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 Clie
ntProtocolException, 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 ClientProtocolE
xception, IOException,
 IllegalStateException, SAXException {
  HttpResponse RSSResponse = client.execute(host, reg);
  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 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.

"description": "The post Integrating MQA into Xamarin.Android app<\\a> appeared first on IB M MobileFirst Platform<\\a>.<\\p>",

"encoded": "It all started when I received an email seeking help on using MQA or to be m ore precise integrating MQA into Xamarin based android app. Before jumping into addressing the proble m, let's define MQA.\n<h4>What is MQA?</\h4>\nMQA stands for & amp;#8220;Mo bile Quality Assurance" and is part of the IBM MobileFirst Platform. em>IBM MQA provides line of business professionals and development t eams with insightful and streamlined quality feedback and metrics from both pre-production and productio n, enabling them to prioritize and take action to support a dynamic mobile app strategy. p></blockguote>\nThe Features of MQA are<\p>\n<div style=\"width: 1058px\" class=\"wp-caption al igncenter\"><\a><p class=\"wp-caption -text">Features of Mobile Quality Assurance.</div>\nNote: To understand more about MQA, visit IBM Mobile Quality Assurance<\/a><\p>\nSo, by now we should b e 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>\nXamarin is a platform to create native iOS, Android, Mac and Windows apps in C# . Xamarin.Android allows us to create native Android applications using the same UI controls we would i n Java, except with the flexibility and elegance of a modern language (C#).hAs we are good wit h the definitions, let's address the problem. em in integrating MQA into Xamarin Android app?\nAt the time of this blog post, the av ailable MQA SDKs are iOS native SDK, Android native SDK and Javascript A SDK. e to find a workaround to address this use-case. The initial step is to download the Android MQA SDK an d see what's provided. you can download it from here<\/a>. Once successfully downloaded and unzipped, we should see a jar file namely < em>MQA-Android-library-<version number&qt:.jar<\/em>Â <\/strong>under lib folder.\n<div style=\"width: 634px\" class=\"wp-caption aligncenter\"><\/a>MQA Android SDK<\/p><\/div>\nAs Xamarin is C# based, What can we do with this jar file?
NWe have Xamarin bindings to our rescue, which helps using in consuming .JARs from C#.\nNote:</stro ng> Steps to consume MQA Android JAR in a Xamarin. Android app is mentioned here<\/a><\/p>\n<div style=\"width: 257px\" class=\"wp-caption aligncenter\"><\a>Xamarin binding project with MQA Android .JAR file<\/p> </div>\nThe files of our interest here are MQA-Android-library-2.7.4.jar<\strong> (Version n umber may vary) and Metadata.xml.<\p>\n\nMQA-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. nnMetadata.xml- Allows changes to be made to the final API, such as changing the namespace o f the generated binding.InBased on the errors thrown while building the project, M etadata.xml in my case looks like this\n<metadat a>\n <!--\n This sample removes the class: android.support.v4.content.AsyncTaskLoader. LoadTask:\n <remove-node path=&quot;\/api\/package[@name='android.support.v4.content'] Vclass[@name='AsyncTaskLoader.LoadTask']" V>\n \n This sample removes the met hod: android.support.v4.content.CursorLoader.loadInBackground:\n <remove-node path=&q uot;\/api\/package[@name='android.support.v4.content']\/class[@name='CursorLoader']\/method[@name ='loadInBackground']" \/>\n -->\n\n <remove-node path=&quot;\/a pi\package[@name='ext.com.google.inject.spi']\class[@name='InjectionPoint.Factory.1']"\/&a mp;qt;\n <remove-node path=&quot;\/api\/package[@name='ext.com.google.inject.spi']\/clas s[@name='InjectionPoint.Factory.2']"\/&qt;\n <remove-node path=&quot;\/api \package[@name='com.applause.android.log']\sinterface[@name='LoggerInterface']"\skamp;gt ;\n <remove-node path=&quot;\/api\/package[@name='ext.com.google.inject.internal']& quot;\/>\n <remove-node path=&quot;\/api\/package[@name='ext.com.google.inject. matcher']"\/>\n <remove-node path=&quot;\/api\/package[@name='com.a pplause.android.util']\footnote{\text{class[@name='AbstractRequest']"\footnote{\text{bamp;qt;\n <remove-node pa}} th="\/api\/package[@name='ext.com.google.inject.spi']\/class[@name='Elements.RecordingBin der']\/method[@name='bind' and count(parameter)=1 and parameter[1][@type='ext.com.google.inject.Key ']]"\/>\n\n<attr path=&quot;\/api\/package[@name='com.applause.android. messages']\/class[@name='Message']\/field[@name='message']" name="managed Name">Message1<\/attr&gt;\n&lt;attr path=&quot;\/api\/package[@name='com.applause.android.log']" name="managedName"&qt;lo g<\/attr&gt;\n\amp;lt;\/metadata&gt;\n\n<\/pre>\nOnce all the errors are fixed and yo ur binding project builds successfully, add a new Xamarin Android project (if you haven't add ed yet). Now, add MQA binding project reference in our Xamarin android app. Note:<\stro ng> Both your binding project and Xamarin. Android project should be of same target fram ework.