


IDUG
Leading the DB2 User
Community since 1988

IDUG DB2 EMEA Tech Conference
Lisbon Portugal | October 2017

 **#IDUGDB2**


DB2 REST API and z/OS Connect – SQL/Stored Procedures Play a Role in Mobile and API Economics

Maryela Weihrauch
IBM Distinguished Engineer z Systems Analytics

Session code: F16

Thursday, 5th October, Time: 8:30 - 9:30 am

Db2 for z/OS



Please Note

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice 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.

2

©2017 IBM Corporation



Please note the disclaimer on this slide as well as

Acknowledgements and Disclaimers

<p>Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.</p>
<p>The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.</p>
<p>All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.</p>
<p>© Copyright IBM Corporation 2014. All rights reserved.</p> <p>— U.S. Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.</p> <p>IBM, the IBM logo, ibm.com, DB2, DB2 Connect and z/OS Connect are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or TM), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at</p> <p>³ *"Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml</p>

©2017 IBM Corporation



Additional 1 disclaimers and trademarks on this slide.

Agenda

- The bigger picture - emerging business model “API Economy” is leveraging APIs to generate new business
- z/OS Connect provides a consistent REST(JSON) interface into z/OS- based business functions
- DB2 Adapter for z/OS Connect for REST(JSON)-enabled SQL and Stored Procedures in DB2 10 and DB2 11
- DB2 11 native RESTful service support provides a new interface into DB2 for z/OS

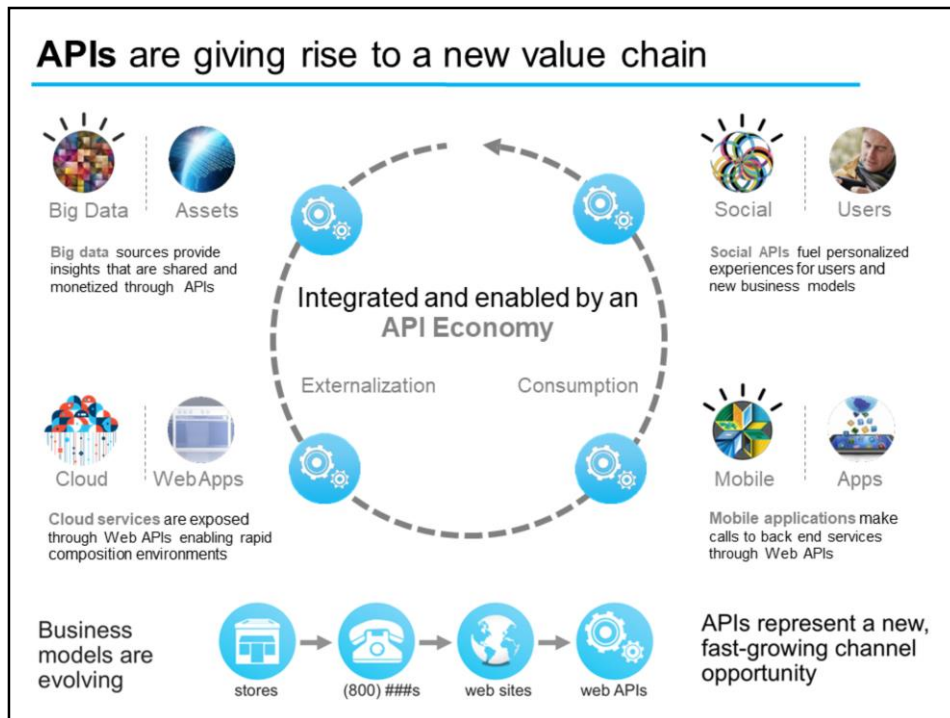
4

©2017 IBM Corporation



First, I'd like to illustrate the bigger picture of the emerging business model “API Economy” and show how APIs can be used to generate new business.

Before I go into the details of DB2 adapter for z/OS Connect and new native DB2 RESTful service support, I'll introduce z/OS Connect as a consistent REST(JSON) API into z/OS –based business functions, ,



APIs, or “application programming interfaces” are the way that software developers can invoke functions outside the code they are developing and introduce new functionality that they didn’t write but wish to use. This is a key part of hybrid cloud: the ability to build “hybrid applications” out of services. And these services are exposed via APIs. All of the new, emerging capabilities (think: CAMSS) either use APIs or can be exposed via APIs. Big Data and analytics results can be shared – and monetized – through APIs. Social tools, such as Twitter, Facebook, etc., are exposed and leveraged in hybrid apps via APIs. Mobile applications are key USERS of APIs, where those Systems of Engagement applications consist of many APIs linked together via application code. And APIs are generally hosted either on public or private cloud platforms

The business model that is emerging is the “API Economy”, where businesses are now exposing some of their key application assets as services and selling them to other users/application developers. OR, they are exposing their APIs to bring additional business to their company. Facebook, Amazon, and most of the “born on the web” companies have done this. Twitter gets more activity through their APIs than through their web site. Other, more traditional companies are now leveraging APIs to generate new business. Banks like PNC and Citi are looking at APIs to monetize the code they use to run their business.



