



# REST APIs for Direct Db2 Access

Tim Rowe – [timmr@us.ibm.com](mailto:timmr@us.ibm.com)

Architect Application  
Development IBM i



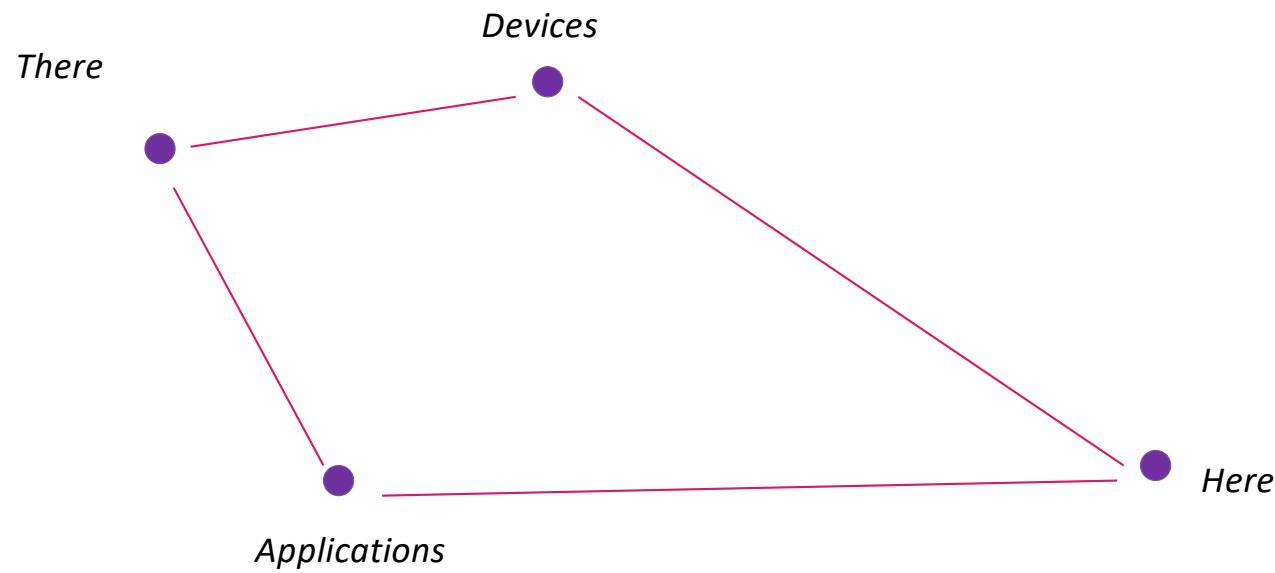


# What is an API - Agenda

- What is an API
- What is a Web Service
- SOAP vs REST
  - What is SOAP
  - What is REST
  - Benefits
  - Drawbacks
- REST for SQL ?? YES!!



# Connections



# API Definition



A  
P  
I

Application  
Programming  
Interface

# API Definition



## Application programming interface

From Wikipedia, the free encyclopedia

*"API" redirects here. For other uses, see [API \(disambiguation\)](#).*



In computer programming, an **Application Programming Interface (API)** is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, computer hardware or software library. An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables or remote calls. **POSIX**, **Microsoft Windows API**, the **C++ Standard Template Library** and **Java APIs** are examples of different forms of APIs. Documentation for the API is usually provided to facilitate usage.



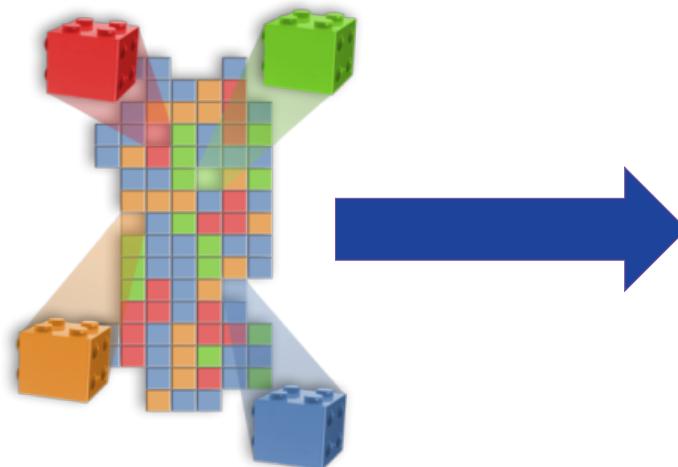
Simple way to connect endpoints. Send a request and receive a response.

