

Db2 for z/OS Mobile Application using Db2 REST service

Jane Man

IBM

Session code: E11

Wednesday, 04.10.2017 11:00-12:00

Platform: Db2 for z/OS



A new era of applications focused around mobile devices is becoming increasingly important in large enterprises. RESTful APIs play an important role in this area. Learn how you can expose Db2 functionality through Db2 REST service that are fully integrated in the Db2 distributed data facility (DDF) and how you can consume a Db2 REST service to build a Db2 for z/OS mobile application using IBM MobileFirst. Using an easy to adapt example, we show what needs to be done from front end to the back end database server to make this happen.

Objectives

- To introduce Db2 REST service
- To illustrate how to expose a Db2 functionality using Db2 REST service
- To show how to consume a Db2 REST service when building a mobile application

Please note

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Agenda

- Overview of Db2 for z/OS native REST service
- How to build a Db2 for z/OS mobile application

Overview of Db2 for z/OS Native REST service

```
POST https://<host>:<port>/services/simpleSelect1
{ "LOCATION": "</location>"}
```

540 million
RESTful
requests per
hour

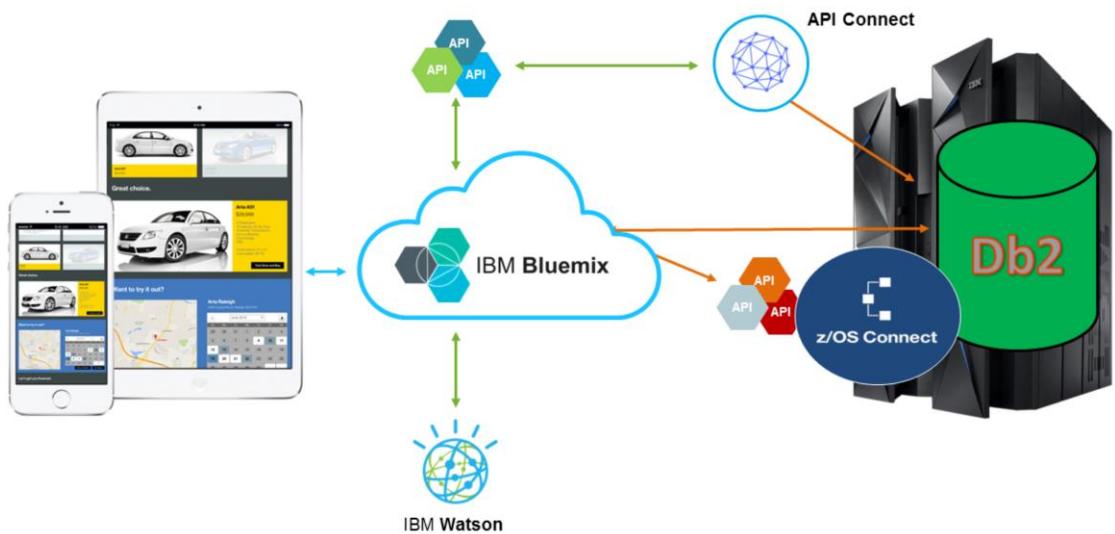
Db2 is now a RESTful service provider (V11 PI66828)

Enabling new business value for your enterprise data

Modernizing using the power of SQL

Unleashing Db2 data for the API Economy

Db2 is now a RESTful service provider! In other words, in addition to “traditional” way of accessing Db2 using jdbc, odbc, you can access Db2 by sending a REST call.



7

A few options in accessing the REST service provided by Db2 from mobile devices.

Db2 as a Service Provider – Solution Overview



Db2 User Created Service – A single SQL statement (INSERT, UPDATE, DELETE, SELECT, CALL) bound into a package.
 Db2 Management Services – Create, discover and invoke a Db2 user created service using a REST API.

DDF Interface – Processes HTTP requests and replies, performs REST/JSON message formatting invokes as a static SQL statement

Availability – Utilizes existing DDF capability including thread pooling, profiling and enclave classification.

Leverages existing package lifecycle – Services are in the catalog tables and saved in the directory.

Security, accounting, statistics, and auditing – Uses existing DDF infrastructure. Supporting classification, profiling, and thread pooling