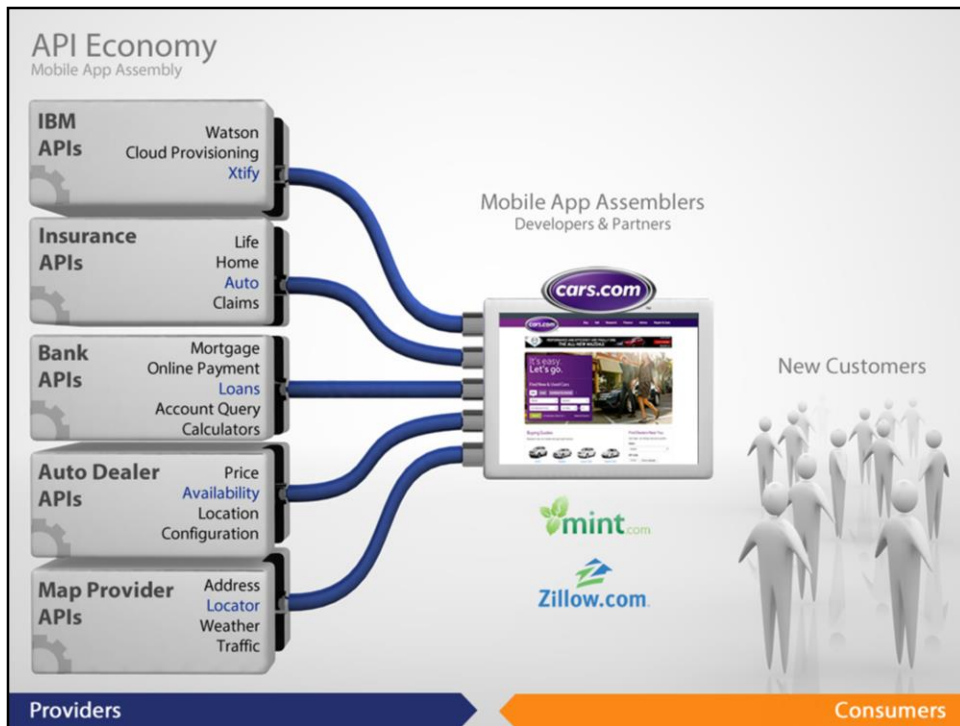
If we see APIs as the building blocks for application.

Moving from Germany, my first car buying experience caused me stowage

You see an example how a provider of external APIs can expanding into new customer base and niches that that company normally wouldn't be able to reach on their own.

The company can focusing on core value

- Easier possibilities in establishing new partnerships and capitalizing on them
- Benefitting from open innovations of crowdsourcing and expert sourcing which are implicitly unlocked through the opening of APIs
- Realize profit from new business models
- Decreasing development costs and time through software produced by third parties
- Keeping up with application demand (new apps, new features)



Lets look at the API consumer. Companies can very quickly provide new applications by using existing assets and services. Typically, the new application provides more value to the user by creatively combining functions that where previously either only available individually or difficult to come by without expert knowledge. They drive innovation by capitalizing on the merging of capabilities provided by different APIs.

Attracting clients to your products and services by joining other (established) API ecosystems

Potentially having freedom of choice in terms of available APIs

Focus on core competence and value add rather than re-inventing common capabilities

Innovate around business models and reach new customers from the provider's ecosystem

nsumer side

Introduction to REST

- **REST** - "Representational State Transfer" ... which uses HTTP/HTTPS and HTTP/HTTPS verbs to allow a client to interact with a server over the TCP/IP network.
- **JSON** - "JavaScript Object Notation" ... a name/value pair representation of data that is relatively lightweight and generally simpler to handle and parse than XML.
- An architectural style for **accessing** and **updating** data
- Simple and intuitive for the end consumer (the developer)



...Lets have a short introduction to REST.

REST is the abbreviation for Representational State Transfer. It is a distributed architecture where the client communicates with the server via HTTP or HTTPS over the TCPIP network.

The payload of the request is expressed in JSON format – JSON means JavaScript Object notation. It is a hierarchical name/value pair representation of the data.

REST(JSON) communication is more lightweight than SOAP(XML), the communication in the Service Oriented Architecture a couple of years ago.

REST is an architectural style for not just reading data but also updating data. Many may recognize it also as application architecture using API and Microservices. It becomes increasingly common as it is simple and intuitive for the application developer with client code mostly generated by tooling. The developer does not need to care about the implementation of the service function, if he knows the programming language the service is implemented in or SQL, database schema etc.

Lets look at an example. Above the line you see the HTTP verbs GET/POST/PUT/DELETE that match the CRUD operations. The service request is uniquely identified by the URI and can contain a list of input parameters.

The service call is always stateless and consist of a pair of request and response. The response contains the http return code, for example 200 means service call was successful and response body with returned data in the request body.

z/OS Connect Enterprise Edition

- **Provides single service channel for z/OS subsystem applications**
 - Industry Standard Interface (OpenAPI Initiative compliant)
 - Reduced interface layering, reduced complexity and costs
 - Scalable, highly available API
 - Secure, EAL5+
- **Delivers RESTful APIs as a discoverable, first-class resource with OpenAPI Spec (Swagger 2.0) descriptions**
 - Ready for consumption by today's enterprise application developers and integration with API management solutions
 - [Swagger specification](#)
- **Comprehensive tooling that enables API developers to create RESTful APIs from z/OS-based assets**
 - [API Editor download](#)
- **Supports standard JSON message format** and conversion to z/OS subsystem backend format requirements



©2017 IBM Corporation

z/OS Connect Enterprise Edition is a z/OS product that provides important functionality to enable services using z/OS applications.