Ä You can verify this by right clicking on your project -& ptions -& amp;gt; Gener al.<\/p>\n<div id=\"attachment_83\" style=\"width: 270px\" class=\"wp-caption aligncenter\"><\/a>Xa marin Android project with added reference to MQA<\p><\/div>\nNow, let&#8217;s start MQA a ndroid session in our Count. Android app. Before doing this, we should create a MQA service on IBM Blue mix. You can follow the instructions mentioned at Getting started with Mobile Quality Ass urance- Bluemix<\/a>Â or watch this video.\n<iframe class='youtube-player' type='text\/html' width='980' height='582' src='https:\/\/ www.youtube.com/embed/zHRfGatcKPM?version=3&rel=1&fs=1&show search=0&showinfo=1&iv load policy=1&wmode=transparent' frameborder='0' allowfullscreen='true'><\iframe><\/span><\/p>\nStarting a\(\text{A}\) <spa n id=\"d6087e24\" class=\"ph\">Mobile Quality Assurance<\/span><\/span>\A session with the Android SD K entails three steps. First, build a configuration to define howA Mobile Quality Assurance<\/span><\/span>A works with your app. Secon d, start the session itself. Third, add tracking to your activities. Open MainActivity.cs<\/strong> fil e (Android Project) and paste the code provided below\npre class=\"brush: csharp; title: ; notranslat e\">using System;\n\nusing Android.App;\nusing Android.Content;\nusing Android.Runtime;\nusing Andro id.Views;\nusing Android.Widget;\nusing Android.OS;\n\/\/MQA references\nusing Com.lbm.Mqa.Config;\ nusing Com.lbm.Mqa;\n\nnamespace Count.Android\n\t\[Activity (Label = & amp;quot;Count.Android& amp;quot;, MainLauncher = true, lcon = & amp;quot;@drawable\/icon")]\n\tpublic class MainActi vity: Activity\n\t\tint count = 1;\n\t\t\v\Use your own generated APP KEY\n\t\tconst string APP KEY=& amp;quot;1g59b7d884f9fdf5426162e5cb1f87a700648bce4fg0g1g379e0d3a";\n\t\tprotected ov erride void OnCreate (Bundle bundle)\n\t\t{\n\t\tbase.OnCreate (bundle);\n\t\t\t\v\MQA Android session c onfiguration \n\t\t\tConfiguration configuration = new Configuration.Builder(this)\n\t\t\t.WithAPIKey(APP KEY) \(\triangle \Provides \) the quality assurance application APP_KEY\n\t\t\.WithMode(MQA.Mode.Qa) \(\triangle \) Select s the quality assurance application mode\n\t\t\t\t.WithReportOnShakeEnabled(true) \WEnables shake rep

```
"guid": {
           "content": "https:\/\developer.ibm.com\/mobilefirstplatform\/?p=16964",
           "isPermaLink": "false"
         },
         "link": "https:\lor\lordeveloper.ibm.com\lormobilefirstplatform\lor2015\lor09\lor01\lorintegrating-mqa-into-xam
arin-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 MobileFirst Platform< \lor 0a> appeared first on IBM MobileFirst Platform< \lor 0a>.< \lor p>",

"encoded": "Contributed By: Chethan Kumar SN (chethankumar.sn@in.ibm.com) and Vitt al Pai (vittalpai@in.ibm.com)\nWith the release of MobileFirst Platform v7.1, one can now migrat e any existing iOS app built for MobileServices on Bluemix to MobileFirst Platform with just a handful of si mple steps.
To elucidate the process, lets look at how to migrate a simple Bluemix iOS app. >\nTo migrate an existing iOS app built for MobileServices on Bluemix to run on MobileFirst Platform, follow the steps below.\n\nExisting Bluemix Server Application figureclient\">Migration of Client Application<\/a><\/li>\nMigration of JAX-RS Application to JAVA Adapter<\a><\li>\n<|i>\a> href=\"#configoauth\">Configuring Custom-OAuth<\a><\li>\li >\nConfiguring Push Capability<\/a><\/li>\nSamp le and Source Code<\/a><\/li>\n<\/ul>\n<h2 id=\"migrateexisting\">Existing Bluemix Server Application<\/ h2>\nThe Bluemix app has the following functionality:\n on stores a list of items and provides a way to add more items to the list. Each item can able to store Nam e, Store, Price and image of the product. The App's are protected by Custom Authenticator v ia AMA security service provided by bluemix.On the server side, the App contains a JAX-RS cl ass to store and manipulate the data. It also contains the server side AMA security implementation. <\ul>\nOn BlueMix we have application with the following configuration:<\p>\n\n\nLiberty Runti