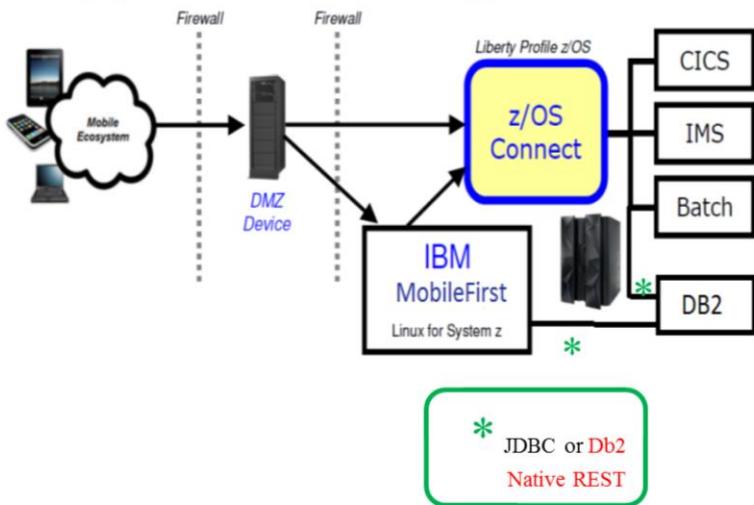
z/OS Connect EE Integration – Uses the z/OS Connect EE REST Client to allow exporting of Db2 svcs into z/OS Connect rest client

9

DDF allows user to create, drop, discover, and invoke a service.

How to build a Db2 for z/OS mobile application

Simplified End-to-End Architecture for Mobile and Cloud Application invoking z Services using APIs



Here is the end-to-end architecture for mobile application to access System z.

Db2 adapter for z/OS Connect v1 is using JDBC to access Db2.

IBM MobileFirst is previously known as IBM Worklight. It can access Db2 using SQL Adapter(JDBC) or REST (HTTP adapter)

IBM MobileFirst Platform

- Integrated development and continuous delivery of mobile applications

From the complexity of many...

- Multiple sets of tools & frameworks
- Four codebases to develop and maintain



To the simplicity of one

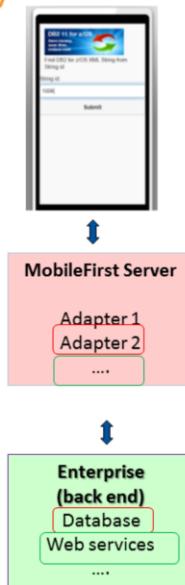
- One development environment
- One codebase to develop and maintain



One strength of IBM MobileFirst is: we no longer need multiple sets of tools & environments.

IBM MobileFirst 7 Studio

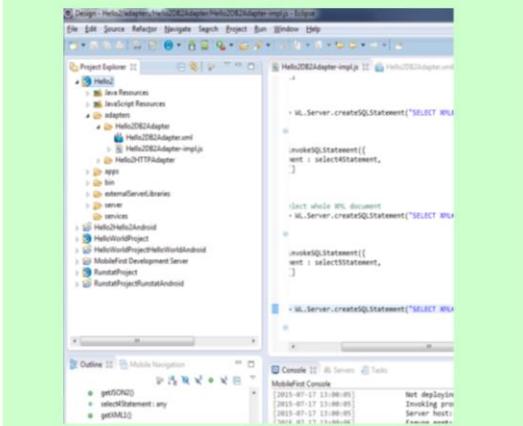
- Native applications for
 - iOS
 - Android
 - Windows Phone 8
 -
- Hybrid development
- Server-side development (adapters)
 - SQL Adapter
 - HTTP adapters
 - Java adapters
 -
- Mobile Console Browser
- ...much more



IBM MobileFirst allows us to do native mobile app, hybrid app, including both client side and server side development. It also provide a console browser to test.

IBM MobileFirst Development Environment

MobileFirst Platform Studio - Eclipse plugin



Command line interface (alternative to Studio)

1. If you have installed the CLI, in the terminal, create a project.

```
1 | $ mfp create HelloWorldProject
2 | A MobileFirst Project was successfully created.
```

2. Go to the context of your newly created project.

```
1 | $ cd HelloWorldProject/
```

3. Define a hybrid application named HelloWorld.

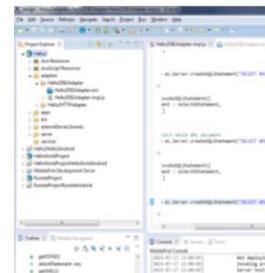
```
1 | $ mfp add hybrid HelloWorld
2 | A new Hybrid App was added at /User:/HelloWorldProject/helloWorld.
```

The command line interface is intended for more advanced users.

MobileFirst Platform Developer Studio – Installation

- JRE 7 (Note: Oracle JRE 7 is required for developing Android native app)
- Install Eclipse IDE for Java EE Developers installed: Juno SR2 (4.2.2), Kepler SR1 (4.3.1), Kepler SR2 (4.3.2), Luna SR1 (4.4.1), or Luna SR2 (4.4.2) Third Topic
- Start Eclipse, and then select Help > Eclipse Marketplace.
 - In the Find field, type “MobileFirst Platform” and click Go.
 - Optional : IBM Dojo Mobile Tools and IBM jQuery Mobile Tools
- (Optional) **Android** development : Android SDK, ADT Eclipse plug-in, creating Android Virtual Device
- (Optional, but highly recommended) Web browser with REST client installed (for creating, invoking Db2 REST services)

(<https://developer.ibm.com/mobilefirstplatform/install/#studio>)