First, it provides a single entry point into z/OS subsystem applications, which could run in WebSphere, CICS, IMS or DB2. Therefore reducing complexity and cost while providing a secure, scalable and highly available API runtime environment. Remember the business motivation in the introduction. Service providers generate additional business value by making their core assets available via API for integration into new applications and channels. A large amount of well optimized business functionality runs on z/OS. It is very simple and cost efficient to API-enable that functionality and make it available for new business use cases.

Second, z/OS Connect generates Swagger description for defined API, making it available to a large set of tooling for discovery and consumption. I added the link to the Swagger specification if you are not familiar with it.

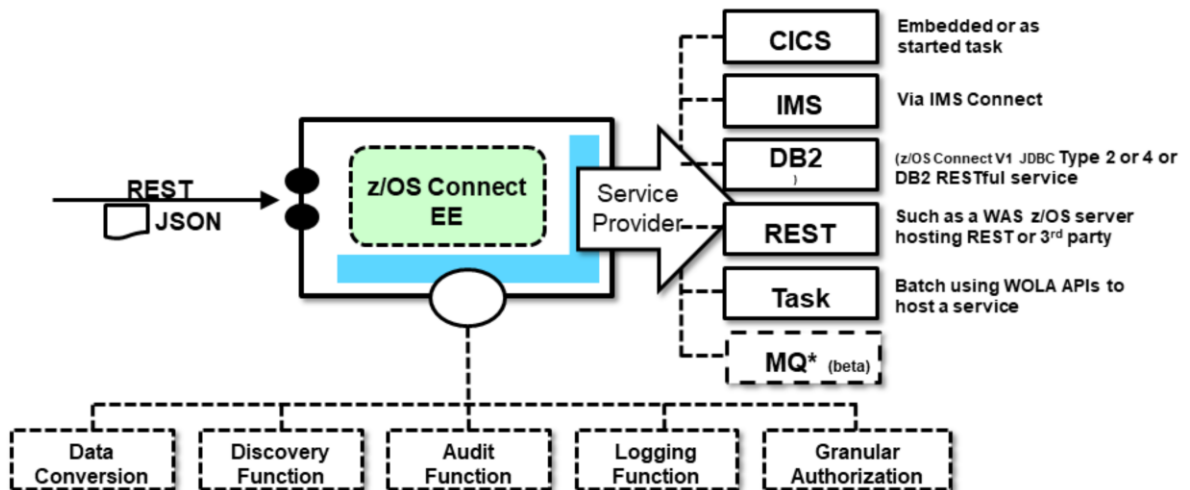
Third, z/OS Connect comes with comprehensive set of tooling for API developers to create APIs from z/OS-based assets.

And last but not least, it provides or integrates with a number of operational functionality including the conversion logic from REST(JSON) into subsystem

backend format.

z/OS Connect EE Structure

- Provides common infrastructure for discovery, authorization, auditing, logging
- Data conversion to service provider subsystem



* <http://www.slideshare.net/MattLeming/hhm3481-ibm-mq-for-zos-enhancing-application-and-messaging-connectivity-in-a-hybrid-world>

Lets have a closer look at the z/OS Connect structure itself.

As you see on the left side of the picture, it will communicate with the client via REST(JSON). All the supported subsystems provided functionality to convert REST(JSON) request to internal request format. For example for CICS into COBOL copy books. We will explain in the upcoming slides how the conversation works for SQL or DB2 Stored procedures.

You can see at the bottom that common operational functionality is provided for service discovery, auditing, logging, authorization.

Why use Swagger?

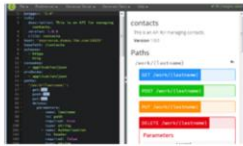


More than just an API description framework...

There are number of Open Source tools available to aid consumption:

Write Swagger

Swagger Editor allows API developers to design their Swagger documents.



Read Swagger

Swagger UI allows API consumers to easily browse and try APIs based on Swagger Doc.



Consume Swagger

Swagger Codegen creates stub code to consume APIs from various languages.



©2017 IBM Corporation

A bit more detail on Swagger. Swagger and Open API are interchangeable (Open API is the new name for Swagger)

Swagger has won against other API description frameworks because of the *open source* tools around it that aid development and consumption.

In its own words, the OPEN API INITIATIVE specification aims to “define a standard, language-agnostic interface to REST APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection. When properly defined, a consumer can understand and interact with the remote service with a minimal amount of implementation logic.”

[<https://www.openapis.org/specification/repo>]

The entire specification is freely available on github:

<https://github.com/OAI/OpenAPI-Specification>

DB2 Adapter for z/OS Connect 1.1 Overview

- DB2 Adapter for z/OS Connect is an integrated solution that enables developers to make (existing) DB2 assets – SQL and Stored Procedures – available in today's growing mobile and cloud application ecosystem
- Consists of 3 components
 - WLP with z/OS Connect 1.1 license
 - DB2 Adapter for z/OS Connect
 - IBM Data Studio, V4.1.2
<http://www.ibm.com/developerworks/downloads/im/data/>
- DB2 Adapter for z/OS Connect ships via DB2 Accessories Suite for z/OS, V3.3 (5697-Q04)
 - Functionality of DB2 10 and DB2 11
 - Some WLP skill preferred for install and configuration
- It is a good start to get familiar with functionality when still on DB2 10
- If already on DB2 11 NFM, recommend to start using native RESTful service