me: which used to run JAX-RS application on Bluemix gives mobile application security and monitoring functionalityIn-Push Service for iOS 8 : which pro vides the capability to use iOS Push features<\/li>\n<\/l>\n<\/l>\n<\/l>\n<\/l>\n<\/l>\n<\/l>\n<\\n<\/l></ur> y contains two projects with JAX-RS service (i.e Custom-oauth-java for Custom Authentication and Local storeAdapter for storing items). The service include the protected resource and the custom identity provid er code. The liberty server is configured with TAI.\n<\/li>\r\cli>Trust Association Interface (TAI) is a servic e provider API that enables the integration of third-party security services with a Liberty profile server. For more info on TAI: click here<\/a> \n<\li>\n\cli>\text{The custom identity provider authenticates a user by sending challenges to the client. Howev er, custom identity providers do not communicate directly with clients. They send challenges and receive responses to the challenges by means of the Advanced Mobile Access service. When a custom identity p rovider successfully authenticates the user, it provides the user identity information to Advanced Mobile A ccess. For more information on custom authentication refer bluemix documentation : click here<\/a>\nThe custom identity provider code is defined by two http API:<\ /p>\nclass=\"brush: plain; title: ; notranslate\">\/startAutorization<\/pre>\n and\n<pre class=\"brush" h: plain; title: ; notranslate\">\/handleChallengeAnswer<\/pre>\/nclass=\"brush: java; title: ; notranslat e\"> @POST\n\t@Consumes ("application\json")\n\t@Path("\/{tenantId}\/ customAuthRealm_3\/startAuthorization")\n\t@Produces(MediaType.APPLICATION_JSON)\n\ tpublic JSONObject startAuthorization(String payload,\n\t\t\@PathParam("tenantId") String deviceId,\n\t\t\@PathParam("realmName") String realmName) throws Exce ption {\n\t\JSONObject returnJson = (JSONObject) JSON.parse(CHALLENGE_JSON);\n\t\treturn returnJ son;\n\t\n\t@POST\n\t@Consumes ("application\/json")\n\t@Path("\/{ tenantId}\/customAuthRealm 3\/handleChallengeAnswer")\n\t@Produces(MediaType.APPLIC ATION_JSON)\n\tpublic JSONObject handleChllengeAnswer(String payload,\n\t\t\@PathParam(&q uot;tenantId") String deviceId,\n\t\t@PathParam("realmName") String re almName) 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(payload == null || payload.isEmpty()) {\n/t/t/treturn failedResponseJson:\n/t/t}\n/t/tJSONObject pa yloadJson = (JSONObject) JSON.parse(payload);\n\t\tJSONObject challengeAnswer = (JSONObject) pay loadJson.get("challengeAnswer");\n\t\t\n\t\tif (challengeAnswer == null) {\n\t\tretur me");\n\t\tString password = (String) challengeAnswer.get("password");\n \t\t\n\t\tif (userName == null || userName.isEmpty() || password == null || password.isEmpty()) {\n\t\t\tretur n failedResponseJson;\n\t\t\n\t\tif (userStoreJson.containsKey(userName)) {\t\n\t\t\JSONObject user InfoJson = (JSONObject) userStoreJson.get(userName);\n\t\t\tString userPassword = (String) userInfoJso n.get("password");\n\t\t\String userDisplayName = (String) userInfoJson.get(& quot;displayName");\n\t\t\n\t\t\n(password.equals(userPassword)) {\n\t\t\t\JSONObject retur nJson = new JSONObject();\n\t\t\t\JSONObject userIdentityJson = new JSONObject();\n\t\t\t\tuserIdentity Json.put("userName", userName);\n\t\t\t\tuserIdentityJson.put("displayNa me", userDisplayName);\n\t\t\t\n\t\t\t\treturnJson.put("status", " ;success");\n\t\t\treturnJson.put("userIdentity", userIdentityJson);\n\t\t\t\treturnJson.put("userIdentity") eturn returnJson;\n\t\t\t\t\n\t\t\n\t\treturn failedResponseJson;\n\t}\n<\pre>\nThe Localstore a dapter contains few http API's to perform some basic operations like Add, Update, Create an d Delete in client application.\nre class=\"brush: java; title: ; notranslate\"> @GET\n\t@Path(& ;quot;\/getAllItems")\n\tpublic String getAllItems() throws IOException{\n\t\tinit();\n\t\tJsonArray j sonArray = new JsonArray();\n\t\tfor(Object key : props.keySet()){\n\t\t\tjsonArray.add(parser.parse(props. getProperty((String) key)).getAsJsonObject());\n\t\treturn jsonArray.toString();\n\t\n\n\t@PUT\n\t@P ath(")\addItem")\n\tpublic void addItem(String itemJson) \n\t\t\throws IOException, 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("da ta.properties"); \n\t\tFile file = new File(url.toURI().getPath());\n\t\t\fileOutputStream foStream = new FileOutputStream(file);\n\t\t\props.store(foStream, "saving new item");\n\t\t\f oStream.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 U

RISvntaxException. IOException{\n\t\trv{\n\t\t\tinit():\n\t\t\tearAllData():\n\t\t\JsonArrav isonArr = parser.

