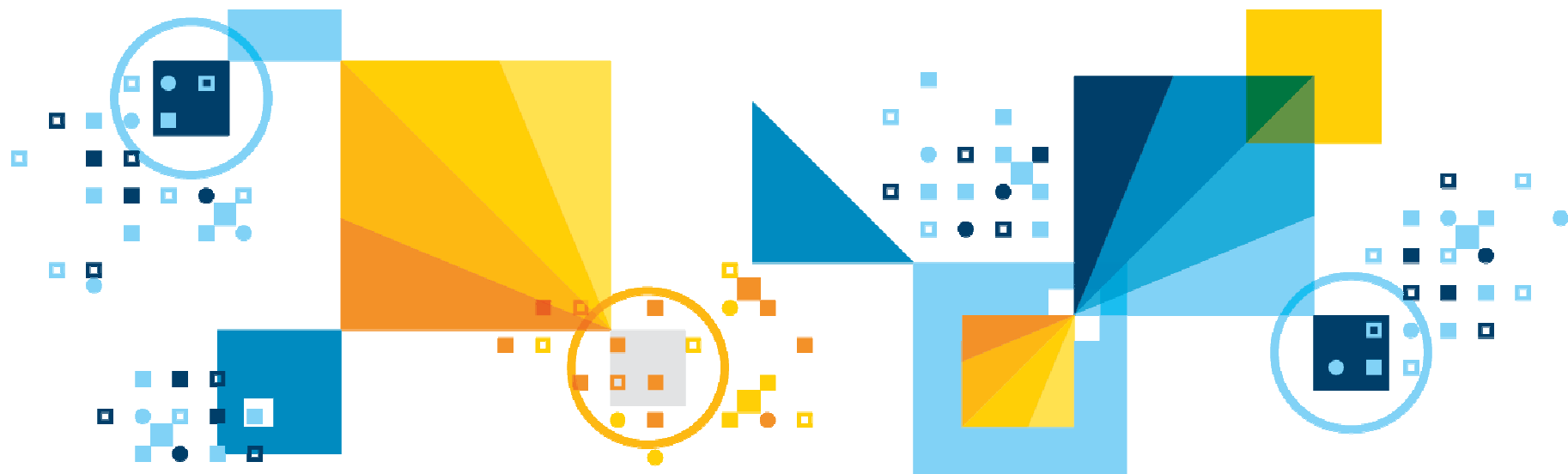


George Baklarz: baklarz@ca.ibm.com

DB2 RESTful Access Using OData with DB2 Data Sources



Safe Harbor Statement

Copyright © IBM Corporation 2017. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication, or disclosure restricted by GSA ADP Schedule Contract with IBM Corporation

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. 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. IN ADDITION, THIS INFORMATION IS BASED ON CURRENT THINKING REGARDING TRENDS AND DIRECTIONS, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. FUNCTION DESCRIBED HEREIN MAY NEVER BE DELIVERED BY IBM. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. 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 ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

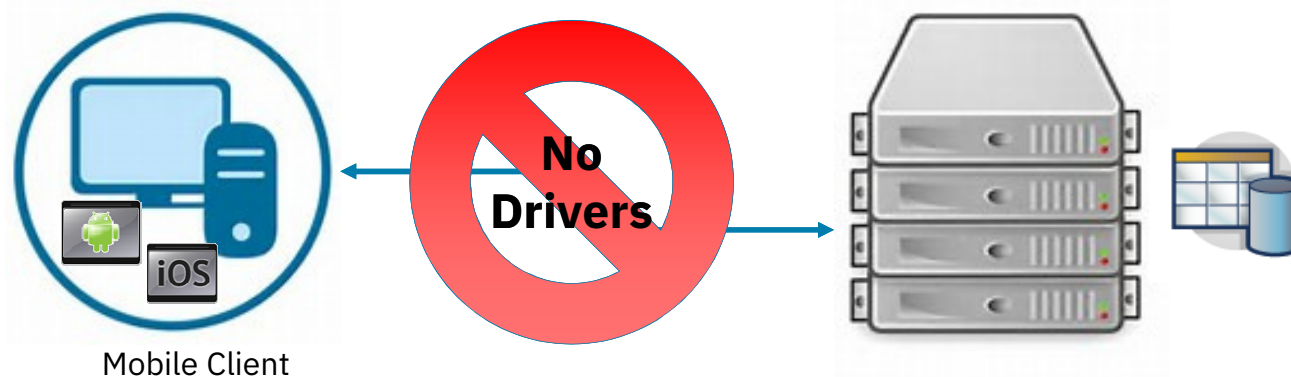
IBM, the IBM logo, ibm.com and Db2 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), 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 “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

Agenda

- **An Introduction to Open Data Protocol (OData)**
- **Installing the IBM Data Server Gateway for OData**
 - Where to obtain the driver
 - Installation Instructions
 - Starting and stopping the service
- **Using the Web Console to Generate a Service**
- **OData Support**
 - OData protocol syntax
 - Creating a service through an application
 - SELECT, INSERT, DELETE, and UPDATE statements
- **Jupyter Notebook Integration**
- **Reference Material**

Data Without Borders

- **Customers want to access Data from non traditional platforms like Mobile platforms, from mobile apps (Android, Windows, iOS, etc.)**
 - And in some cases want to publish access to data for 3rd parties
- **Mobile and Cloud developers expect direct HTTP and JSON access to data without requiring Database Drivers**
- **OData is standards based way for our customers to leverage Db2 in modern Cloud and Mobile Application Models**



A Brief Introduction to OData

*"OData (Open Data Protocol) is an **ISO/IEC approved, OASIS standard** that defines a set of best practices for building and consuming **RESTful APIs**."*

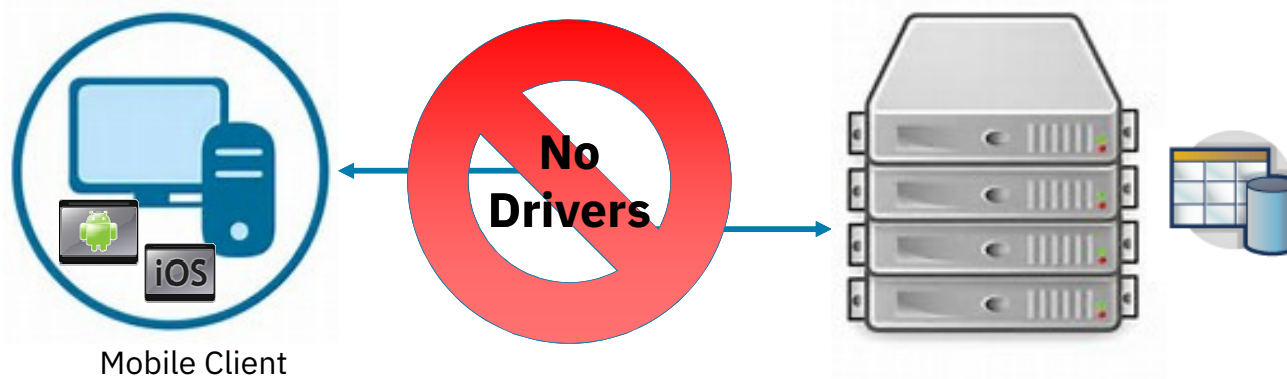
*OData helps you **focus on your business logic** while building **RESTful APIs** without having to worry about the various approaches to define request and response headers, status codes, HTTP methods, URL conventions, media types, payload formats, query options, etc.*

OData also provides guidance for tracking changes, defining functions/actions for reusable procedures, and sending asynchronous/batch requests."