12 ©2017 IBM Corporation



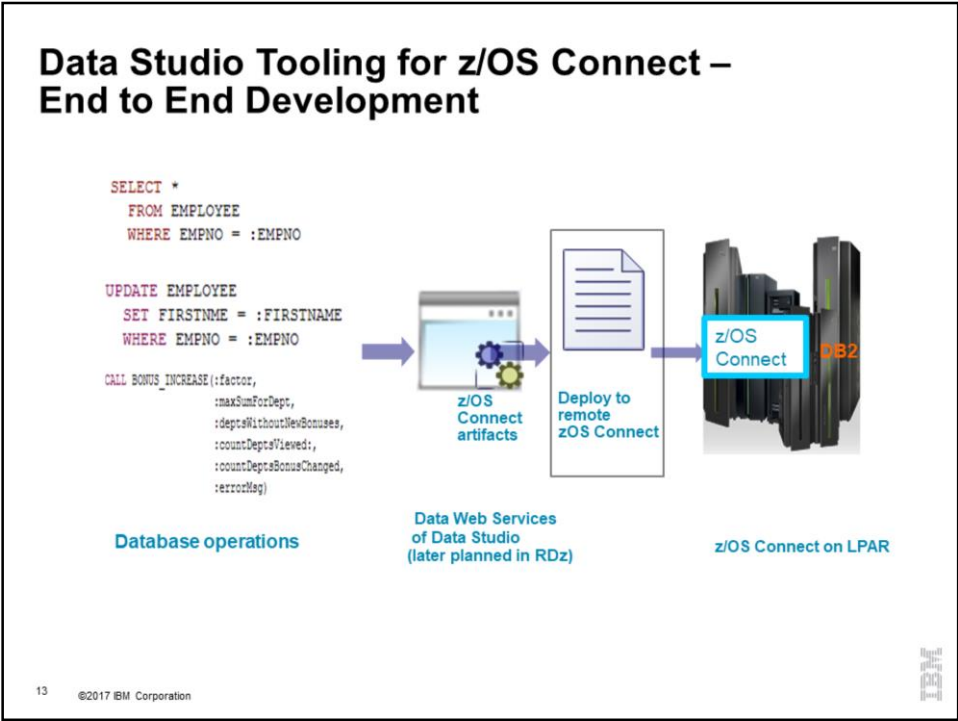
After setting the context, let's look how DB2 REST(JSON)-enables DB2 assets - SQL and Stored Procedures.

The first implementation was done in the DB2 Adapter for z/OS Connect as an integrated solution.

It is delivered as a prebuild package consisting of z/OS Connect V1.1 and DB2 Adapter for z/OS Connect in the DB2 Accessories Suite for z/OS V3.3 that can be ordered as a free feature of DB2 for z/OS.

If you are still running DB2 V10 and have no plan to migrate to DB2 V11 soon, it is a good start to get familiar with the technology. If you are running DB2 11 NFM or plan to get there soon, we recommend to start using the native RESTful service support.

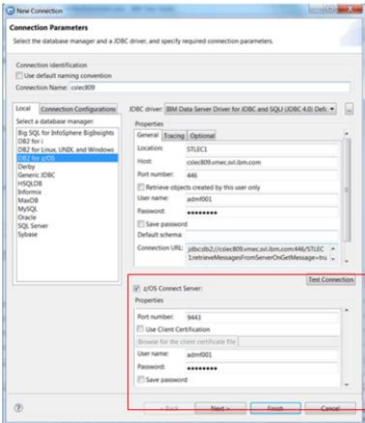
Version 4.1.2 of DataStudio provides functionality to REST-enable SQL and Stored Procedures and Deploy them to z/OS Connect where the DB2 Adapter is running or directly into DB2 when connected to DB2 11 NFM system.



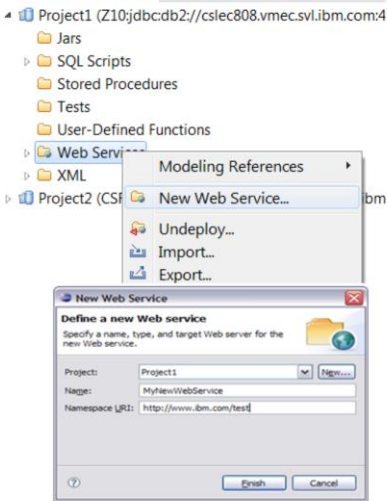
Lets have a closer look how it works. On a high level, DataStudio provides simple drag and drop fu

User Interface

1. Define z/OS Connect together with Database connection



2. Define a new service by right click on Web Services folder



14

©2017 IBM Corporation

There are 4 steps involved.

First, after you installed DataStudio, besides defining your DB2 for z/OS target system as a datasource, you also provide connectivity information for z/OS Connect where the DB2 Adapter is running. The userid provided needs to have the authority to call the DB2 service definition services and becomes the service owner.

I will explain that in more detail later.

As second step, you create a new WebService in the data project view under WebServices folder. You provide a name and short description.

User Interface...

3. Drag 'n drop

4. Right click to Deploy in z/OS Connect

The screenshot shows the IBM Data Explorer interface. On the left, the 'Data Project Explorer' pane displays a tree view of project assets including 'SQL Scripts', 'Stored Procedures', and 'User-Defined Functions'. A blue arrow points from a 'Stored Procedure' item to a 'Web Service' item in the 'Web Services' pane on the right. A text box with the text 'Drag and Drop SQL statements and stored procedures' is positioned over the arrow. Below this, a context menu is open for a selected web service, with the option 'Deploy on z/OS Connect Server' highlighted by a red rectangle. The menu also includes options like 'Build and Deploy...', 'Undeploy', and 'Export...'. The main editor area shows a SQL query: 'SELECT * FROM VENDOR.EMPLOYEE WHERE EMPLOYEE.EMPNO = :empno'.