 $... \\ option = nonlinear + n$ parse(itemsJson).getAsJsonArray();\n\t\t\ffor(int i=0;i<jsonArr.size();i++){\n\t\t\t\props.put(Stri ng.valueOf(i+1), jsonArr.get(i).toString()):\n\t\t\t\URL url = this.getClass().getClassLoader().getReso urce("data.properties"); \n\t\tFile file = new File(url.toURI().getPath());\n\t\ t\tFileOutputStream foStream = new FileOutputStream(file);\n\t\tprops.store(foStream, "s aving new item");\n\t\t\tfoStream.close();\n\t\t\treturn "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("\/clearAll")\n\tpublic String clearAllData() \n\t\tthrows Mi ssingConfigurationOptionException, URISyntaxException, IOException{\n\t\t\tinit();\n\t\t\tprops.clear();\n\t\t \tSystem.out.println("Size: "+props.size());\n\t\t\URL url = this.getClass().getClassL oader().getResource("data.properties"); \n\t\t\File file = new File(url.toURI().getPath ());\n\t\tFileOutputStream foStream = new FileOutputStream(file);\n\t\t\tprops.store(foStream, & mp;quot ;clearing all data");\n\t\tfoStream.close();\n\t\treturn "cleared";\n\t}\n<\p re>\n<\/li> Add TAI Extension in the following path of server directory server\/usr\/extensions
br \/>\ nTAI Extension Link: Download the extension.zip from here<\/a>\n<\/li> Add TAI Security constraint i n web.xml file for both the projects.\n<security-constr aint>\n \t<web-resource-collection&gt;\n \t <web-resource-name&gt ;LocalstoreApplication<\/web-resource-name&gt;\n \t <url-pattern&gt;\/apps\/ *<\/url-pattern&gt;\n \t&lt;\/web-resource-collection&gt;\n \t<auth-constr aint&qt:\n <role-name&gt;TAIUserRole&lt;\/role-name&gt;\/n It;\auth-constraint&qt;\n<\security-constraint&qt;\n&lt;security-role id=&quot;S ecurityRole TAIUserRole" &qt;\n <role-name&gt;TAIUserRole&lt;\/ro le-name>\n<\/security-role&gt;<\/pre>\n<\/li>\nAdd OAuthTai feature in server.xml \n\nclass=\"brush: plain: title: ; notranslate\"><:feature>usr:OAuthTai-1.0<\/featur e>\n<\li>Protect the Url&#8217;s using TAI by adding following code in server. xml\nre class=\"brush: xml; title: ; notranslate\"> &|t;usr_OAuthTAI id="myOAuthTAI&a mp;quot; realmName="imfRealm">\n\t\t<securityConstraint httpMeth ods="GET, POST" securedURLs="\/LocalstoreAdapter\/*"\/&a mp;qt:\n\t\amp;lt:securityConstraint httpMethods="GET, POST" securedURLs=&a mp;quot;\/custom-oauth-java\/*"\/&qt;\n\t<\/usr OAuthTAl&gt;\n\n &lt; webApplication id="custom-oauth-java" location="custom-oauth-java.war " name="custom-oauth-java"&qt;\n <application-bnd& qt;\n\t\t<security-role name=&quot;TAIUserRole&quot;&gt;\n\t\t\t&lt;special-su bject type="ALL AUTHENTICATED USERS"V>\n\t\t<Vsecurity-role& amp;qt;\n\t<\/application-bnd&qt;\n\t&lt;\/webApplication&gt;\n\t&lt;webApp lication id="LocalstoreAdapter" location="LocalstoreAdapter.war" ot; name="LocalstoreAdapter">\n <application-bnd&gt;\n\t\t &|t;security-role name="TAIUserRole">\n\t\t\&|t;special-subject ty pe="ALL AUTHENTICATED USERS"V&qt;\n\t\t<Vsecurity-role&qt ;\n\t<\application-bnd&gt; \n\t&lt;\webApplication&gt;<\pre>\n<\li>\nSpecify th e IMF Auth Url inside Server.env file in liberty.\nre class=\"brush: xml; title: ; notranslate\">imfServiceUr I=https:\/vimf-authserver.ng.bluemix.net\/imf-authserver<\/pre>\n<\/li>\nCreate a server package whic h contains above two applications using following command.\npre class=\"brush: plain; title: ; notranslate \">.\/server package \${server name} --include=usr<\/pre>\n<\/li>\nPush the newly created server package \${server name} --include=usr<\/pre>\n<\/li> kage to bluemix using following command.\ncf push \${app} name} -p \${path_to_server_package_zip}<\/pre>\n<\/li>\n<\/ul>\n<A3>Advance Mobile Access service<\/h 3>\n\nBind the pushed application to Advance Mobile Access Service.\n<\a>\n<\/li>\nRegister your client application in AMA dashboard. For more info refer documentation : click here<\/a>\n<img src=\"https:\//developer.ibm.com/mobilefirstplatform/wp-content/upl oads\/sites\/32\/2015\/07\/Screen-Shot-2015-07-17-at-3.42.32-pm.png\" alt=\"AMA Client Registration\" wi

dth_\"Q25\" haight_\"459\" alaga_\"alignpana ciza full wa imaga 14009\" \/ 2//a/\a/\lik \n //is \ AMA provid

```
DIVUIQ AIVIA <||>| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 1|| 402 | 402 | 1|| 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 | 402 |
es Facebook, Google, or a custom identity provider to authenticate access to protected resources. Add C
ustom identity provider feature as it can be migrated to MFPF and specify the corresponding jax-rs custo
m authentication application url and realm name.<br/>
-\n<a href=\"https:\/\developer.ibm.com\/mobilefirs"
tplatform/wp-content/uploads/sites/32/2015/07/Screen-Shot-2015-07-17-at-4.03.21-pm.png\"><img s
rc=\"https:\/\developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/Screen-Sh
ot-2015-07-17-at-4.03.21-pm.png\" alt=\"Custom Auth AMA\" width=\"955\" height=\"375\" class=\"alignno
ne size-full wp-image-14890\" \/><\/a>\n<\/li> Add the following code inside didFinishLaunchingWith
Options function in AppDelegate of client application which will register the realm and initialize connection
with Bluemix Application.\nre class=\"brush: plain; title: ; notranslate\"> IMFClient.sharedInstance().regi
sterAuthenticationDelegate(customAuthDelegate, forRealm: & amp;quot;customAuthRealm 3& amp;quot;)