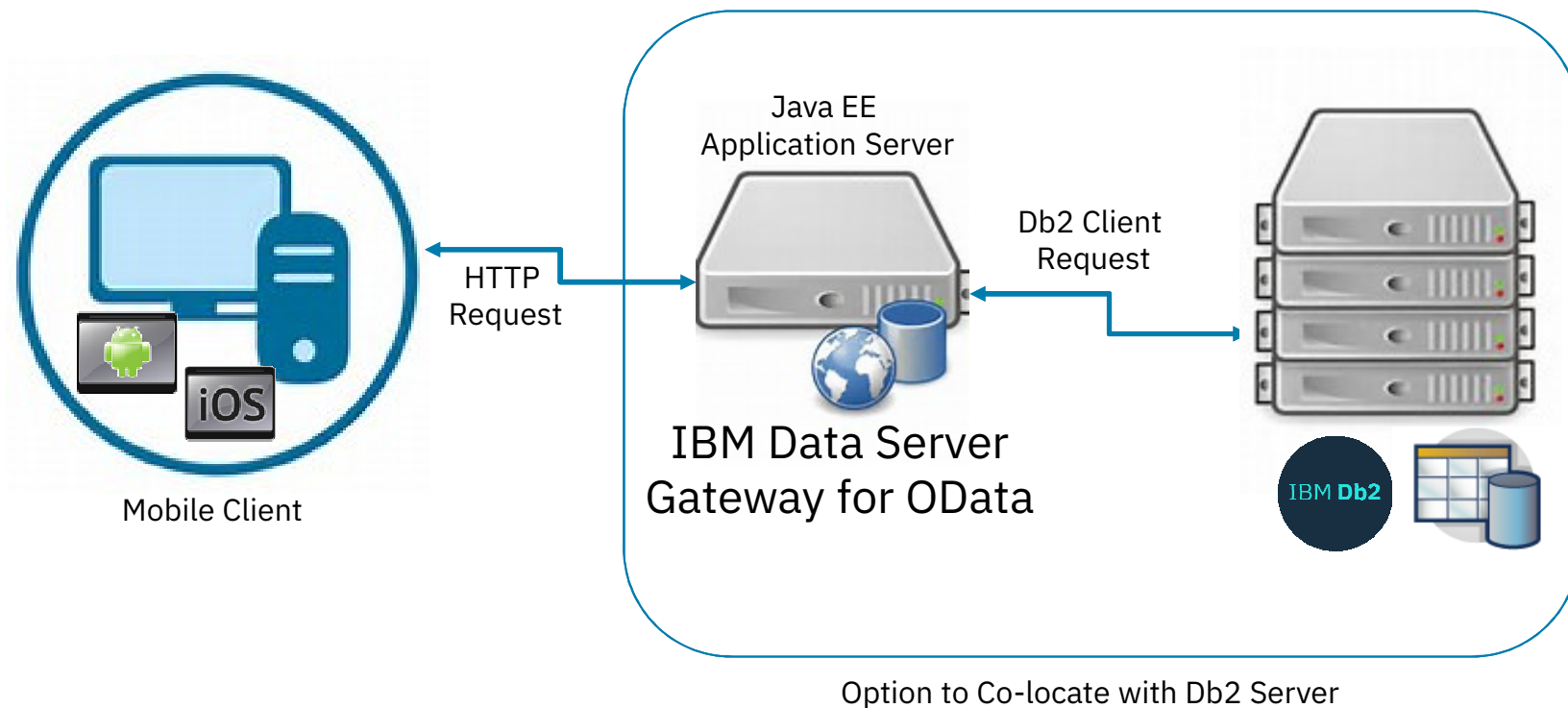
<http://www.odata.org/>

IBM Data Server Gateway for OData

- The **IBM Data Server Gateway for OData** (OData Gateway) is a standards-based way to leverage DB2 in Cloud and Mobile applications
- No requirement for a database driver with data access available from non-traditional platforms like Mobile platforms (Android, Windows, iOS etc.) using REST API endpoints
- Reduces the overall application complexity by eliminating the need to write middle-tier components that do nothing but transform data into JSON or XML



OData Gateway Architecture



IBM DS Gateway for OData Code

- ***IBM Data Server Gateway for OData*** is released in two formats

1. Integrated with WebSphere Liberty profile

- IBM Data Server Gateway for OData release is integrated with WebSphere Liberty Profile Application server
- Configured with a default server that enables you to quickly start the server and be productive


2. Web Application Resource (WAR) file

- IBM Data Server Gateway for OData is also released as a WAR file that can be deployed to any Java servlet container (Application server) of your choice

IBM DS Gateway for OData Download

▪ Download location:

- https://www-945.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm~Information%2BManagement&product=ibm/Information+Management/IBM+Data+Server+Client+Packages&release=11.1.*&platform=Linux&function=fixId&fixids=*odata*FP001*&includeSupersede s=0&source=fc
- Or use the quick link! ibm.biz/Db20Data

<input type="checkbox"/>	1	fix pack:  DSclients--ibm_gateway_server_for_odata_war-1.0.0-FP001 IBM Data Server Gateway for ODATA WAR File V1.0.0 Fix Pack 1	2016/12/13
<input type="checkbox"/>	2	fix pack:  DSclients--ibm_gateway_server_for_odata-1.0.0-FP001 IBM Data Server Gateway for ODATA V1.0.0	2016/12/13

Download with Integrated Server

Download files using HTTPS

Information Management, IBM Data Server Client Packages (11.1.*, Linux)

 [Subscribe to support notifications](#)

Download files using your web browser

Click the download link next to each file to download it.

Order number:	268763861
---------------	-----------


Total size:	83.97 MB
-------------	----------

[Show normalized list](#) | [Hide normalized list](#)

fix pack: DSclients--ibm_gateway_server_for_odata-1.0.0-FP001

IBM Data Server Gateway for ODATA V1.0.0

The following files implement this fix.

