\t\URL url = this.getClass().getClassLoader ().getResource("data.properties"); \n\t\tFile file = new File(url.toURI().getPath());\n\t\tFileOutputStream foStream = new FileOutputStream(file);\n\t\t\tprops.st ore(foStream, "saving new item");\n\t\tfoStream.close();\n\t\t\treturn &a mp;amp;quot;;true";\n\t\t}catch(IOException ioe){\n\t\t\tioe.printStackTrace();\n\t\t}\n\t\tret urn "false";\n\t}\n\n\t@DELETE\n\t@OAuthSecurity(enabled=false)\n\t@ Path("\/clearAll"\/n\tpublic String clearAllData() \n\t\\throws MissingConfiguration OptionException, URISyntaxException, IOException{\n\t\t\tinit();\n\t\t\tprops.clear();\n\t\t\tSystem.out.pri ntln("Size: "+props.size());\n\t\t\URL url = this.getClass().getClassLoader().getRe source("data.properties"); \n\t\tFile file = new File(url.toURI().getPath());\n\t\tFile OutputStream foStream = new FileOutputStream(file);\n\t\t\props.store(foStream, & mp;quot;clearing al I data");\n\t\tfoStream.close();\n\t\treturn "cleared";\n\t}\n<\/pre>\n<h3 id=\"configoauth\">Configuring Custom-OAuth<\/h3>\n\nAdd realm with same name you had on BlueMix and login module to the authenticationConfig.xml.\npre class=\"brush: xml; title: ; notranslate\" ><realm name=&quot;customAuthRealm_3&quot; loginModule=&quot;customAut hLoginModule 3">\n<className&gt;com.worklight.core.auth.ext.Custom IdentityAuthenticator<\/className&gt;\t\n&lt;parameter name=&quot;providerUrl& amp;quot; value="http:\/\localhost:10080\/ParkStoreMFP\/adapters\/Customauth"\ />\n<\realm&gt;\n\n&lt;loginModule name=&quot;customAuthLoginModul e 3" expirationInSeconds="3600"&qt;\n<className& gt;com.worklight.core.auth.ext.CustomIdentityLoginModule<\/className&gt;\n&lt;\/logi

nModule><\/pre>\n<\/li>\nAdd Custom-oauth Realm in userIdentityRealms in Application De scriptor file of iOS Native API\nre class=\"brush: xml; title: ; notranslate\"><userIdentityRealms >customAuthRealm_3<\/userIdentityRealms&gt;<\/pre>\n<\/li>\n<\/ul>\n<h3 id=\"con figurepush\">Configuring Push Capability<\/h3>\n\nAdd apns p12 certificate which is generated from Apple Developer Account under iOS Native API Folder\nhttps://developer.ibm.com\ /mobilefirstplatform/wp-content/uploads/sites/32/2015/07/Screen-Shot-2015-07-12-at-6.58.03-pm.p ng\"><\/a>\n<\/li> Add Push config uration in Application Descriptor file of iOS Native API and include the password of added apns certifica te.\n<pushSender password=&quot;password& amp;quot;\/>\n<tags&gt;\n &lt;tag&gt;\n &lt;name&gt;parkstor e<\/name&gt;\n &lt;\/tag&gt;\n&lt;\/tags&gt;<\/pre>\n<\/li> HTTP Push Adapter with following function code which will send the user push notification to the device s which is subscribed to tag "parkstore".\npre class=\"brush: xml; title: ; notra nslate\">function sendTagNotification(notificationText) {\n var notificationOptions = {};\n notification Options.message = {};\n notificationOptions.target = {};\n\n notificationOptions.message.alert = notif icationText;\n notificationOptions.target.tagNames = ["parkstore"];\n\n ver.sendMessage("ParkStoreMFP", notificationOptions);\n\n return {\n It: "Notification sent to users subscribed to the tag parkstore."\n \{\n}<\/pre>\n<\/l i>\n<\vul>\nBy performing above steps one can easily run iOS app built for Bluemix on MobileFirst P latform and following are the links to samples.
\n<h3 id=\"sample\">Sample and Source Code<\h3</p> >\nBluemix Server : Parkst ore bluemix server<\/a><br \/>\nBluemix Client : Parkstore bluemix<\a><br \\>\nMFP Server : Parkstore mfp server<\/a><br \/>\nMFP Client :Parkstore mfp<Va><Vp>\nThe post Try on Bluemix and migrate to on-prem MobileFirst Platform appeared first on 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

The attached sample (https://github.com/MobileFirst-Platform-Developer-Center/JavaAdapters/tree/release71) includes an adapter called RSSReader to test the adapter inside an application.