# What is an API? It's a Web Service....what is a Web Service?



... a service?

A repeatable  
business task –  
e.g., check  
customer credit;  
open new account



Available on the Web



# SOAP vs REST



# Simple Object Access Protocol





# What is SOAP

- Exposes **operations** that implement logic
- Designed for distributed computing
- Standardized
- Aligns with Enterprise Application needs
  - Support multi transport
  - Enterprise security – WS.Security
  - Governance with strong typing
  - Broad Development tooling support
- XML Based message protocol
- Uses WSDL as a contract between consumer and provider

# REpresentational State



T  
ransfer



## What is REST

- Architectural Style as described by Roy Fielding
- Resource focused
- Every request is via hyperlink ie http request
- Easily consumed by any client, especially web clients
- Light weight
  - Uses JSON vs XML
  - No required header for each message
- Resources are driven by HTTP Specification
  - GET, PUT, DELETE, POST

# Why one vs the other ? Philosophical Difference



## SOAP

- Enterprise Driven
- Contract based
- Robust Infrastructure
- More Security Options

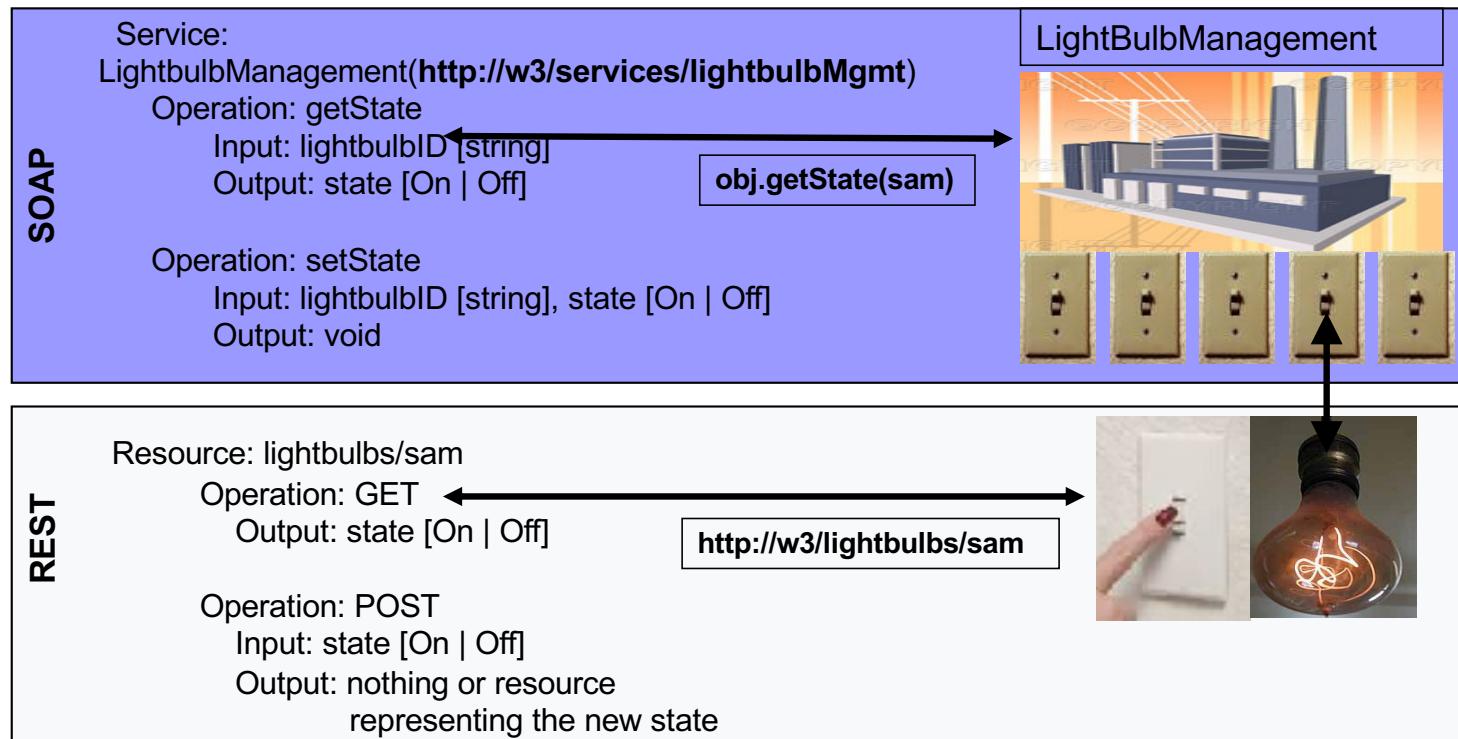


## Rest

- Simplicity
- Small packet size
- HTTP focused
- Easy to call from JavaScript

# SOAP vs. REST example

Is the light bulb currently on?