 [v11.1.1fp1_ibm_gateway_server_for_odata.zip](#) (83.97 MB)

Installation Instructions

- **Create a Directory to hold the IBM Data Server Gateway for OData file**
 - `mkdir db2odata` or `md db2odata`
- **Copy the zip file into the newly created directory and unzip it**
 - `pkunzip` or `tar -xvf` the file
 - This will create a directory called `v1.0.0_ibm_gateway_server_for_odata`
- **Change the Permission bits on the following files (UNIX, Linux)**
 - `chmod 555 start_ibm_gateway_server_odata`
 - `chmod 555 stop_ibm_gateway_server_odata`
 - `chmod 555 server/bin/server`

Starting and Stopping the Application Server

- **To Start the server:**

- In Windows run `start_ibm_gateway_server_odata.bat`
- In Linux, AIX or Mac run `./start_ibm_gateway_server_odata`

- **Make sure the permission bits have been updated or otherwise you will get a Permission denied error!**

- **To Stop the server:**

- In Windows run `stop_ibm_gateway_server_odata.bat`
- In Linux, AIX or Mac run `./stop_ibm_gateway_server_odata`

- **Note – The service requires that port 9080 be open on your machine**

- The following command are examples only and will be different depending on your O/S and firewall standards

```
sudo iptables -I INPUT -p tcp --dport 9080 -j ACCEPT  
sudo service iptables save
```

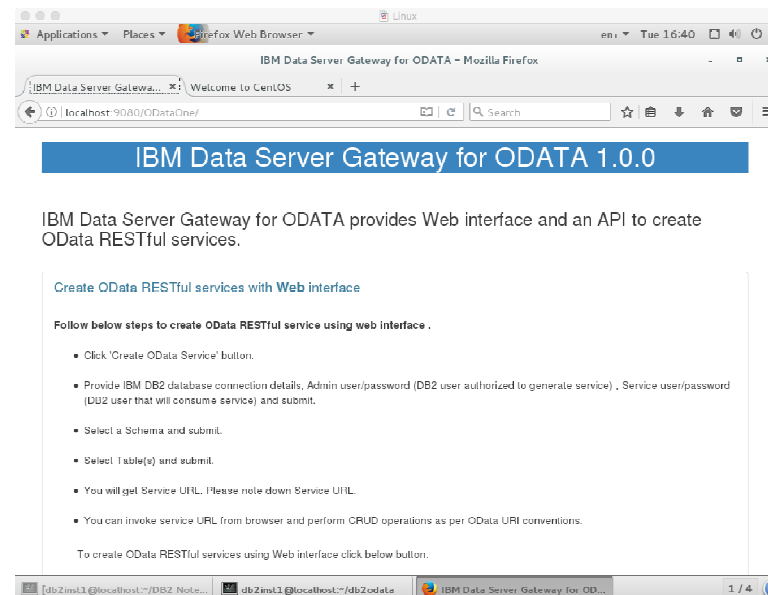
Validating Installation

- When starting the service, you should see the following output:

```
db2inst1@localhost:~/db2odata
File Edit View Search Terminal Help
[db2inst1@localhost ~]$ cd db2odata
[db2inst1@localhost db2odata]$ ./start*
Starting server ibmgatewayserverodata.
Server ibmgatewayserverodata started with process ID 33963.
[db2inst1@localhost db2odata]$
```

- After the service has started, use a web browser to view the service:

<http://localhost:9080/ODataOne/>



IBM DS Gateway for OData Support

- The gateway is based on the OData V4.0 REST API and provides endpoints and services for **selected database tables**
- The IBM OData Gateway provides a RESTful service that provides for the following operations:
 - GET (SELECT)
 - POST (INSERT)
 - PUT (UPDATE a full row)
 - PATCH (UPDATE selected columns in a row)
 - DELETE (Delete a row)
- All data types are supported **except** LOBs (CLOB, BLOB, DBCLOB) and XML

IBM DS Gateway for OData Support

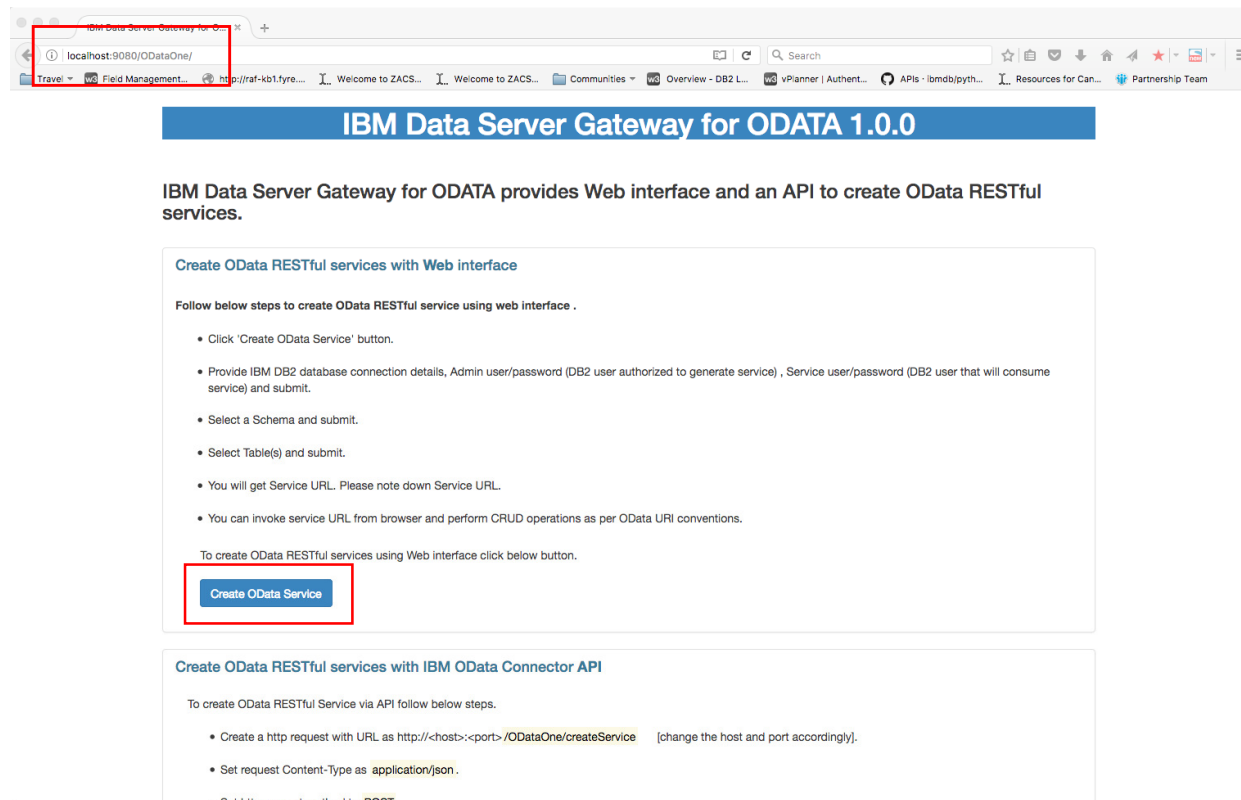
- **Persistence of REST API endpoints/services are kept in an Apache Derby database**
 - Small footprint database used for mapping REST API calls to resources in the Db2 database
- **Authentication and Authorization are used during OData REST API endpoints and service generation**
 - Client needs to authenticate mobile user using whatever local protocol is applicable
 - Password
 - Fingerprint
 - Two-factor authentication
 - Authentication is not handled as part of the OData implementation
- **SSL/TLS supported**
 - Secure transmission of requests and retrieval of data

Generating a Service Using Web Console⁽¹⁾

- Use the following URL to access the service console:

<http://localhost:9080/ODataOne/>

- Click on the Create OData Service button



Generating a Service Using Web Console⁽²⁾

- Fill in the following fields

IBM Data Server Gateway for ODATA 1.0.0

Configure IBM DB2 DataSource for OData Service Generation

DATA SOURCE PROPERTIES

Database Name:

Hostname:

Port Number:

DB2 Administrator User :

DB2 Administrator Password:

DB2 User for OData Service:

DB2 Password for OData Service:

☐ Use SSL

Submit

Reset

Generating a Service Using Web Console⁽³⁾

▪ Database Name

- The Database that the service will connect to

▪ Host Name and Port Number

- Host name and port number that the database is located on
- If the database is local to the OData service then use `localhost`

▪ Db2 Administrative User and Password

- Db2 administrator privileges are required to create the service, but these credentials are NOT used for accessing the data

▪ Db2 Userid and password for OData Service

- The userid and password that will be used to access the tables that you will define as part of the service
- This could be the same as the administrative user when you are testing and debugging your applications, but should not be the same in production!