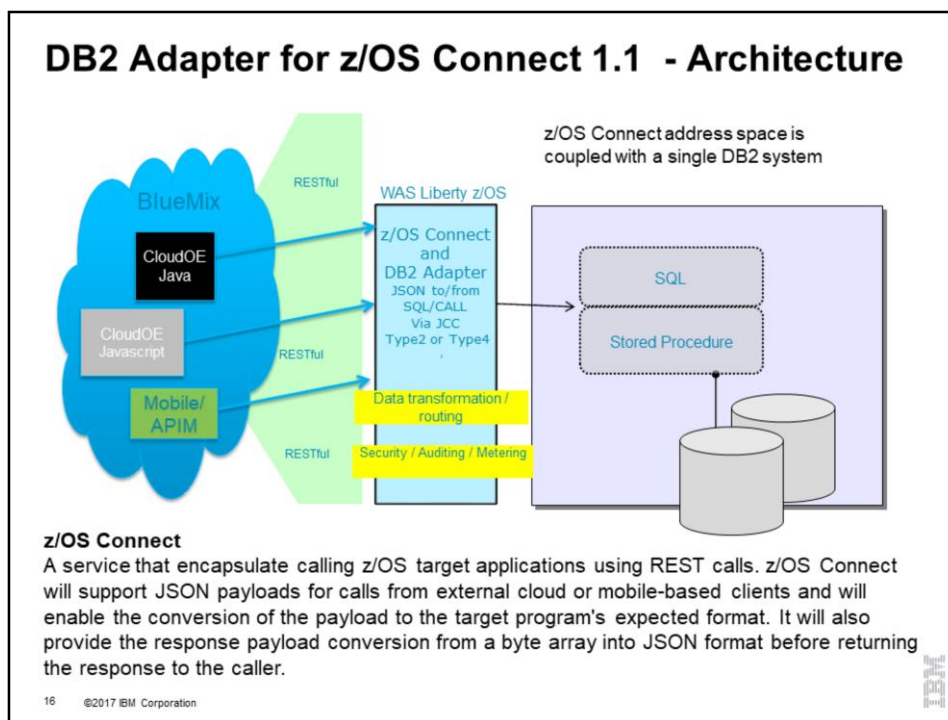
Service definitions are deployed in z/OS Connect – ready to use

15

©2017 IBM Corporation

Third, you drag and drop the stored procedure from the data exploration view or a predefined SQL from the project view onto the created webservices.

As last step, by right-clicking the service, select deploy to z/OS Connect server to deploy the service definition into the target z/OS Connect. Remember, that is the z/OS Connect where the DB2 Adapter runs.



This slide shows the end-to-end architecture with the DB2 Adapter providing the conversation from REST(JSON) service request to SQL and generating the service responds from SQL output.

z/OS Connect provides authentication, auditing and security functionality.

The DB2 Adapter uses a JDBC trusted context to connect to DB2.

Installation and Configuration

- The DB2 Adapter for z/OS Connect functionality does not run “in” DB2.
- Runs under the IBM Liberty Profile for z/OS application server runtime environment.
- Although we will include DB2 Adapter for z/OS Connect specific Liberty Profile for z/OS server “templates” to aid in server creation and customization, the installation will be easier for someone with some prior “WebSphere” installation and configuration skills.
- Much of the customization is performed by modifying the Liberty Profile for z/OS “server.xml” and associated files.
- Installation and configuration will also require coordination with SAF security administrator to create z/OS Connect security groups, etc.

©2017 IBM Corporation



You may also have realized, the DB2 Adapter runs in the z/OS Connect address space and is not native to DB2.

Also we include installation templates and documentation specific to installing and configuring the DB2 Adapter, it will be easier for someone with some prior WebSphere configuration experience.

DB2 systems programmers are just not used to define configuration in xml files that sit somewhere in a file system.

That was actually one reason why we moved the support into DB2.

DB2 Native RESTful Service Support

DB2 11 New Function Mode

PI66828 / PTF UI41625 (Closed Nov. 2, 2016)
PI70477 / PTF UI42683 (Closed Dec. 2, 2016)
PI74409 / PTF UI44779 (Closed Mar. 8, 2017)
PI80087 / PTF UI48082 (closed Apr 17, 2017)

DB2 12

PI70652 / PTF UI43239 (Closed Dec. 16, 2016)
PI74515 / PTF UI44784
PI80088 / PTF UI48178

©2017 IBM Corporation



The native RESTful service support is delivered in DB2 11 NFM and after with the listed PTFs. You can see that functionality is enhanced quickly and still existing limitations will be removed in future PTFs.

Motivation for Integrated Solution in DB2

DB2 native RESTful service functionality **maintaining z/OS Connect externals** as applicable and

- Is ready for use out of the box
- Reuses DDF functionality that provides a highly available, highly scalable and highly serviceable network interface
- DDF provides security, auditing, and data transformation as being provided using z/OS Connect
 - PERMIT DSN.REST CLASS(DSNR) ID(DB2USER) ACCESS(READ)
- Support of clientInfo fields
- Allows a database administrator to manage services
 - New catalog table SYSIBM.DSNSERVICE
 - Static execution pattern – auth checking, object dependency tracking

©2017 IBM Corporation



There are a number of reasons for moving the RESTful service functionality into DB2 itself.