# SOAP vs. REST example data flows



## SOAP request

```
POST /services/LightBulbManager HTTP/1.1
Host: example.com
Content-Type: text/xml; charset=UTF-8
SOAPAction: "LightBulbManager#getState"
```

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="...">
<env:Body>
  <ns1:getState xmlns:ns1="...">
    <in0 xsi:type="xsd:string">SAM</in0>
  </ns1:add>
</SOAPenv:Body>
</env:Envelope>
```

## REST request

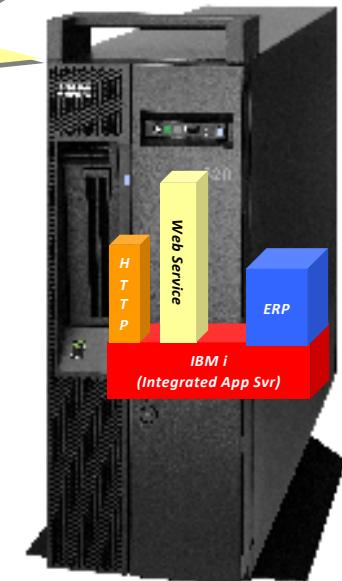
```
GET http://w3/lightbulbs/SAM HTTP/1.1
Host: example.com
Accept: application/xml
```

# IBM i Integrated Web Services Environment



*IBM i: Integrated Web Services Server **SOAP & REST***