\nIMFClient.sharedInstance().initializeWithBackendRoute("https:\//parkstore.mybluemix.net&a
mp;quot;, backendGUID: "5e3ad88d-dd48-469d-b46f-2c4ad66b5345")\n<\/li>
>\nThe following is the sample code to invoke the Rest url's in client application.\npre c
lass=\"brush: plain; title: ; notranslate\">var request: IMFResourceRequest = IMFResourceRequest(path:
&amp:guot:https:\/\parkstore.mybluemix.net\/LocalstoreAdapter\/apps\/5e3ad88d-dd48-469d-b46f-2c4ad
66b5345\/localstore\/getAllItems&quot:, method: &quot:GET&quot:\\n
                                                                                                               request.sendWi
thCompletionHandler { (wlResponse:IMFResponse!, err:NSError!) -&amp:qt: Void in<\/pre>\n<\/li>\n<\/ul>
\n<h3>Push Service for iOS 8<\/h3>\n\nBind the application with Push Service for iOS 8<br/>br\/>\n
<a href=\"https:\//developer.ibm.com/mobilefirstplatform/wp-content/uploads/sites/32/2015/07/Scree
n-Shot-2015-07-17-at-4.07.01-pm.png\"><img src=\"https:\//developer.ibm.com\/mobilefirstplatform\/wp-c
ontent/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\" \lor > \lor a > n < \forall li > n  C
onfigure Apple Push Notification service (APNs) which requires Apple Developer Account and Generate
pl2 certificates. Documentation link : <a href="https:\/\www.ng.bluemix.net/docs/services/mobilepush/i
ndex.html#certificates\" target=\" blank\">click here<\/a>\n<\/li>\li>\upload the generated pl2 certificate
in Push service dashboard\n<a href=\"https:\/\developer.ibm.com\/mobilefirstplatform\/wp-content\/up
loads/sites/32/2015/07/Screen-Shot-2015-07-12-at-6.47.14-pm.png\"><img src=\"https://developer.ib
m.com/mobilefirstplatform/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-im
age-14816\" \/><\a>\n<\/li>\ndli>Add the following code inside didFinishLaunchingWithOptions function in
AppDelegate of client application which will register notifications in client app.\npre class=\"brush: plain:
title: ; notranslate\"> let notificationTypes: UIUserNotificationType = UIUserNotificationType.Badge | UIUs
erNotificationType.Alert | UIUserNotificationType.Sound\n
                                                                              let notificationSettings: UIUserNotification
Settings = UIUserNotificationSettings(forTypes: notificationTypes, categories: nil)\n
                                                                                                             \n
                                                                                                                      applicatio
n.registerUserNotificationSettings(notificationSettings)\n
                                                                            application.registerForRemoteNotifications()
nction in AppDelegate of client application which will register pushclient and subscribe to tag in client app
.\nre class=\"brush: plain: title:: notranslate\">IMFPushClient.sharedInstance().registerDeviceToken(de
viceToken, completionHandler: { (response, error) -& amp;qt; Void in\n
                                                                                                if error != nil {\n
println("Error during device registration \\((error.description)\")\n
                                                                                                                 }\n
                                                                                                                              els
                  println("Response during device registration json: \((response.responseJson.de
scription)")\n
                                         var tags = [&amp:guot:parkstore&amp:guot:]\n
                                                                                                                IMFPushClie
nt.sharedInstance().subscribeToTags(tags, completionHandler: { (response:IMFResponse!, err:NSError!)
-> Void in\n
                                       if err != nil {\n
                                                                           println("There was an error while
e subscribing to tag")\n
                                                          }else{\n
                                                                                      println("Successfully s
ubscribe to tag parkstore")\n
                                                                }\n
                                                                                })\n
                                                                                              }<\pre>\n<\li>\nAdd th
e following function inside Appdelegate which triggers when push notification arrived in client app.\npre
class=\"brush: plain; title:; notranslate\">func application(application: UIApplication, didReceiveRemoteN
otification userInfo: [NSObject : AnyObject]) {\n
                                                                println(&amp:quot:Got remote Notification. Data: \\(u
serInfo.description)")\n
                                                let info = userInfo as NSDictionary\n
                                                                                                   let data = info.objectFor
Key("aps")?.objectForKey("alert") as! NSDictionary\n
serData = data.objectForKey("body") as! String\n