16

Most of these are free to download (under Open source license)

Use Case : SQL statement

Find string from string id in SYSIBM.SYSXMLSTRINGS table

```
SELECT STRINGID, SUBSTR(STRING,1,60), IBMREQD  
FROM SYSIBM.SYSXMLSTRINGS;
```

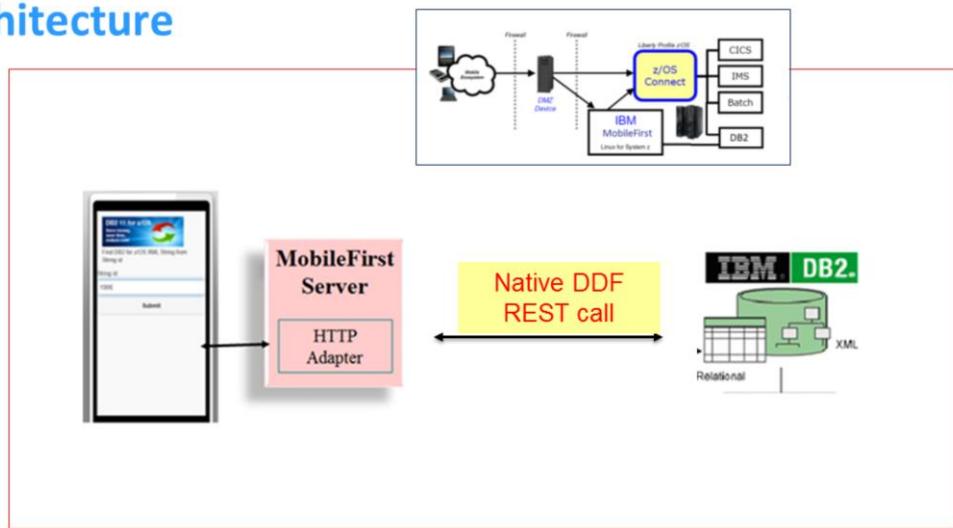
File Edit Format View Help	
STRINGID	STRING
1001	product
1002	description
1003	name
1004	detail
1005	http://www.w3.org/2000/xmlns/
1006	space
1007	http://posample.org
1008	pid
1009	details
1010	price
1011	weight

17

We start with a scenario to select from a catalog table.

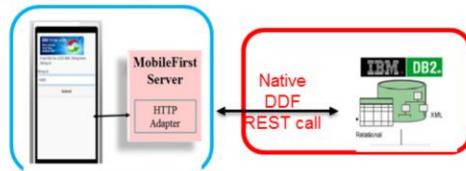
SYSIBM.SYSXMLSTRINGS is a catalog table which contain the mapping between actual string and string id that are used in XML storage. That is, when you insert an XML document into Db2 for z/OS, Db2 convert the element name, attribute name, prefix name, etc into a 4 bytes string id according to the mapping in this catalog table.

Architecture



This is the overall architecture for the mobile application we are going to build.

Steps Overview



Back end : create a Db2 REST service

1. Customize and Run DSNTIJRS to create Db2 RESTful service database and table (pseudo catalog table)
2. Authorize user to access services
3. Create a service using a REST client browser

Front end : consume a Db2 REST service

1. Create client project in MobileFirst Developer Studio
2. Create a HTTP Adapter
3. Build and Deploy

The steps can be divided to “back end” and “front end”. More details follow.

Backend details : create a Db2 REST service

1. Customize and Run DSNTIJRS to create Db2 RESTful service database and table
2. Authorize user to access services
Access to Db2 REST is protected by a SAF DSNR class <ssid>.REST profile
3. Create a Db2 REST service

```
//RESTDSNRJOB CLASS=A,MSGCLASS=H,REGION=0M,
//   USER=RACF000,PASSWORD=PASSWORD
//*****
//** RACF COMMANDS TO ADD DSNR CLASS PROFILES FOR REST
//*****
/STEP1 EXEC PGM=IKJEFT01
//SYSRACF DD DSN=SYS1.RACF,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RDEFINE DSNR (DB2A.REST) UACC(READ)
```

POST ▾

http://dtec731.vmec.svl.ibm.com:446/services/DB2ServiceManager

Headers

The headers specified below will be added to any sent by the browser or specified elsewhere.

Name	Value
Accept	application/json
Content-Type	application/json
Accept-Charset	UTF-8

Enter the data and its corresponding MIME type below.

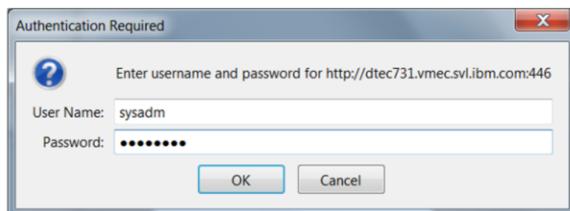
application/json

```
{
  "requestType": "createService",
  "sqlStmt": "SELECT SUBSTR(STRING,1,60) as STRING from
  SYSIBM.SYXMLSTRINGS WHERE STRINGID= ?",
  "serviceName": "selectSYSXMLStrings",
  "description": "Select string from XML string id"
}
```

20

We don't need a DBA to create a Db2 REST service. The easiest way to create a REST service is to use a web browser with REST client installed.