A screenshot of the IBM Web Administration for i interface. The title bar says "IBM Web Administration for i". The menu bar has "Setup", "Manage" (which is selected), "Advanced", and "Related Links". The sub-menu "All Servers | HTTP Servers | Application Servers" is shown. A dropdown menu "Server" shows "INTAPPSVR2 - V7.1 (int app svr)". The left sidebar has sections like "Common Tasks and Wizards" (Create Web Services Server, Create HTTP Server, Create Application Server, Create WebSphere Portal), "Application Server Wizards" (Create Database Connection, Install New Application), "Server Properties" (Properties, Server Tracing, View HTTP Servers), "Applications" (Manage Installed Applications), "Resource Configuration" (Manage Database Connections), "Web Performance" (Web Performance Advisor), and "Problem Determination" (View Logs, View Create Summary). The main content area is titled "Manage Integrated Web Application Server" for Server INTAPPSVR2. It contains a grid of puzzle pieces representing application components and a "Manage Installed Applications" section with icons for Eclipse, ERP, Hello, and Sample.

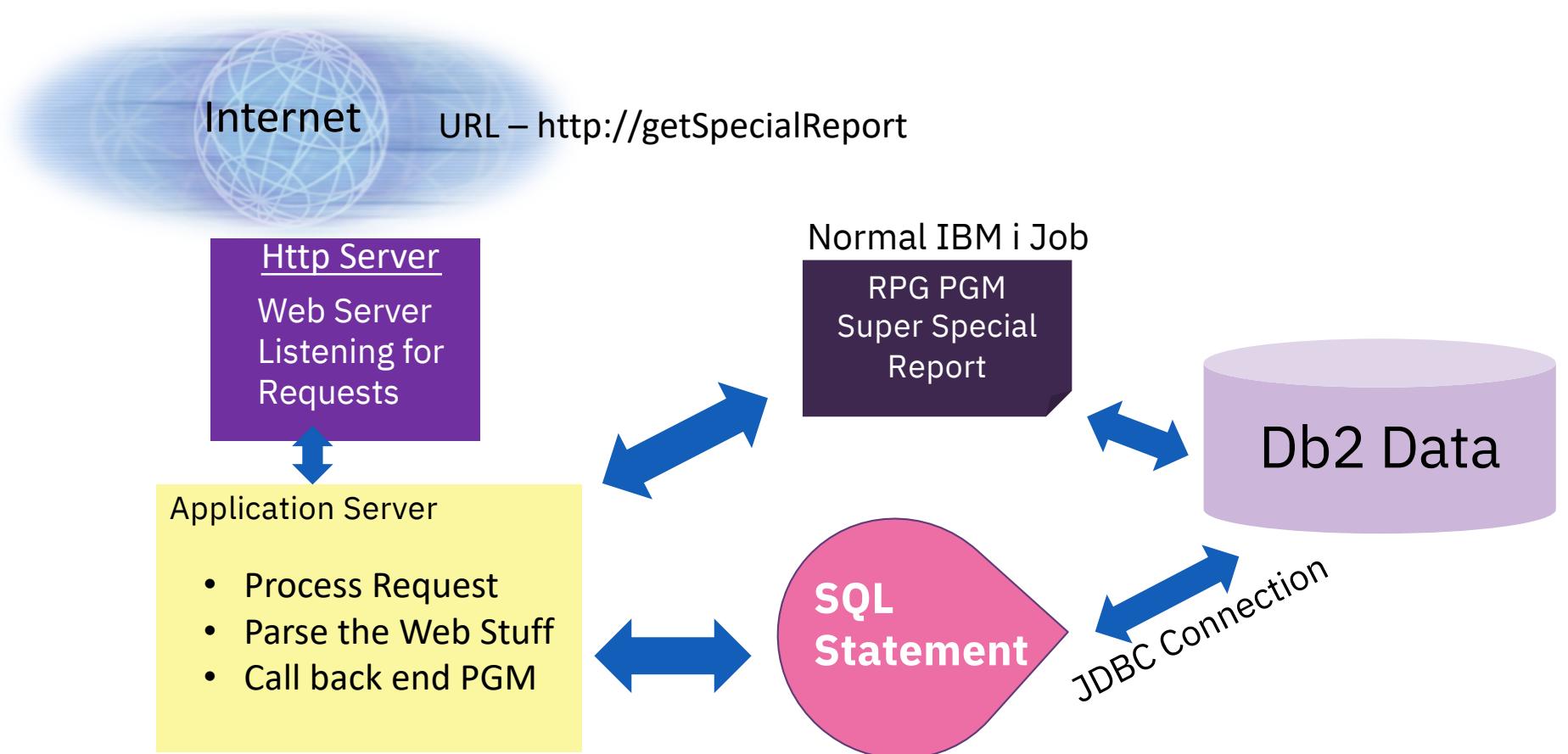




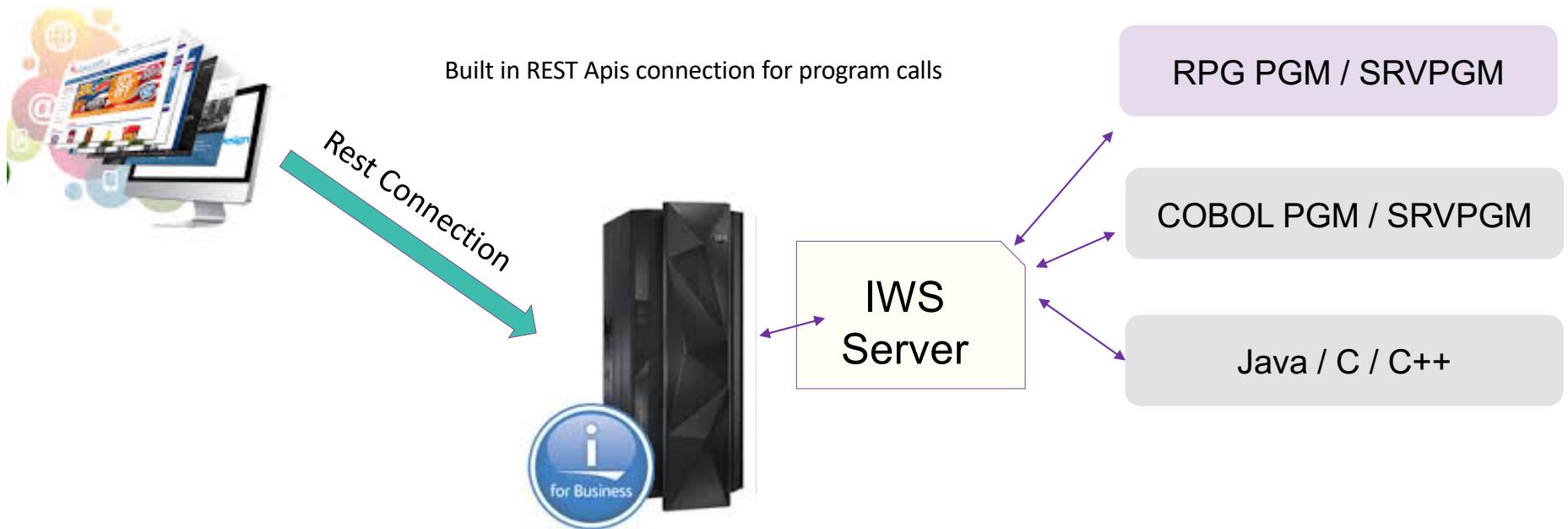