                                                                                              let alertView = UIAlertView(titl
e: "WishList!", message: "\\(userData)", delegate: nil, cancelBu
ttonTitle: "OK")\n
                                                      alertView.show()\n
                                                                                  grateblu\">Existing Bluemix Client Application<\/h2>\nAdd the following Code snippets to the existing
Bluemix Client Application and name the application with same name which you have registered in Advan
```

ce Mobile Access Dashboard. ions function in AppDelegate of client application which will register the realm and initialize connection wit h Bluemix Application.\nre class=\"brush: plain; title: ; notranslate\"> IMFClient.sharedInstance().registe rAuthenticationDelegate(customAuthDelegate, forRealm: & amp:guot:customAuthRealm 3& amp:guot:)\nl MFClient.sharedInstance().initializeWithBackendRoute("https:\//parkstore.mybluemix.net& ;quot;, backendGUID: "5e3ad88d-dd48-469d-b46f-2c4ad66b5345")<\pre>\n<\li>\n < The following is the sample code to invoke the Rest url&#8217;s in client application.\n<pre>pre clas s=\"brush: plain; title:; notranslate\">var request: IMFResourceRequest = IMFResourceRequest(path: &a mp;quot;https://parkstore.mybluemix.net/LocalstoreAdapter/apps/5e3ad88d-dd48-469d-b46f-2c4ad66 b5345\localstore\getAllItems", method: "GET")\n request.sendWith CompletionHandler { (wlResponse:IMFResponse!, err:NSError!) -&qt; Void in<\/pre>\n<\li>\nAdd the following code inside didFinishLaunchingWithOptions function in AppDelegate of client application wh ich will register notifications in client app.\npre class=\"brush: plain; title: ; notranslate\"> let notificationT ypes: UIUserNotificationType = UIUserNotificationType.Badge | UIUserNotificationType.Alert | UIUserNoti ficationType.Sound\n let notificationSettings: UIUserNotificationSettings = UIUserNotificationSettings (forTypes: notificationTypes, categories: nil)\n \n application.registerUserNotificationSettings(not ificationSettings)\n application.registerForRemoteNotifications()\n<\li>\nAdd the following code inside didRegisterForRemoteNotificationsWithDeviceToken function in AppDelegate of client applic ation which will register pushclient and subscribe to tag in client app.\npre class=\"brush: plain; title: ; not ranslate\">IMFPushClient.sharedInstance().registerDeviceToken(deviceToken, completionHandler: { (res ponse, error) -& amp;gt; Void in\n if error != nil {\n println("Error during device registration \\(error.description)")\n else {\n println("Res ponse during device registration json: \\((response.responseJson.description)\)")\\n tags = ["parkstore"]\n IMFPushClient.sharedInstance().subscribeToTags(t ags, completionHandler: { (response:IMFResponse!, err:NSError!) -> Void in\n if err! println("There was an error while subscribing to tag")\n $= nil \{ \n$ }else{\n println("Successfully subscribe to tag parkstore")\n }\n<\li>Add the following function inside Appdelegate which trigger })\n }\n s when push notification arrived in client app.\nfunc applica tion(application: UIApplication, didReceiveRemoteNotification userInfo: [NSObject : AnyObject]) {\n р rintln("Got remote Notification. Data:\\(userInfo.description)")\n nfo as NSDictionary\n let data = info.objectForKey("aps")?.objectForKey(& quot;alert") as! NSDictionary\n let userData = data.objectForKey("body&q uot:) as! String\n let alertView = UIAlertView(title: ":WishList!":, message: & quot;\\(userData)", delegate: nil, cancelButtonTitle: "OK")\\n \n\<\pre>\n<\li>\n\end{ali}n\end{ali ef=\"https:\/\developer.ibm.com\/mobilefirstplatform\/wp-content\/uploads\/sites\/32\/2015\/07\/IMG_0020. jpg\"><\a><\a><\a><i mg 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-im age-14919\" \/><\a><\a>\n<\li>\n<\vul>\n<h2>Migration to On-Prem<\/h2>\n <h3 id=\"configureclient\">Migration of Client Application<\\h3>\nMigration of Client Application includ es following two steps\nConfiguring Cocoapods\rli>Client App Migration\rli>\n<h3 id=\"co coapods\">Configuring Cocoapods<\/h3>\nIf CocoaPods has not been installed on a specific comput er:\n<ul

n: http:///guides.cocoapods.org/using/getting-started.htmlOpen "Terminal&# 8221; at the installation location and run the "pod init" command<\/li> p>The following steps assume that the client application is working with CocoPods. If not, follow this &am p;#8220;Using CocoaPods" documentation : http:///guides.cocoapods.org/using/ using-cocoapods.html\" target=\"_blank\">click here<\/a><\/p>\nIn both cases, the instructions below explain how to edit the "Podfile" file.\n\nOpen the "Pod file" file located in the root of your XCode project in a favourite text editor.<\/li>\nCommen t out or remove the existing content.Add the following lines:\npre class=\"brush: plain; title: ; n otranslate\">source 'https://github.rtp.raleigh.ibm.com/imflocalsdks/imf-client-sdk-specs.git'\npod 'IMFC ompatibility'\n<\li>Open &#8220;Terminal&#8221; at the location of &#8220; Podfile".<\/li>\nli>\verify that the XCode project is closed.<\/li>\nRun the &#8220;po d install" command.li>\n<\ol>\nOpen the [MyProject].xcworkspace file in XCode. This f ile is located side by side with [MyProject].xcodeproj.