Backend details : create a Db2 REST service



Output of creating Db2
REST service

```
{  
  "StatusCode": 201,  
  "StatusDescription": "DB2 Rest Service selectSYSXMLStrings was created  
successfully.",  
  "URL":  
  "http://dtec731.vmec.svl.ibm.com:446/services/SYSIBMSERVICE/selectSYSXMLStrings"  
}
```

21

Message for a successful creation of service. You may want to write down the service url or you can use the discovery service in the next slide.

Backend details : Discover all Db2 REST service (optional)

GET <http://dtec731.vmec.svl.ibm.com:446/services>

```
{
    "DB2Services": [
        {
            "ServiceName": "DB2ServiceDiscover",
            "ServiceCollectionID": null,
            "ServiceDescription": "DB2 service to list all available services.",
            "ServiceProvider": "db2service-1.0",
            "ServiceURL": "http://dtec731.vmec.svl.ibm.com:446/services/DB2ServiceDiscover"
        },
        {
            "ServiceName": "DB2ServiceManager",
            "ServiceCollectionID": null,
            "ServiceDescription": "DB2 service to create, drop, or alter a user defined
service.",
            "ServiceProvider": "db2service-1.0",
            "ServiceURL": "http://dtec731.vmec.svl.ibm.com:446/services/DB2ServiceManager"
        },
        {
            "ServiceName": "selectSYSXMLStrings",
            "ServiceCollectionID": "SYSIBMSERVICE",
            "ServiceDescription": "Select string from XML string id",
            "ServiceProvider": "db2service-1.0",
            "ServiceURL": ""
        }
    ],
    "ServiceURL": "http://dtec731.vmec.svl.ibm.com:446/services/SYSIBMSERVICE/selectSYSXMLStrings"
}
}
```

The one we circled is the one we(user) created.

Can I find my service in Db2 Catalog?

SELECT * FROM "SYSIBM"."DSNSERVICE"

NAME	COLLID	CONTOKEN	ENABLED	CREATEDTS
ALTEREDTS	DESCRIPTION			
selectSYSXMLStrings	SYSIBMSERVICE	c8e3e3d7d9c5e2e3	Y	2017-05-26
10:18:00.20094 20	17-05-26 10:18:00.20094	Select string from XML string id		
...				

SELECT NAME, COLLID, CONTOKEN, STATEMENT FROM SYSIBM.SYSPACKSTMT where name='selectSYSXMLStrings'

NAME	COLLID	CONTOKEN	STATEMENT
selectSYSXMLStrings	SYSIBMSERVICE	c8e3e3d7d9c5e2e3	DECLARE C1 CURSOR FOR SELECT SUBSTR(STRING,1,60) as STRING from SYSIBM.SYSXMLSTRINGS WHERE STRINGID= :P1

23

If you are interested, you can find your service in the catalog.

Backend details : create a Db2 REST svr (cont'd)

4. Get details for the service

```
{
    "StatusCode": 201,
    "StatusDescription": "DB2 Rest Service selectSYSXMLStrings was created successfully.",
    "URL":
    "http://dtec731.vmec.svl.ibm.com:446/services/SYSIBMSERVICE/selectSYSXMLStrings"
}
```

Output of creating
Db2 REST service

GET <http://dtec731.vmec.svl.ibm.com:446/services/SYSIBMSERVICE/selectSYSXMLStrings>

```
{"selectSYSXMLStrings": {
    "serviceName": "selectSYSXMLStrings",
    "serviceCollectionID": "SYSIBMSERVICE",
    ...
    "serviceStatus": "started",
    "RequestSchema": {
        "properties": {
            "p1": {
                "type": ["null", "integer"],
                "multipleOf": 1,
                "minimum": -2147483648,
                "maximum": 2147483647,
                "description": "Nullable INTEGER"
            }
        },
        "required": ["P1"], ...
    }
}
```

INPUT

As a developer, you may want to know the input/output parameter name/datatype of a service. You can retrieve details of a REST service by sending a GET call.

Backend details : create a Db2 REST service (cont'd)

Details on `selectSYSXMLStrings` service

```
"ResponseSchema": {  
    "$schema": "http://json-schema.org/draft-04/schema#",  
    "type": "object",  
    "properties": {  
        "ResultSet Output": {  
            "type": "array",  
            "items": {  
                "type": "object",  
                "properties": {  
                    "STRING": {  
                        "type": "string",  
                        "description": "CHAR(60)"  
                    }  
                },  
                "required": ["STRING"],...  
            }  
        }  
    }  
}
```

OUTPUT

The output parameter name may change, it is better to check.