Generating a Service Using Web Console⁽⁴⁾

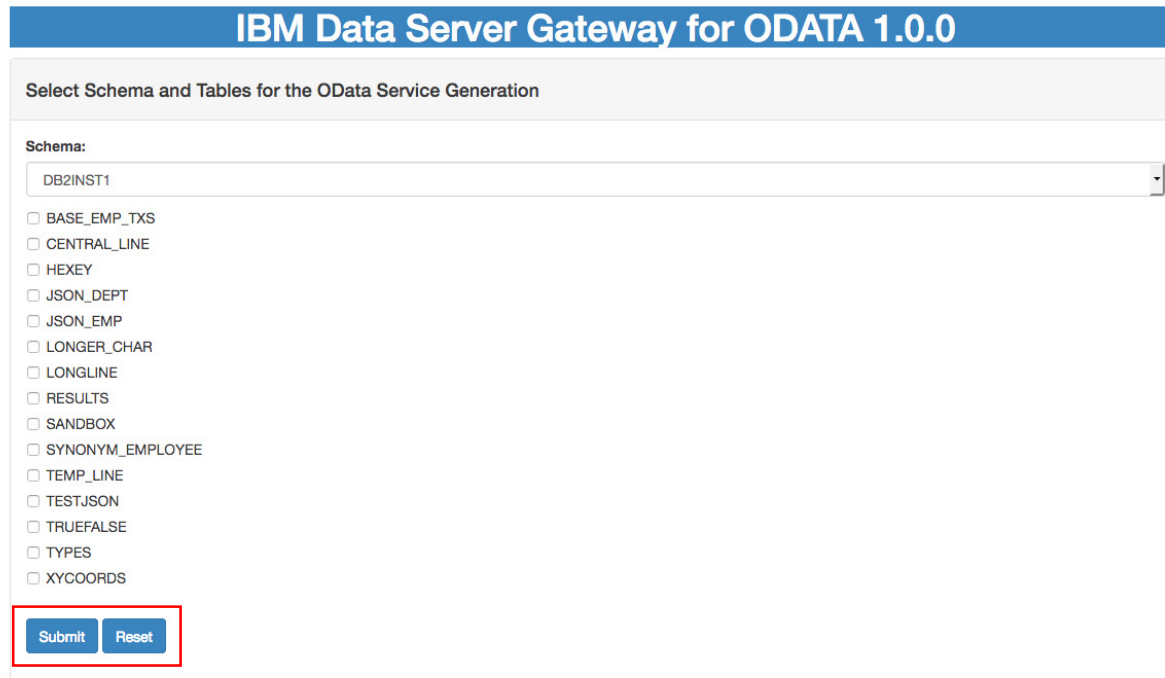
- Press the submit button to test the connection

The screenshot shows a web browser window with the URL `localhost:9080/ODataOne/generateService`. The page title is "IBM Data Server Gateway for ODATA 1.0.0". A blue banner at the top displays the title. Below it, a light blue box indicates "Connection Successful!". The main section is titled "Select Schema and Tables for the OData Service Generation". It contains a "Schema:" label and a dropdown menu with "DB2DEMO" selected. At the bottom of this section, there are two buttons: "Submit" and "Reset", which are highlighted by a red rectangular box.

- Select the Schema that you want to use and press submit

Generating a Service Using Web Console⁽⁵⁾

- Select the tables that you want the service to access



IBM Data Server Gateway for ODATA 1.0.0

Select Schema and Tables for the OData Service Generation

Schema:

DB2INST1

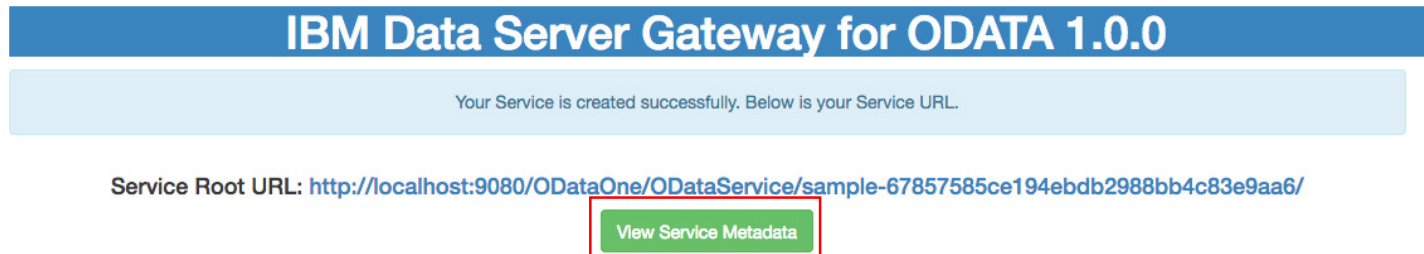
- ☐ BASE_EMP_TXS
- ☐ CENTRAL_LINE
- ☐ HEXEY
- ☐ JSON_DEPT
- ☐ JSON_EMP
- ☐ LONGER_CHAR
- ☐ LONGLINE
- ☐ RESULTS
- ☐ SANDBOX
- ☐ SYNONYM_EMPLOYEE
- ☐ TEMP_LINE
- ☐ TESTJSON
- ☐ TRUEFALSE
- ☐ TYPES
- ☐ XYCOORDS