Before I describe them, I need to point out that the externals of the defined services using the DB2 Adapter for z/OS are not changed and client application will continue to work even the service definition is migrated to the native support.

The first big benefit, every DB2 sysprog will appreciate, the functionality is ready to use after the PTFs are applied and HOLD actions are done – something they are very familiar with.

The implementation builds on the DB2 DDF functionality that was optimized over the last couple of DB2 versions and provides a highly scalable, highly available network interface.

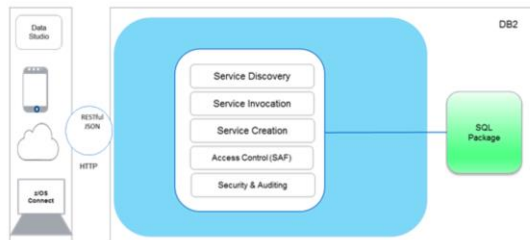
DDF provides security, auditing and very efficient data transformation. A new RACF permission REST is introduced to allow a user or group to access DB2 via HTTP/HTTPS.

The services are not managed in XML files but in a new catalog table

SYSIBM.DSNSERVICES in DB2, familiar to a DBA. DB2 uses a static execution pattern for authorization checking and can now do object dependency checking. That means if an SQL as an implementation of a service is not operational for any reason, the service is marked as not operational. That can be detected and corrected without calling a service.

DB2 Native RESTful Service Support ...

- Create|DROP|START|STOP Service using either Data Studio support, or direct DB2 RESTful service call
 POST https://<host>:<port>/services/DB2ServiceManager
 {"requestType": "createService«...."}
- Discover all services and discover details for a specific service using direct DB2 RESTful service call
 POST https://<host>:<port>/services/DB2ServiceDiscover
 { ["collectionID": "<collectionID>"]}
- DB2 RESTful service invocation.
 - DB2 Adapter for z/OS Connect V1 invocation API syntax
 - New DB2 native REST invoke API.



©2017 IBM Corporation

21



I mentioned before that DB2 uses system provided services to create and discover services.

DB2ServiceManager service is used to create/drop/start or stop a service. A newly created service is in stopped state and can only be called after it is started.

DB2 ServiceDiscover service is used to discover defined services and service description such as request schema and respond schema.

Besides those 2 services, DB2 supports service invocation of user-defined services.

Display of DB2 Native REST service locations and threads

You can issue the -DISPLAY THREAD command to identify and display threads that actively execute DB2® REST services. The correlation ID for a thread that processes a service is set to "DB2_REST" while the application name is the name of the service that is invoked.

The following is a sample output of the -DISPLAY THREAD command which identifies "DB2A" as the active thread and "simpleSelect1" as the application.

```
00- 17.39.04      -DB2ADIS THREAD(*) DET
- 17.39.04 STC00179 DSNV401I -DB2A DISPLAY THREAD REPORT FOLLOWS -
- 17.39.04 STC00179 DSNV402I -DB2A ACTIVE THREADS -
- NAME      ST A  REQ ID      AUTHID  PLAN      ASID TOKEN
- DB2A      RA *  0 028.DBAA  02 SYSOPR      0071  7
- V437-WORKSTATION=Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/2
- 0100101 Firefox/45.0
- USERID=user001
- APPLICATION NAME=simpleSelect1
- V441-ACCOUNTING=POST SYSIBMSERVICE.simpleSelect1
- V442-CRTKN=::FFFF:<host>.<port>.D149FF032587
- V445-G034E317.FC7D.D149FF032587=7 ACCESSING DATA FOR
- ( 1)::FFFF:<host>
- V447--INDEX SESSID      A ST TIME
- V448--( 1) <port>      R2 1624617373942
- DISCONN DA *  0 NONE      NONE      DISTSERV 0071  9
- V471-USIBMSY.SYEC1DB2.D149FF2F06EA=9
- DISPLAY ACTIVE REPORT COMPLETE
- 17.39.04 STC00179 DSN9022I -DB2A DSNVDT '-DIS THREAD' NORMAL COMPLETION
```

©2017 IBM Corporation



DB2 commands can be used to manage services similar to distributed workload. This example shows a DISPLAY THREAD(*) DETAILS output. You see connectivity information such as authid used as well as client info fields, which are set by default, for example clientapplication name is set to service name.

DB2 Native RESTful Service Support

- Uses cloud and mobile application programming HTTP and HTTPS protocols
 - Needs to be HTTPS for connectivity from DataStudio
- zIIP eligibility of distributed workload
- All DB2 Native REST APIs uses static SQL packages
 - Dynamic SQL with SQL as parameter to SQL PL SP
- DB2 Native RESTful service integrates with z/OS Connect EE
 - Coming: Tooling to simplify service definition in z/OS Connect
- **References**

[The easiest way on the planet to create RESTful Web Services!](#)

©2017 IBM Corporation



A couple more characteristics of native RESTful service support.

DB2 supports both HTTP and HTTPS. HTTP requests will connect to SQL port, you need to define a secure port for HTTPS requests. DataStudio will connect to DB2 via HTTPS.

I get this question frequently, DB2 service requests are zIIP eligible as any distributed workload.

All defined RESTful services uses static SQL packages and follow the static SQL authorization model. That means, the owner of the service needs appropriate authority to execute the SQL or stored procedure. The service caller needs execute authority on the service package.