Backend details : create a Db2 REST service (cont'd)

5. Invoke a service

POST <http://dtec731.vmec.svl.ibm.com:446/services/SYSIBMSERVICE/selectSYSXMLStrings>

- Headers

The headers specified below will be added to any sent by the browser or specified elsewhere.

Name	Value
Accept	application/json
Content-Type	application/json
Accept-Charset	UTF-8

- Data

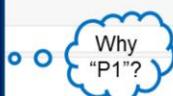
Select one of the options below to include data with the request.

Custom

Enter the data and its corresponding Content-Type

application/json
 {
 "P1": 1006
 }

application/json
 {
 "P1":1006
 }



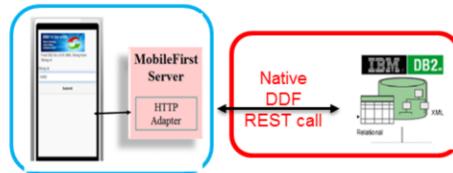
OUTPUT

```
{
  "ResultSet Output": [ {
    "STRING": "space" },
   "StatusCode": 200,
   "StatusDescription": "Execution Successful"
 }]
```

Why P1? – check previous slides ;-)

Demo

Steps Overview



Front end : consume a Db2 REST service

1. Create client project in MobileFirst Developer Studio
2. Create a HTTP Adapter
3. Build and Deploy

28

Front end step are in MobileFirst.

Mobile device is only one example to invoke the Db2 REST service. Your front end can be any application, as long as they can issue a REST call.

Application Logic Flow



1. Front end interface is implemented in `index.html`, which call a function(`loadFeeds`) in `main.js`

2. `loadFeeds` in `main.js` call HTTP Adatper

3. HTTP Adapter call Db2 using REST
4. Result is passed back to `main.js` and display in the front end.

29

IBM MobileFirst allows us to do native mobile app, hybrid app, including both client side and server side development. It also provide a console browser to test.

Client development using studio

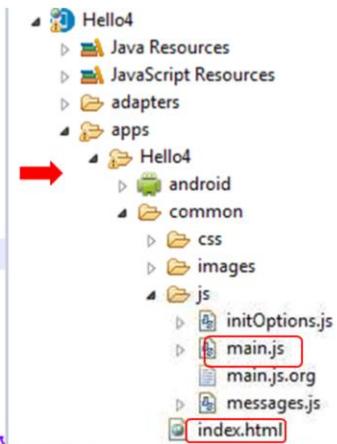
Create a new MobileFirst Project:

File>New>MobileFirst Project, select "Hybrid Application"

```
<!--application UI goes here-->

<br/>
    Find DB2 for z/OS XML String from String id
    using native DDF REST support
</div>
<label for="text">String id:</label>
<input type="text" name="text" id="stringid">
<input type="button" value="Submit" src="js/main.js"
    onclick="loadFeeds document.getElementById('stringid').value">
<div id="wrapper">
    <ul id="itemsList"></ul>
</div>
```

Index.html



30

The front end is in index.html. We just need a text field and a button.

Client development using studio

```
25
26 function loadFeeds(stringid){
27     WL.Logger.debug("Inside loadFeeds");
28     busyIndicator.show();
29     stringidInput = stringid;
30
31     // Use HTTP Adapter
32     WL.Logger.debug("Inside loadFeeds(): using HTTP Adapter for native DDF REST");
33     var invocationData = {
34         adapter : 'Hello4HTTPAdapter', ←
35         procedure : 'getXMLString',
36         parameters : [stringid]
37     };
38     WL.Client.invokeProcedure(invocationData,{
39         onSuccess : loadFeedsSuccess,
40         onFailure : loadFeedsFailure
41     });
42
43
44 }
```

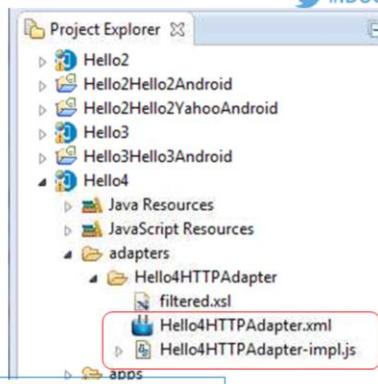
Main.js

31

When the submit button is clicked, loadFeeds function in main.js is invoked.