Submit Reset

- Press submit to create the service

Generating a Service Using Web Console⁽⁶⁾

- **Service URL is shown on the final panel**



- This service root URL is used for all OData requests
- Adding \$metadata to the end of URL will display the tables and columns that can be accessed through the URL

```

<extends:Edm.Version>4.0</extends>
<SchemaName>'sample'</SchemaName>
  <EntityType>'CENTRAL_LINK'</EntityType>
    <Property Name='STATION' Type='Edm.String' />
    <Property Name='STATION_ID' Type='Edm.Int32' />
    <Property Name='UPPER_STATION' Type='Edm.String' />
    <EntityType />
    <EntityType Name='JSON_DEPT'</EntityType>
    <Property Name='DEPT_DATA' Type='Edm.Stream' />
    <Property Name='SIQ' Type='Edm.Int32' />
    <EntityType />
    <EntityType Name='JSON_IMP'</EntityType>
    <Property Name='IMP_DATA' Type='Edm.Stream' />
    <Property Name='SIQ' Type='Edm.Int32' />
    <EntityType />
    <EntityType Name='LONGLINE'</EntityType>
    <Property Name='NAME' Type='Edm.String' />
    <EntityType />
    <EntityType Name='RESULTS'</EntityType>
    <Property Name='RESULT' Type='Edm.Decimal Precision=35' Scale=22 />
    <Property Name='TYPE' Type='Edm.String' />
    <EntityType />
    <EntityType Name='SANDBOX'</EntityType>
    <Property Name='DATA' Type='Edm.Stream' />
    <EntityType />
    <EntityType Name='SYNONYM_EMPLOYEE'</EntityType>
    <Property Name='BONUS' Type='Edm.Int16' />
    <Property Name='BONUS_RATE' Type='Edm.Double' />
    <Property Name='COMMISSION' Type='Edm.Int12' />
    <Property Name='COMMISSION_RATE' Type='Edm.Double' />
    <Property Name='NAME' Type='Edm.String' />
    <Property Name='SALARY' Type='Edm.Int32' />
    <EntityType />
    <EntityType Name='TESTSTRING'</EntityType>
    <Property Name='JSON_FIELD' Type='Edm.Stream' />
    <EntityType />
    <EntityType Name='TRUEFALSE'</EntityType>
    <Property Name='EXAMPLE' Type='Edm.Int32' />
    <Property Name='STATE' Type='Edm.Boolean' />
    <EntityType />

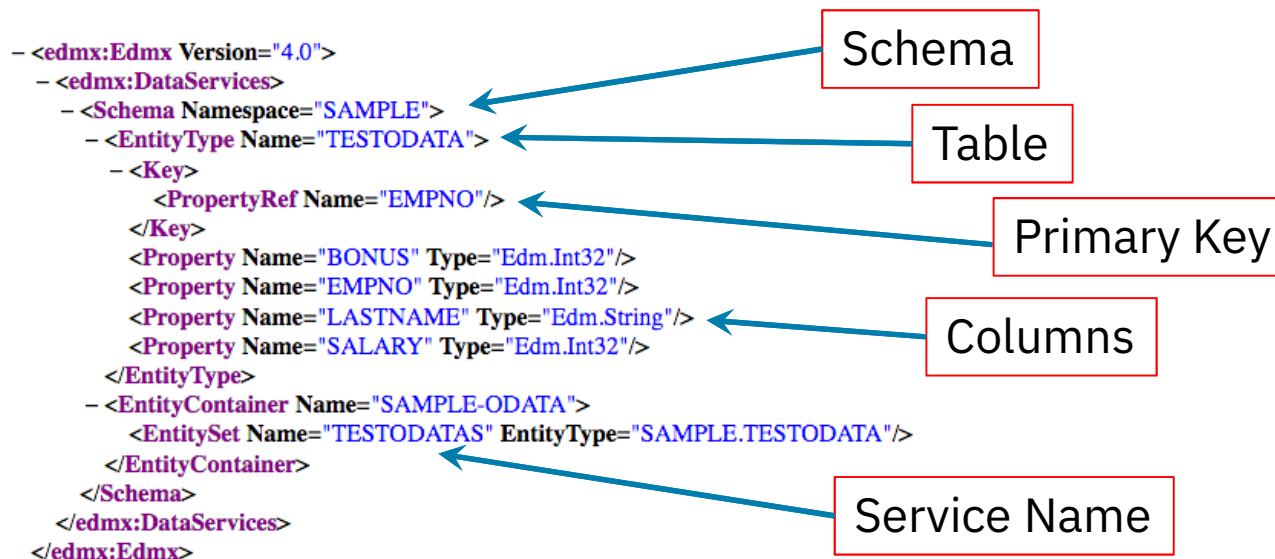
```

Metadata Information

- The service root URL can be used to view metadata information about the tables defined to it:

[http://localhost:9080/0DataOne/0DataService/SAMPLE-db2ab6eeb40d4a75a525333ee610fb50/\\$metadata](http://localhost:9080/0DataOne/0DataService/SAMPLE-db2ab6eeb40d4a75a525333ee610fb50/$metadata)

- Detailed information returned about tables, primary keys, and individual column attributes are returned in an XML document^(PIA)



Metadata Data Types

- The OData data types do not map exactly to Db2 data types
- The following table gives the OData keyword value (Edm.type) and the data type it represents

Data Type	Contents
Binary	Binary data
Boolean	Binary-valued logic
Byte	Unsigned 8-bit integer
Date	Date without a time-zone offset
Decimal	Numeric values with fixed precision and scale
Double	IEEE 754 binary64 floating-point number (15-17 decimal digits)
Duration	Signed duration in days, hours, minutes, and (sub)seconds
Guid	16-byte (128-bit) unique identifier
Int16	Signed 16-bit integer
Int32	Signed 32-bit integer
Int64	Signed 64-bit integer
SByte	Signed 8-bit integer
Single	IEEE 754 binary32 floating-point number (6-9 decimal digits)
String	Sequence of UTF-8 characters
TimeOfDay	Clock time 00:00-23:59:59.999999999999

Generating a Service Using RESTful Calls⁽¹⁾

- **Create a JSON string with the following content**

```
{  
  "database": "<database_name>",  
  "host": "<database_host>", "port": "<database_port>",  
  "db2AdminUser": "<db2_admin_user>",  
  "db2AdminPassword": "<db2_admin_password>",  
  "db2ServiceUser": "<db2_service_user>",  
  "db2ServicePassword": "<db2_service_password>",  
  "ssl": false,  
  "schema": "<schema_name>",  
  "tablenames": ["table1", "table2",...]  
}
```

- **Send a POST request with the above string to:**

`http://<host>:<port>/ODataOne/createService`

- **Notes:**

- Content-Type must be set to application/json
- The JSON string must be in the request header

Generating a Service Using RESTful Calls⁽²⁾

- **OData service will return the following information**

Your service is created successfully !