-kp \/>\nAn usual CocoaPods-based project is mana ged as a workspace containing the application (the executable) and the library (all project dependencies brought by the CocoaPods manager).In Xcode's Build Settings, search for & Description of the CocoaPods manager (Notes that the CocoaPods manager). #8220;Other Linker Flags" and insert \${inherited} (if -ObjC is defined in this field, you can jus t delete it, since it is configured in the CocoaPod project).\n<h3>Client App Migration</h3>\n\n Search for bluemix dependency imports like\n#import & amp;lt;IMFCore\IMFCore.h>\n#import <IMFPush\h&gt;<\/pre>\nReplac e the above imports with \nre class=\"brush: plain; title: ; notranslate\">#import <IMFComp atibility/IMFCompatibility.h>\n<\li>Look for a call to the &#8220;initializeWithBa ckendRoute&:#8221; method and replace the route URL with your on-premise server URL. For exam ple:\nple:\npre class=\"brush: plain; title: ; notranslate\">IMFClient.sharedInstance().initializeWithBackendRout e("https://vparkstore.mybluemix.net", backendGUID: "5e3ad88d-dd48-46 9d-b46f-2c4ad66b5345"\nshould be replaced with your on-premise MFP server UR L<\/p>\nclass=\"brush: plain; title: ; notranslate\">IMFClient.sharedInstance().initializeWithBackendR oute("http:\/\localhost:10080\/ParkStoreMFP", backendGUID: "5e3ad88d -dd48-469d-b46f-2c4ad66b5345"\nNote, that backendGUID parameter is ignored a nd can be empty. Look for all instantiations of IMFResourceRequest class and update it<\/li> r all instantiations of IMFResourceRequest class and update the request URL with absolute or relative pat h to the resource. For example:\n\nre class=\"brush: plain; title: ; notranslate\">var request: IMFResource Request = IMFResourceRequest(path: "https:\/\parkstore.mybluemix.net\/LocalstoreAdapter\/ apps\/5e3ad88d-dd48-469d-b46f-2c4ad66b5345\/localstore\/getAllItems", method: " GET")\nshould be replaced with\nclass=\"brush: plain; title: ; notranslate\" >var request: IMFResourceRequest = IMFResourceRequest(path: "http:\/\localhost:10080\/Pa rkStoreMFP\adapters\LocalstoreAdapter\localstore\getAllItems", method: "GET& amp;quot;)\n<\li>\nAdd the following code inside didRegisterForRemoteNotificationsWithDevic eToken function in Appdelegate of Client application.\nre class=\"brush: plain; title: ; notranslate\"> WL Push.sharedInstance().tokenFromClient = deviceToken.description<\/pre>\n<\/li>\nAll on-premise app lications require the "worklight.plist" file to be present in the application resource s. In the <code>IBMMobileFirstPlatformFoundationNativeSDK<\/code> pod we supply a file named <stro ng>sample.worklight.plist.\n\nLocate the &#8220;sample.worklight.plist&#82 21; file in the â€~IBMMobileFirstPlatformFoundationNativeSDK' pod.<\/li>\nl>Copy this file to the pa rent (application) project and rename it to "worklight.plist".<\/li>\nEdit the &a mp;#8220;worklight.plist" file by setting the "application id" key to th e name of your application deployed to the on-premise MFPF server<\/li>\n<\/ul>\n<\/li>\n<\/ol>\n<\/ol> "migratemfp\">Migration of JAX-RS Application to JAVA Adapter<//h3>\n\n\nTo migrate JAX-RS ap plication to on-prem (MobileFirst Foundation) server we need to do the following steps for server:\n Create MobileFirst Project & Driver & Create Native API app for iOS

\/>\n >\n<\ /a><\p>\n<img src=\"https:\//developer.ibm.com\/mobilefirs tplatform\wp-content\uploads\sites\/32\/2015\/07\/Screen-Shot-2015-07-12-at-6.51.13-pm.png\" alt=\"Sc reen Shot 2015-07-12 at 6.51.13 pm\" width=\"598\" height=\"590\" class=\"alignnone size-full wp-image-

\sites\\32\\2015\\07\\Screen-Shot-2015-07-12-at-6.52.28-pm.png\"><\/a><\/li>Add two adapters for Custom Authentication and Localstore and migrate the JAX-RS code as shown in the following example. x code and paste it in the newly created Localstore Java adapter JAX-RS file.hAdd and remove the following changes in your adapter code.\nremove <code>\{\text{tenantId}}\<\code><\li>\n remove the <code>@PathParam -&gt; PathParam(\"tenantId\") String deviceId<\code> and <c ode>@PathParam(\"realmName\") String realmName<\/code><\/li>\nhAdd scope to the all http api res ource <code>@OAuthSecurity (scope=\"customAuthRealm 3\")<\/code><\/li>\n<\/ul>\nThe code look s like the following\nclass=\"brush: plain; title: ; notranslate\">\n\t@GET\n\t@OAuthSecurity (sc ope="customAuthRealm 3")\n\t@Path("\/getAllItems")\n\tpubli c String getAllItems() throws MissingConfigurationOptionException{\n\t\tinit();\n\t\tJsonArray jsonArray = n ew JsonArray();\n\t\tfor(Object key: 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@OAuthSecurit y (scope="customAuthRealm_3")\n\t@Path("\/addItem")\n\tpu blic void addItem(String itemJson) \n\t\tthrows MissingConfigurationOptionException, URISyntaxExcepti