Creating a HTTP Adapter

1. Click **File > New > MobileFirst Adapter**.
2. Select **Hello4** as project name.
3. Select **HTTP Adapter**.
4. Enter **Hello4HTTPAdapter** as adapter name.
5. Click **Finish** button



```

<connectivity>
  <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
    <protocol>http</protocol>
    <domain>dtec731.vmec.svl.ibm.com</domain>
    <portL>446</portL>
    ....
    <!-- Below is added by Jane -->
    <procedure name="getXMLString"/>
  
```

Hello4HTTPAdapter.xml

32

HTTP Adapter allows us to send a REST call. After a HTTP Adapter (Hello4HTTPAdapter) is created, there will be files: Hello4HTTPAdapter.xml and Hello4HTTPAdapter-impl.js. The .xml file is for configuration purpose while the .js is for implementation.

Creating a HTTP Adapter – cont'd

```

90  function getXMLString(stringid) {
91    //var path = "http://dtec731.vmec.svl.ibm.com:446/services/
92    var path = "/services/SYSIBMSERVICE/selectSYSXMLStrings";
93    var stringidInt = parseInt(stringid);
94
95    var input = {
96      method : 'post',
97      returnedContentType : 'JSON',
98      path : path,
99      headers: {
100        Authorization: "Basic xxxxxxxxxxxxxxxxxxxxxx",
101        Accept: 'application/json'
102      },
103      body : {
104        contentType: 'application/json',
105        content: {"P1": stringidInt}
106      }
107    };
108
109    return WL.Server.invokeHttp(input);
110  }

```

Hello4HTTPAdapter-
impl.js

▼ Request headers (2.401 KB)

Host:	dtec731.vmec.svl.ibm.com:446
User-Agent:	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept:	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language:	en-US,en;q=0.5
Accept-Encoding:	gzip, deflate
Content-Type:	application/json
Content-Length:	18
Cookie:	UnicaNIODID=LBgxaZysX2T-Zi7hAo6...1496
Authorization:	Basic d...c...1496
Connection:	keep-alive

Why
“P1”?

Fill in details for the REST call.

Deploy and Test a HTTP Adapter

1. Right-click the **Hello4HTTPAdapter > Run As > Deploy MobileFirst Adapter.**

2. In **Call MobileFirst Procedure GUI**, enter the value of the parameter, then click **Run**

```
{  
    "ResultSet Output": [  
        {  
            "STRING": "space"  
        }  
    ],  
    "StatusCode": 200,  
    "StatusDescription":  
    "Execution Successful",  
    "isSuccessful": true,  
    "responseHeaders": {  
        ...  
    },  
    "responseTime": 824,  
    "statusCode": 200,  
    "statusReason": "OK",  
    "totalTime": 829  
}
```

What
format is
this?

JSON!!

Very easy to test using built-in tool.

How to use the adapter result?

```

97     function loadFeedsSuccess(result){
98         WL.Logger.debug("Feed retrieve success");
99         busyIndicator.hide();
100
101        if (result.responseJSON["ResultSet Output"] != null)
102            displayFeeds(result);
103        else
104            loadFeedsFailure();
105
106    }
107
108    function displayFeeds(result){
109        var UI = $('#itemlist');
110        WL.Logger.debug("Inside displayFeeds, result= " + JSON.stringify(result));
111        var ResultSetOutput = result.responseJSON["ResultSet Output"];
112        var description = ResultSetOutput[0].STRING;
113        description = description.replace(/(\r\n|\n|\r)/gm, '');
114
115        WL.SimpleDialog.show(description , "String id for " + stringidInput + " is found.",
116            [
117                {
118                    text : 'Reload',
119                    handler : WL.Client.reloadApp
120                }
121            ]
122        );
123    }

```



```

26     function loadFeeds(stringid){
27         WL.Logger.debug("Inside loadFeeds");
28         busyIndicator.show();
29         stringidInput = stringid;
30
31         // Use HTTP Adapter
32         WL.Logger.debug("Inside loadFeeds(): using
33         var invocationData = (
34             adapter : 'Hello4HTTPAdapter',
35             procedure : 'getXMLString',
36             parameters : [stringid]
37         );
38         WL.Client.invokeProcedure(invocationData,
39             onSuccess : loadFeedsSuccess,
40             onFailure : loadFeedsFailure
41         ));

```

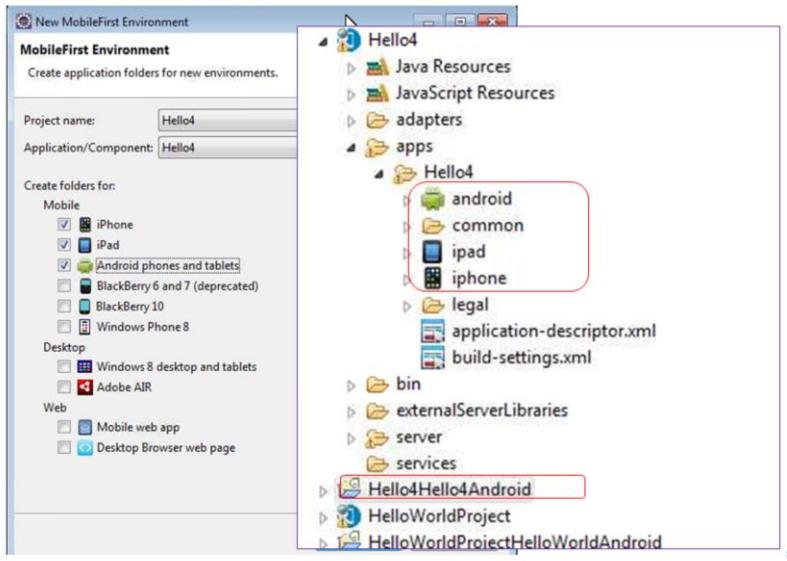
main.js

35

To use the REST result from HTTP Adapter, use `result.responseJSON` object.