Service Root URL:

`http://<server_host>:<port>/ODataOne/ODataService/sample-a1085cdc671c4b5dbd969c3be3e411e8`

Service Metadata URL:

`http://<server_host>:<port>/ODataOne/ODataService/sample-a1085cdc671c4b5dbd969c3be3e411e8/$metadata`

- **The Service Root URL is used for subsequent OData calls**

OData RESTful Calls

- **The IBM OData Gateway provides a RESTful service that provides for the following operations:**
 - GET (SELECT)
 - POST (INSERT)
 - PUT (UPDATE a full row)
 - PATCH (UPDATE selected columns in a row)
 - DELETE (Delete a row)
- **The POST, PUT, PATCH, and DELETE requests require a primary key to be available**
 - GET can also use a primary key, but it is not required if using filtering (predicates)
 - Views are also a way of combining tables into a more sophisticated result set (but update operations would not be supported)

OData RESTful Calling Structure

```
SELECT LASTNAME, SALARY  
FROM    EMPLOYEE  
WHERE   SALARY > 50000
```

Service Address

```
url = http://localhost:9080/ODataOne/ODataService/SAMPLE-  
db2ab6eeb40d4a75a525333ee610fb50
```

```
header = {  
    'Content-Type': 'application/json',  
    'Accept': 'application/json'  
}
```

Handshake

```
Odata = "/EMPLOYEES?$select=LASTNAME,SALARY&$filter=SALARY gt 50000"
```

```
RESTful.get(url + Odata, headers=header)
```

OData Request

OData RESTful Syntax

▪ Header file

- The POST, GET, PATCH, PUT, and DELETE RESTful all use a common header that tells the service how the content is formatted (JSON)
`{'Content-Type':'application/json','Accept':'application/json'}`

▪ Service URL

- URL refers to the service URL that allows access to the table(s) that you have registered in the IBM OData Gateway

▪ Service Name

- Db2 table name is appended with the letter 'S' becomes the service name^(PIA)
- i.e. `EMPLOYEE` -> `EMPLOYEES`

▪ OData command is always appended to the end of the Service URL

- `RESTful.get(url + OData, headers=header)`

▪ A ? is added to the end of the service name if there are optional parameters in the OData command

OData GET (SELECT) Command

- The OData GET command will return data from **one** table or view
- The select is accomplished through the RESTful GET request:
 - `RESTful.get(url + OData, headers=header)`
- The OData request contains the name of the table service and the optional parameters used to filter and format the rows
`/EMPLOYEES?$select=LASTNAME,SALARY&$filter=SALARY gt 50000`

OData GET Syntax

- **The GET command will return data from one table**

- There is no ability to join tables with the current implementation of OData
- You can use views in Db2 as a way to join tables and use that in your selects
- The table name in the OData command always has an "S" appended to it since it refers to the service name

- **Syntax:**

- `/<table>S(primary key)[?options]`
- `/<table>S[?options]`

- **Options:**

- `$select=column1, columns, ...`
- `$filter=logic`
- `$top=x [number of rows]`
- `$format=json`
- `$count=true`

OData GET Options⁽¹⁾

- **The Options in a GET clause can be used in any order but must be separated by an ampersand (&) character**

`$select=NAME&$top=1`

- **Options are case and whitespace sensitive**

- You must place spaces between arguments in the \$filter option

- **\$select**

- This clause specifies which columns you want retrieved from the table
- You must use the UPPERCASE name of the column if the table was created using the traditional Db2 naming rules (i.e. the name is folded to uppercase)
- The order makes no difference in how they are returned
- Columns are always returned in alphabetical order, regardless of how you specify them in the command^(PIA)
- Commas separate the column names with no extra spaces
- Example: `$select=LASTNAME,FIRSTNME`

OData GET Options⁽²⁾

- **\$top=x**

- Limits the result set to the first "x" rows from the top
- Example: `$top=5`

- **\$format=json**

- Return the result as JSON records (default)

- **\$count=true**

- Return the count of records found
- This also returns the actual rows, so this isn't the same as a Db2 COUNT(*) clause^(PIA)
- Best practice would be to limit the output to one column in the answer set to minimize the data returned if all you really wanted was the count
- Using \$top will result in \$count reflecting ONLY the records returned, not the true count of the table^(PIA)

OData GET Options⁽³⁾

- **\$filter**

- Used to apply filtering logic to the rows being returned

- **Comparison operators**

- `lt` (<), `gt` (>), `le` (<=), `ge` (>=), `ne` (<>), `eq` (=)

- **Logical Operators**

- `and`, `or`, `(,)`
- No "not" available^(PIA)

- **Mathematical Operators**

- No math operators are allowed, including the unary minus operator^(PIA)
- Use a VIEW to do these calculations i.e. `COMM + BONUS > SALARY * .20`

- **String Operator**

- `contains(field,pattern)`
- Similar to the LIKE clause except that it searches for the pattern anywhere

OData GET Examples⁽⁴⁾

- Select employee '000010' with all columns returned (using primary key)
`/EMPLOYEES('000010')`
- Select employee '000010' with their first and last name retrieved
`/EMPLOYEES?$select=FIRSTNAME, LASTNAME&$filter=EMPNO eq '000010'`
- Return the count of all employees (which includes their records too!)
`/EMPLOYEES?$count=true`
- Select the LASTNAME and SALARY of the employees whose salary is greater than 50000 and limit the output to the first 5
`/EMPLOYEES?$select=LASTNAME, SALARY&$filter=SALARY gt 50000&$top=5`
- Select the LASTNAME and BONUS of the employees who last name has the letters 'AA' in it, and whose bonus is equal to 1000
`/EMPLOYEES?$select=LASTNAME, BONUS&$filter=contains(LASTNAME, 'AA') and BONUS eq 1000`
- Select the employees who work in department A00 or work in D01 and have a salary > 60000
`/EMPLOYEES?$filter=WORKDEPT eq 'A00' or (WORKDEPT eq 'D01' and SALARY gt 60000)`

OData POST Format⁽¹⁾

- The OData POST command will INSERT a single row into a table
- An INSERT is accomplished through the RESTful POST request:
 - `RESTful.post(url + OData, headers=header, parameters)`
- The OData request contains the name of the table service
`/EMPLOYEES`
- The OData request is appended to the end of the URL:
`http://localhost:9080/ODataOne/ODataService/SAMPLE-db2ab6eeb40d4a75a525333ee610fb50/EMPLOYEES`

OData POST Format⁽²⁾

- The parameter field is a JSON record that has the names and contents of all of the columns

```
{  
  'EMPNO': 1,  
  'LASTNAME': 'Fred',  
  'SALARY': 10000,  
  'BONUS': 1000  
}
```

- The data is checked by Db2 to make sure it has the proper format
 - If a column name is not provided, the default value will be NULL
 - If the column is defined as NOT NULL (without a default), then an error will be returned

SQL0407N Assignment of a NULL value to a NOT NULL column "TBSPACEID=2, TABLEID=22, COLNO=2" is not allowed. SQLSTATE=23502
SQLCODE=-407

OData PATCH Format⁽¹⁾

- The OData PATCH command will UPDATE fields in a single row in a table
- An UPDATE is accomplished through the RESTful PATCH request:
`RESTful.patch(url + OData, headers=header, parameters)`
- The OData request contains the name of the table service and the primary key of the record to be updated

`/EMPLOYEES('000010')`

- You must provide a unique key for the record to update
 - Multi-row updates are not supported with this syntax
- You can replace all of the values in a record by using the PUT command and providing all of the fields and values in the parameter list

OData PATCH Format⁽²⁾

- The parameter field is a JSON record that has the names and contents of all of the columns that you want to change

```
{  
  'SALARY': 10000,  
  'BONUS': null  
}
```

- The data is checked by Db2 to make sure it has the proper format
 - If the column is defined as NOT NULL (without a default), then an error will be returned if you try to set it to NULL

SQL0407N Assignment of a NULL value to a NOT NULL column "TBSPACEID=2, TABLEID=22, COLNO=2" is not allowed. SQLSTATE=23502
SQLCODE=-407