By default, a service caller should not know that the service implementation is through SQL and would never see it. In specific cases, if you want to pass SQL as service input and want to execute as dynamic SQL, you would map it as input to a SQL PL stored procedure that prepares the input SQL and executes it as dynamic SQL.

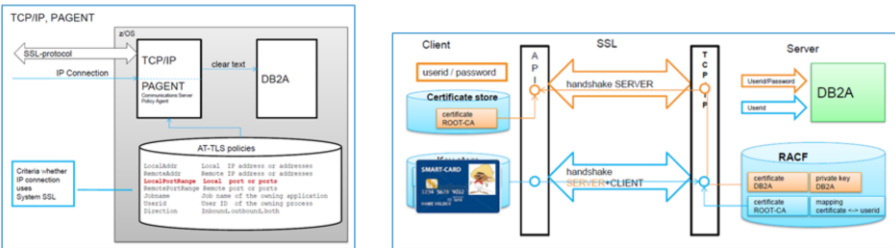
Very likely, you will have the question what is the value of z/OS Connect if DB2 supports RESTful services. Please revisit slide #9. DB2 integrates with z/OS Connect to benefit from API mapping, Swagger description generation and single entry point and building on the major distributed technology.

I added a references to a blog that describes how to build a RESTful service. The story behind is that we delivered the first PTF and the blogger used some spare time over the weekend to try it out. He sent an email with the blog link on the following Monday. Looks like testing the new functionality was more interesting than the TV program. Check it out and share your experience.

Security of DB2 Native REST service connections

DB2® supports HTTPS REST service requests by using the z/OS® Communications Server IP Application Transparent Transport Layer Security (AT-TLS) capability. The policy-driven AT-TLS support is configured in the TCP/IP stack and performs the TLS check on behalf of DB2 by invoking the z/OS system SSL feature in the TCP layer.

To support HTTPS requests, you must use a secure port for SSL connections. Make sure that the DDF TCP/IP SQL Listener service is capable of listening to a secondary secure port for inbound SSL connections. DDF verifies that all connections to DB2 through the secure port are protected by AT-TLS policies.



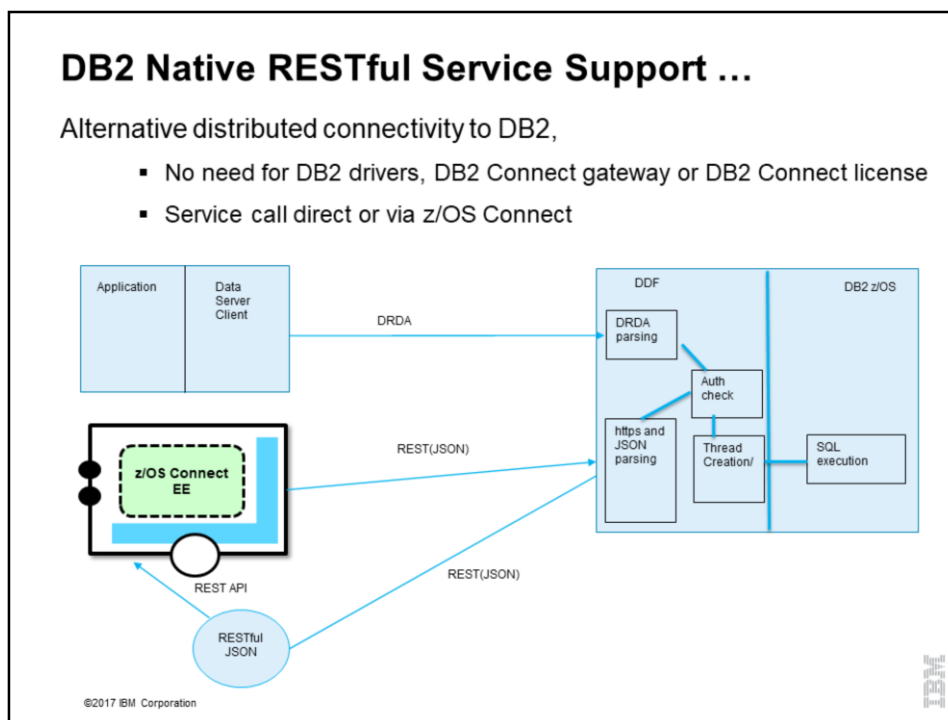
Setup SSL and client cert authentication

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4799.pdf>

<http://www.redbooks.ibm.com/redbooks/pdfs/sg248099.pdf> (chapter 12)

©2017 IBM Corporation

I mentioned it before, in order to accept service requests via HTTPS, a secure port needs to be defined for DB2. The 2 links are intended to give guidance on how to do it.



I added this picture to illustrate the DB2 connectivity options in DB2 11 with applied PTFs.

Everyone is familiar with the existing DB2 distributed support, where a distributed application communicates with DB2 server via DB2 Connect or Data Server Driver for JDBC/SQLJ or CLI/.NET. The used communication protocol is DRDA.