Create and Build all Environments

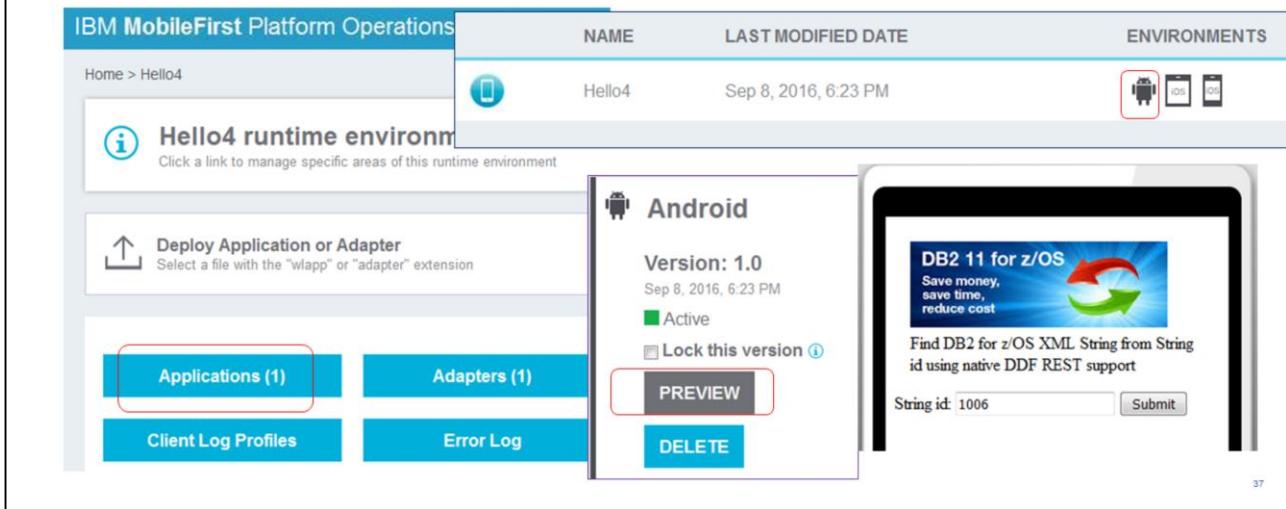
1. File > new > MobileFirst Environment
2. Select iPhone, iPad, Android phones and tablets
3. Right click <apps>Run As>Build All Environments



Build multiple environments just in a few clicks.

MobileFirst Console

- Right Click <project> > Open MobileFirst Console



The screenshot shows the IBM MobileFirst Platform Operations interface. On the left, a sidebar for the 'Hello4' runtime environment is visible, featuring links for 'Deploy Application or Adapter', 'Applications (1)', 'Adapters (1)', 'Client Log Profiles', and 'Error Log'. The main area displays a table with one entry:

NAME	LAST MODIFIED DATE	ENVIRONMENTS
Hello4	Sep 8, 2016, 6:23 PM	

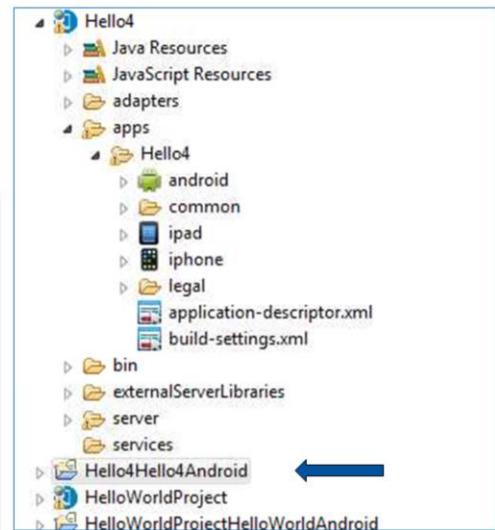
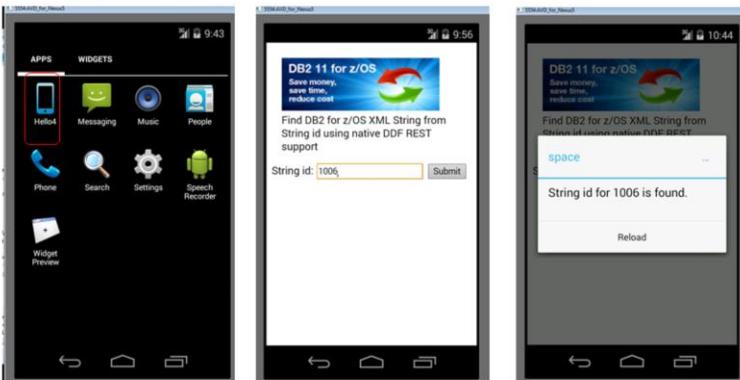
A detailed view of the 'Hello4' application is shown on the right, specifically for the 'Android' environment. It shows a version labeled 'Version: 1.0' from 'Sep 8, 2016, 6:23 PM' in 'Active' status. There are buttons for 'PREVIEW' and 'DELETE'. To the right of this is a preview of a mobile application screen titled 'DB2 11 for z/OS' with the tagline 'Save money, save time, reduce cost'. The screen content includes a red and green circular arrow graphic and the text 'Find DB2 for z/OS XML String from String id using native DDF REST support'. At the bottom is a text input field 'String id: 1006' and a 'Submit' button.