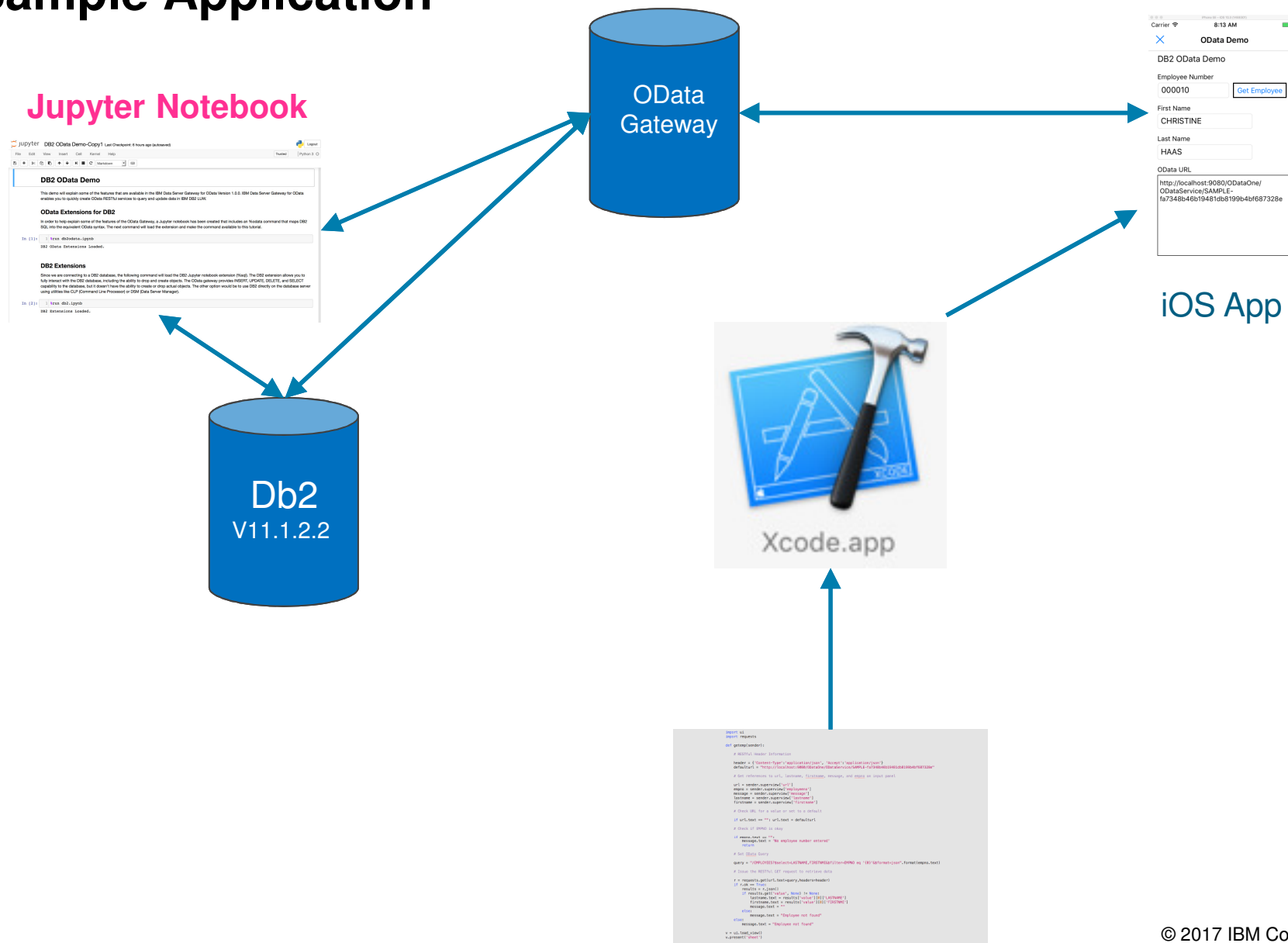
OData DELETE Format

- The OData DELETE command will delete a single row in a table
- An DELETE is accomplished through the RESTful DELETE request:
`RESTful.delete(url + OData, headers=header, parameters)`
- The OData request contains the name of the table service and the primary key of the record to be deleted
`/EMPLOYEES('000010')`
- You must provide a unique key for the record to delete
 - Multi-row deletes are not supported with this syntax

Sample Db2 OData Notebooks

- **Jupyter notebooks are available to try OData functions against Db2 11**
 - Includes a Jupyter notebook extension (%odata) to generate a valid service URL for requests to a Db2 Database
 - Translates SQL syntax for INSERT, UPDATE, DELETE, and SELECT commands into equivalent OData commands
 - <https://github.com/DB2-Samples/db2odata.git>
- **An introduction to the OData gateway is found in the following developerWorks article:**
 - <http://ibm.biz/Db20DataArticle>
- **The code can be obtained through the following link:**
 - ibm.biz/Db20Data
- **Videos**
 - https://www.youtube.com/watch?v=z06bl_K6Ckc
 - <https://www.youtube.com/watch?v=dHVBiTphoCM>

Sample Application



Expand Db2 Ecosystem/Skills

▪ New Db2 Developer-C Edition

- Freely downloadable just like Express-C
- Fully functional Db2 database server
 - pureScale and DPF deployments
 - Compression and BLU Acceleration

▪ Use Limitations

- For development and non-production only
- **Unsupported edition (non-warranted)**
- Environment limited usage:
 - 4 cores, 16GB of memory
 - 100GB of data in user tablespaces