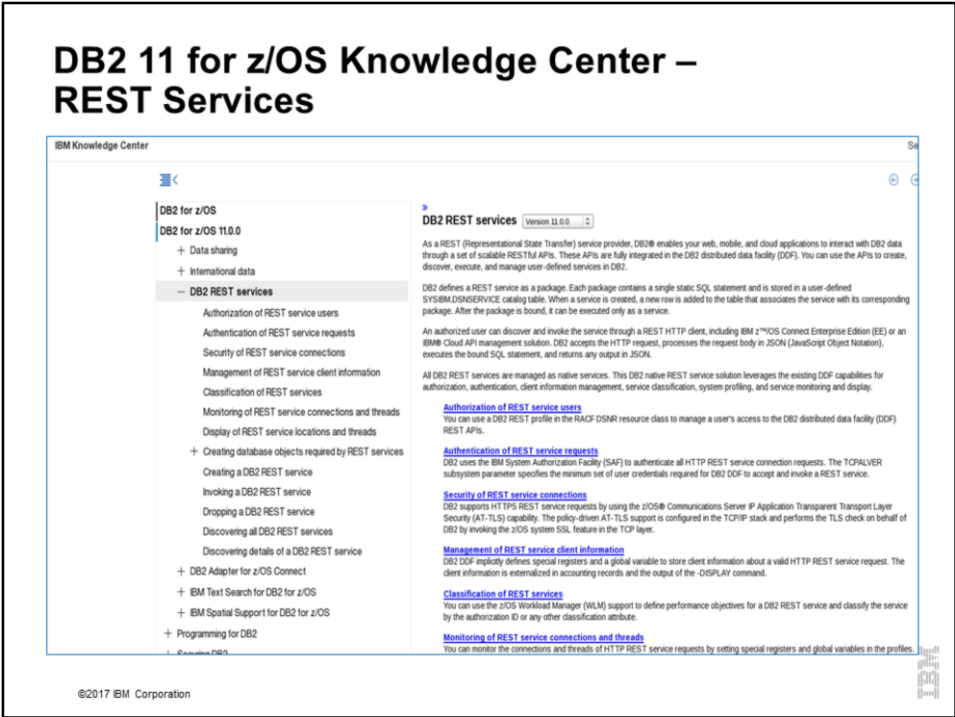
With the RESTful service support, an application communicates with DB2 server using REST(JSON) as communication protocol. z/OS Connect would be used as service gateway for added functionality. DB2 Data Server Driver or DB2 Connect gateway is not required and therefore no DB2 Connect license is needed for this type of connectivity.

z/OS Connect EE Configuration Options with DB2 V11

P RESTAPI
P Program
C COBOL
S Static SQL
D Dynamic SQL

Diagram	Hop #	Pay load	Network Protocol	SQL Statement	zIIP Offload	Dev. Skill
	3	DB2 native data	TCP/IP/ Sync Inbound	Coded in Client App. Dynamic SQL	DDF connection part only	Easy
	4	DB2 native data	TCP/IP/ Sync Inbound	Coded in Client App. Dynamic SQL	DDF connection part only	Easy+
	4	CTG native data	TCP/IP/ Sync Inbound	Coded in CICS App. Static SQL	Some CTG portion	Mode rate
	5	MQ MSG	TCP/IP/ Asyn In & Out bound	Coded in CICS App. Static SQL	All GCP	Mode rate+
	3	REST/JSON	HTTP/ Sync Inbound*	Defined in DB2 z/OS. Static SQL	DDF connection partly. 90%+ on z/OSCEE	Super Easy
	4	REST/JSON	HTTP/ Sync Inbound*	Coded in CICS App. Static SQL	90%+ on z/OSCEE	Easy

Note* : zOSCEE is planning to support outbound. (But, subject to be changed)



Please search the DB2 11 Knowledge Center for documentation on REST services support.

II14827: DB2 For z/OS Native RESTful Services – Additional Documentation

Known Restrictions/Limitations at DB2 V11 APAR PI70477 level, and DB2 V12 APAR PI70652 levels with the plan to address by a future APAR:

- A DB2 RESTful service created with an SQL statement can not be executed on IBM DB2 Analytics Accelerator for z/OS (IDAA).
- DB2 system profile monitoring is not supported for a DB2 RESTful service invocation.
- A DB2 RESTful service created with an SQL statement which refers to data at a remote DRDA location is not supported.
- Data Studio does not support the following DB2 Native REST service create features
 - Package bind options
 - Package collection identifier, default used = SYSIBMSERVICE

©2017 IBM Corporation



Summary

- Emerging business model “API Economy” - leveraging APIs to generate new business
 - Companies are now exposing some of their key application assets as services to other users/application developers or external businesses
- DB2 Adapter for z/OS Connect is available for DB2 10 and DB2 11
 - Exploits z/OS Connect to REST(JSON)-enable SQL and Stored Procedures
 - It is a good start to get familiar with functionality when still on DB2 10
- DB2 11 RESTful service support provides a new native interface into DB2 for z/OS
 - Recommended functionality for DB2 11 NFM or later
- Integrates with z/OS Connect to provide a consistent REST(JSON) interface into z/OS- based business functions and development tooling support

29

©2017 IBM Corporation



Youtube

- DB2 11 and 12 for z/OS Native REST Services – Part #1
Demonstrates how to create and invoke DB2 native REST services using SQL and stored procedures

<https://youtu.be/6XkiEysnsCg>

- DB2 11 and 12 for z/OS Native REST Services and z/OS Connect EE 2.0 – Part #2:
- The video shows how to enhance DB2 REST services using z/OS Connect EE 2.0 and transform them into REST APIs.
- <https://youtu.be/91ji0w01E3A>





IDUG DB2 EMEA Tech Conference
Lisbon Portugal | October 2017

 #IDUGDB2

Maryela Weihrauch, weihrau@us.ibm.com
IBM

DB2 Adapter for z/OS Connect –
SQL/Stored Procedures Play a Role
in Mobile and API Economics

Session code: F16

*Please fill out your
session evaluation
before leaving!*



31