MobileFirst Console show all applications, adapters, etc. associated with the same project.

Run on Android Native Emulator

(require Android SDK, ADT Eclipse plug-in, creating
Android Virtual Device)

Right Click <project> <apps>Android>Run As> Android
Application



Run on native emulator is simple.

Bonus

**Node.js application to
invoke
Db2 for z/OS Native
REST service**

Let's add a bonus part : how to implement a node.js application to invoke Db2 for z/OS Native REST service.

Node.js and Db2 Native REST

Pre-req:

1. Install Node.js from <https://nodejs.org/en/download/>

NPM(tool to manage Node modules) will be installed as well.

By default, the install directory is:

C:\Program Files\nodejs\

2. To verify

```
C:\Users\IBM_ADMIN>node -v  
v6.9.2
```

```
C:\Users\IBM_ADMIN>npm -v  
3.10.9
```

3. Install DB2 Node.js driver

```
C:\NodeApp>npm install ibm_db
```

4. Install node-rest-client

```
C:\NodeApp>npm install node-rest-client
```

Comparing to Python, Spark, the pre-req for node.js is simpler.

Now, we can create a Node.js application to invoke the REST service we created

```
var Client = require('node-rest-client').Client;
var client = new Client();

//create a base64 encoding of userid:password , need to fill out with actual userid and password
var userpw = "userid:password";
var buffer = new Buffer(userpw);
var userPwBase64 = buffer.toString('base64');

var args = {
  data: { "P1": 1006 },
  headers: { "Content-Type": "application/json",
    "Authorization": "Basic " + userPwBase64,
    "Accept": "application/json"   }
};

client.post("http://dtec731.vmec.svl.ibm.com:446/services/SYSIBMSERVICE/selectSYSXMLStrings",
args, function (data, response) {
  console.log(data);
  var ResultSetOutput = data["ResultSet Output"];
  var description = ResultSetOutput[0].STRING;
  console.log(description);
});
```

Content of
nodeREST.js

The source is self explained.

We will use node-rest-client.post() function to send request to the Db2 REST service we created earlier. Before that, we need to prepare the arguments. The final part of the source just display the response of the REST service.

Very simple + straight forward.

Run the Node.js Application

```
C:\NodeApp>node nodeREST.js
```

Output:

```
{ 'ResultSet Output': [ { STRING: 'space  
' } ],  
  StatusCode: 200,  
  StatusDescription: 'Execution Successful'  
}  
  
space } From  
          console.log(description)
```

} From
 console.log(data)

Summary

- Overview of Db2 for z/OS native REST service
- How to build a Db2 for z/OS mobile application
- Invocation of Db2 REST service from Node.js application

Resources

- *DB2 for z/OS Native DDF REST services - 2 whitepaper by Jane Man*
https://www.ibm.com/developerworks/community/blogs/e429a8a2-b27f-48f3-aa73-ca13d5b69759/entry/DB2_for_z_OS_Native_DDF_REST_services?lang=en
- **DB2 REST services**
http://www.ibm.com/support/knowledgecenter/en/SSEPEK_11.0.0/restserv/src/tpp/db2z_restservices.html
- *Node.js Application and DB2 REST service by Jane Man*
https://www.ibm.com/developerworks/community/blogs/e429a8a2-b27f-48f3-aa73-ca13d5b69759/entry/Node_js_Application_and_DB2_REST_services?lang=en
- **IBM MobileFirst 7 Developer Edition**
<https://developer.ibm.com/mobilefirstplatform/install/#studio>





Jane Man

IBM

janeman@us.ibm.com

Db2 for z/OS Mobile Application using Db2 REST service

Session code: E11



*Please fill out your session
evaluation before leaving!*