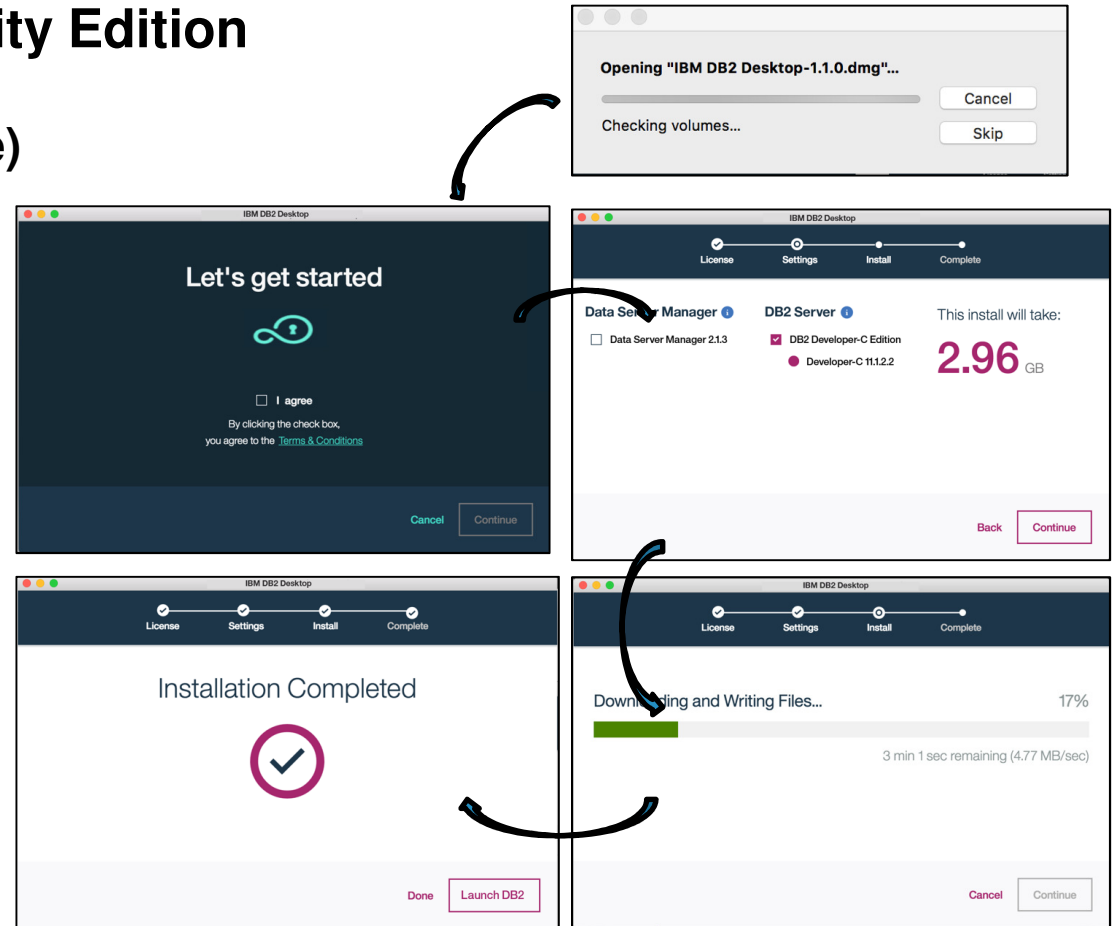
▪ Functional Differences from Db2 Developer Edition

- Excludes some “Supporting Programs”
 - Cognos, IDA, WebSphere AS / MQ, DSM Enterprise



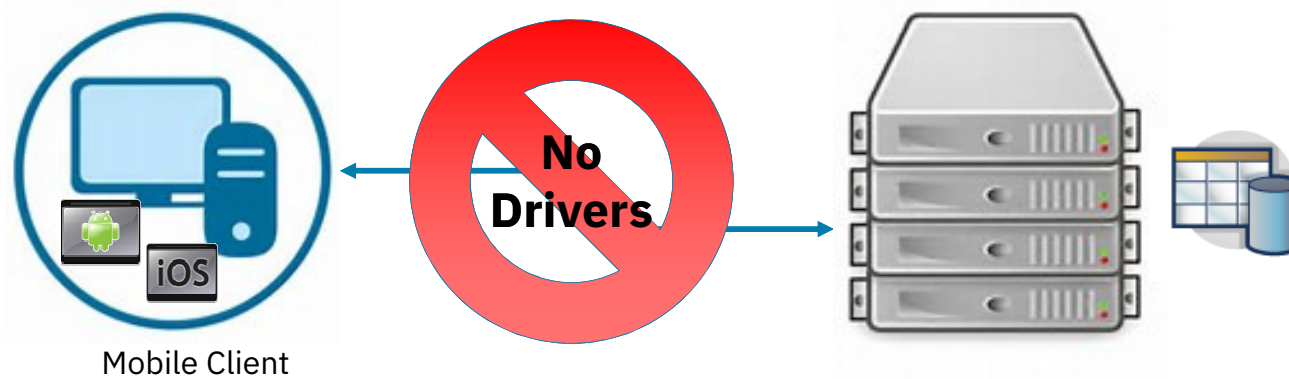
Announcing the new “Download and Go” experience! IBM Db2 Developer Community Edition

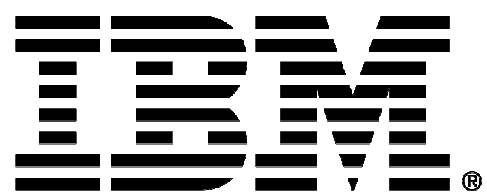
- Small download (Docker image)
less than 15 minutes from
“I want” to *“I got”*
- Platform specific executable
 - Mac, Windows, Linux
- From 10+ screens and
10+ mandatory input fields
to **3 clicks** and **userID/pwd
input only**
- Download package includes
 - Db2 Developer-C 11.1.2.2
 - Data Server Manager 2.1.4
 - Data Studio (optional)
- Same limitations as Developer-C
 - Development and **non-production only**
 - Core, memory, and storage **limitations**
 - **Unsupported** edition (non-warranted)



Summary

- The ***IBM Data Server Gateway for OData*** (OData Gateway) is a standards-based way to leverage DB2 in Cloud and Mobile applications
- No requirement for a database driver with data access available from non-traditional platforms like Mobile platforms (Android, Windows, iOS etc.) using REST API endpoints
- Reduces the overall application complexity by eliminating the need to write middle-tier components that do nothing but transform data into JSON or XML





Legal Disclaimer

- © IBM Corporation 2017. All Rights Reserved.
- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication 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.
- 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. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. 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.
- If the text contains performance statistics or references to benchmarks, insert the following language; otherwise delete:
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.
- If the text includes any customer examples, please confirm we have prior written approval from such customer and insert the following language; otherwise delete:
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.
- Please review text for proper trademark attribution of IBM products. At first use, each product name must be the full name and include appropriate trademark symbols (e.g., IBM Lotus® Sametime® Unyte™). Subsequent references can drop "IBM" but should include the proper branding (e.g., Lotus Sametime Gateway, or WebSphere Application Server). Please refer to <http://www.ibm.com/legal/copytrade.shtml> for guidance on which trademarks require the ® or ™ symbol. Do not use abbreviations for IBM product names in your presentation. All product names must be used as adjectives rather than nouns. Please list all of the trademarks that you use in your presentation as follows; delete any not included in your presentation. IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, Quickr, Sametime, WebSphere, UC2, PartnerWorld and Lotusphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. Unyte is a trademark of WebDialogs, Inc., in the United States, other countries, or both.
- If you reference Adobe® in the text, please mark the first use and include the following; otherwise delete:
Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- If you reference Java™ in the text, please mark the first use and include the following; otherwise delete:
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- If you reference Microsoft® and/or Windows® in the text, please mark the first use and include the following, as applicable; otherwise delete:
Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- If you reference Intel® and/or any of the following Intel products in the text, please mark the first use and include those that you use as follows; otherwise delete:
Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- If you reference UNIX® in the text, please mark the first use and include the following; otherwise delete:
UNIX is a registered trademark of The Open Group in the United States and other countries.
- If you reference Linux® in your presentation, please mark the first use and include the following; otherwise delete:
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.
- If the text/graphics include screenshots, no actual IBM employee names may be used (even your own), if your screenshots include fictitious company names (e.g., Renovations, Zeta Bank, Acme) please update and insert the following; otherwise delete: All references to [insert fictitious company name] refer to a fictitious company and are used for illustration purposes only.