on, IOException{\n\t\try{\n\t\t\tinit();\n\t\t\tint newKey = props.keySet().size()+1;\n\t\t\tprops.put(String.valu eOf(newKey), itemJson);\n\t\t\URL url = this.getClass().getClassLoader().getResource("data.p roperties"); \n\t\tFile file = new File(url.toURI().getPath());\n\t\t\tFileOutputStream foStream = n ew FileOutputStream(file);\n\t\tprops.store(foStream, & p;quot;saving new item");\n\t\t\foStr eam.close();\n\n\t\t}catch(IOException ioe){\n\t\t\tioe.printStackTrace();\n\t\t}\n\n\t\p\n\n\t\@POST\n\t@OAut hSecurity (scope="customAuthRealm_3")\n\t@Path("\/addAllItems& quot;\\n\tpublic String addAllItems(String itemsJson) \n\t\tthrows MissingConfigurationOptionException, URISyntaxException, IOException{\n\t\ttry{\n\t\tinit();\n\t\t\tlearAllData();\n\t\t\JsonArray jsonArr = parse r.parse(itemsJson).getAsJsonArray();\n\t\tfor(int i=0;i<jsonArr.size();i++){\n\t\t\tprops.put(St ring.valueOf(i+1), isonArr.get(i).toString());\n\t\t\t\t\URL url = this.getClass().getClassLoader().getRes ource("data.properties"); \n\t\tFile file = new File(url.toURI().getPath());\n\ t\t\tFileOutputStream foStream = new FileOutputStream(file);\n\t\t\tprops.store(foStream, & mp; amp; quot; saving new item");\n\t\t\tfoStream.close();\n\t\t\treturn "true&quo t;;\n\t\t}catch(IOException ioe){\n\t\t\tioe.printStackTrace();\n\t\t}\n\t\treturn "false&am p;quot;;\n\t\n\n\t@DELETE\n\t@OAuthSecurity(enabled=false)\n\t@Path("\clearAll" ;)\n\tpublic String clearAllData() \n\t\throws MissingConfigurationOptionException, URISyntaxException, IOException{\n\t\t\init();\n\t\t\props.clear();\n\t\t\System.out.println("Size: "+props.s ize());\n\t\t\URL url = this.getClass().getClassLoader().getResource("data.properties&quo t;); \n\t\t\File file = new File(url.toURI().getPath());\n\t\tFileOutputStream foStream = new FileOutputStre am(file);\n\t\t\tprops.store(foStream, "clearing all data");\n\t\t\foStream.close();\n\t\t\t return & amp;quot;cleared & amp;quot;;\n\t}\n<\/pre>\n<h3 id=\"configoauth\">Configuring Custom-OAuth<\ /h3>\nAdd realm with same name you had on BlueMix and login module to the authenticationC onfig.xml.\n<realm name=&quot;customAuthRea Im_3" loginModule="customAuthLoginModule_3">\n<class Name>com.worklight.core.auth.ext.CustomIdentityAuthenticator<\/className&gt;\t\n &|t;parameter name="providerUrl" value="http:\/\localhost:10080\/P arkStoreMFP\adapters\Customauth"\\>\n\amp;lt;\\realm>\n\n<\loginMo dule name="customAuthLoginModule_3" expirationInSeconds="3600&a mp;quot;>\n<className&gt;com.worklight.core.auth.ext.CustomIdentityLoginModule &|t;\/className>\n&|t;\/loginModule><\/pre>\n<\/li>Add Custom-oauth Rea Im in userIdentityRealms in Application Descriptor file of iOS Native API\nre class=\"brush: xml; title: ; n otranslate\"><userIdentityRealms&gt;customAuthRealm_3&lt;\/userIdentityRealms& ;gt;\n<\/li>\n<\/ul>\n<h3 id=\"configurepush\">Configuring Push Capability<\/h3>\n\nAdd a pns p12 certificate which is generated from Apple Developer Account under iOS Native API Folder\n <\/a \n_\/li\n_li\ Add Push configuration in Application Descriptor file of iOS Native API and include the pas

ZIIN VIIN IIN AUU TUSH GOHIIGUIALIOH IH APPIIGALIOH DESCRIPTOF IIIE OFFOOT NALIVE AFTALIO HIGIAGE LIE PAS sword of added apns certificate.\nre class=\"brush: xml; title: ; notranslate\"><pushSender pass word="password"\/>\n<tags&gt;\n &lt;tag&gt;\n &a mp;lt;\name>\n <\/tag&gt;\n&lt;\/tag&gt;\n&lt;\/tag&gt;\/ /pre>\n<\/li>\Create HTTP Push Adapter with following function code which will send the user push notification to the devices which is subscribed to tag "parkstore".\nre class=\" brush: xml; title: ; notranslate\">function sendTagNotification(notificationText) {\n var notificationOptions = {};\n notificationOptions.message = {};\n notificationOptions.target = {};\n\n notificationOptions.me ssage.alert = notificationText;\n notificationOptions.target.tagNames = ["parkstore" WL.Server.sendMessage("ParkStoreMFP", notificationOptions);\n\n retur n {\n result: & amp; quot; Notification sent to users subscribed to the tag parkstore. & amp; quot; \n \ \};\n \/c/pre>\n<\/li>\/nBy performing above steps one can easily run iOS app built for Bluemix on M obileFirst Platform and following are the links to samples.\n<h3 id=\"sample\">Sample and Source C ode<\h3>\nBluemix Server : Parkstore bluemix server<\/a><br \>\nBluemix Client : Parkstore bluemix<\/a><br \/>\nMFP Server : Parkstore mfp server<\a>
hMFP Client : Parkstore mfpThe post Try on Bluemix an d 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-m
fp∨",
         "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) includes an adapter called RSSAdapter and a hybrid application called RSSReader to test the adapter inside an application.