## About Integrated Web Services

- Released December of **2007** on IBM i 5.4, 6.1, and 7.1
  - Installed as part of base operating system option 3
  - Always load latest HTTP Group PTF for latest fixes and enhancements
- Consists of two separate entities
  - Integrated web services client for ILE
  - Integrated web services server
- Latest information, including product prerequisites, can be found at  
<http://www.ibm.com/systems/i/software/iws/>
- Continues to be re-invented and enhanced on 7.2 & 7.3 & 7.4

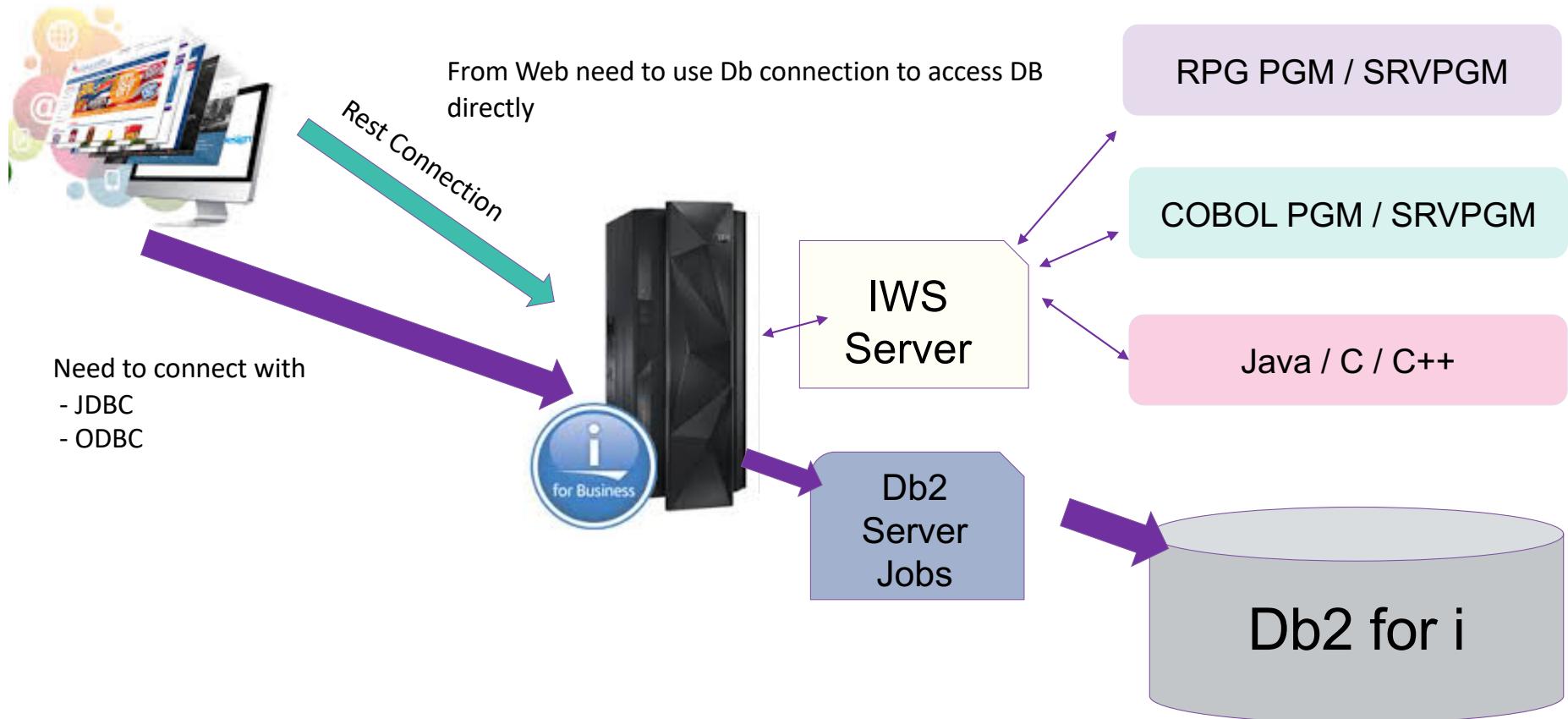
# Logistics



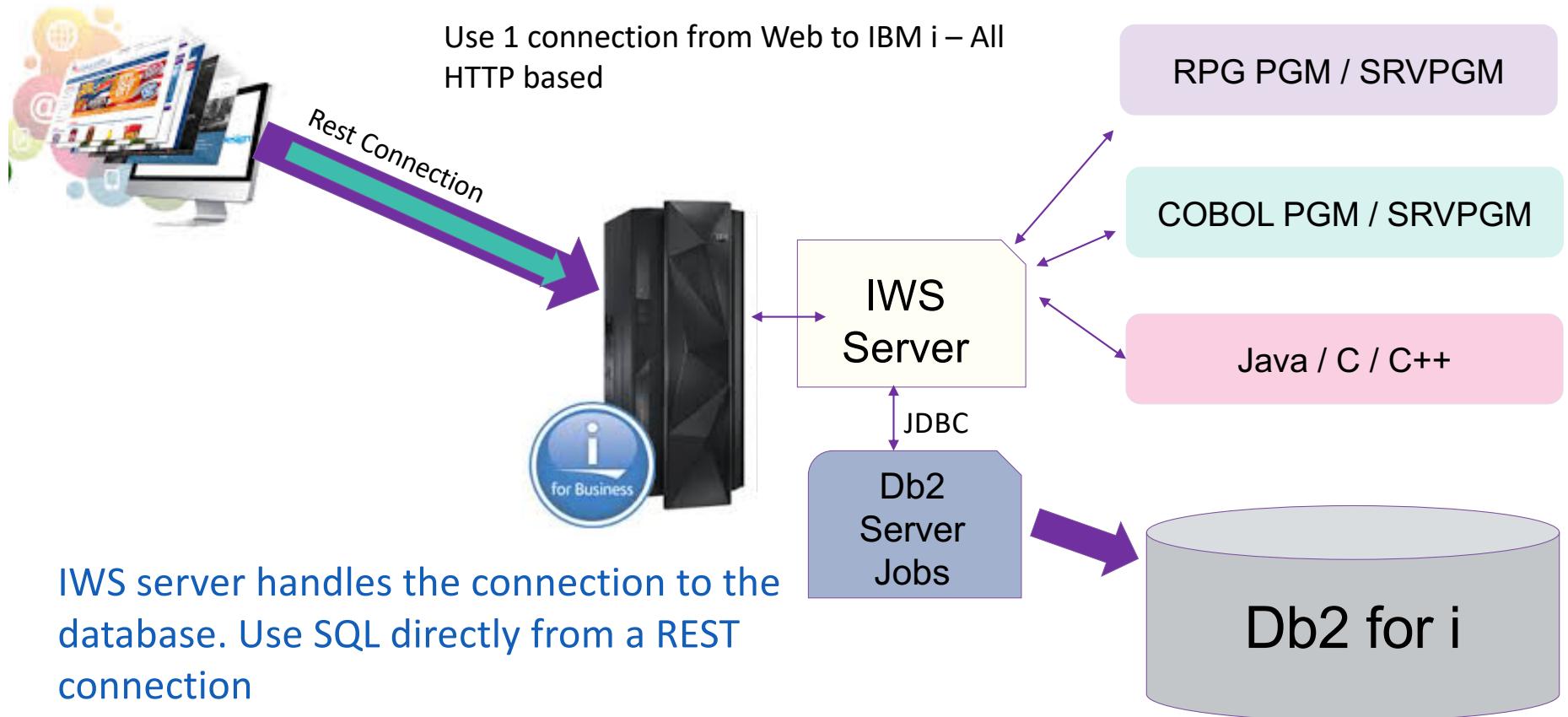
# Rest Apis - Today



# Connect to the Database - Today

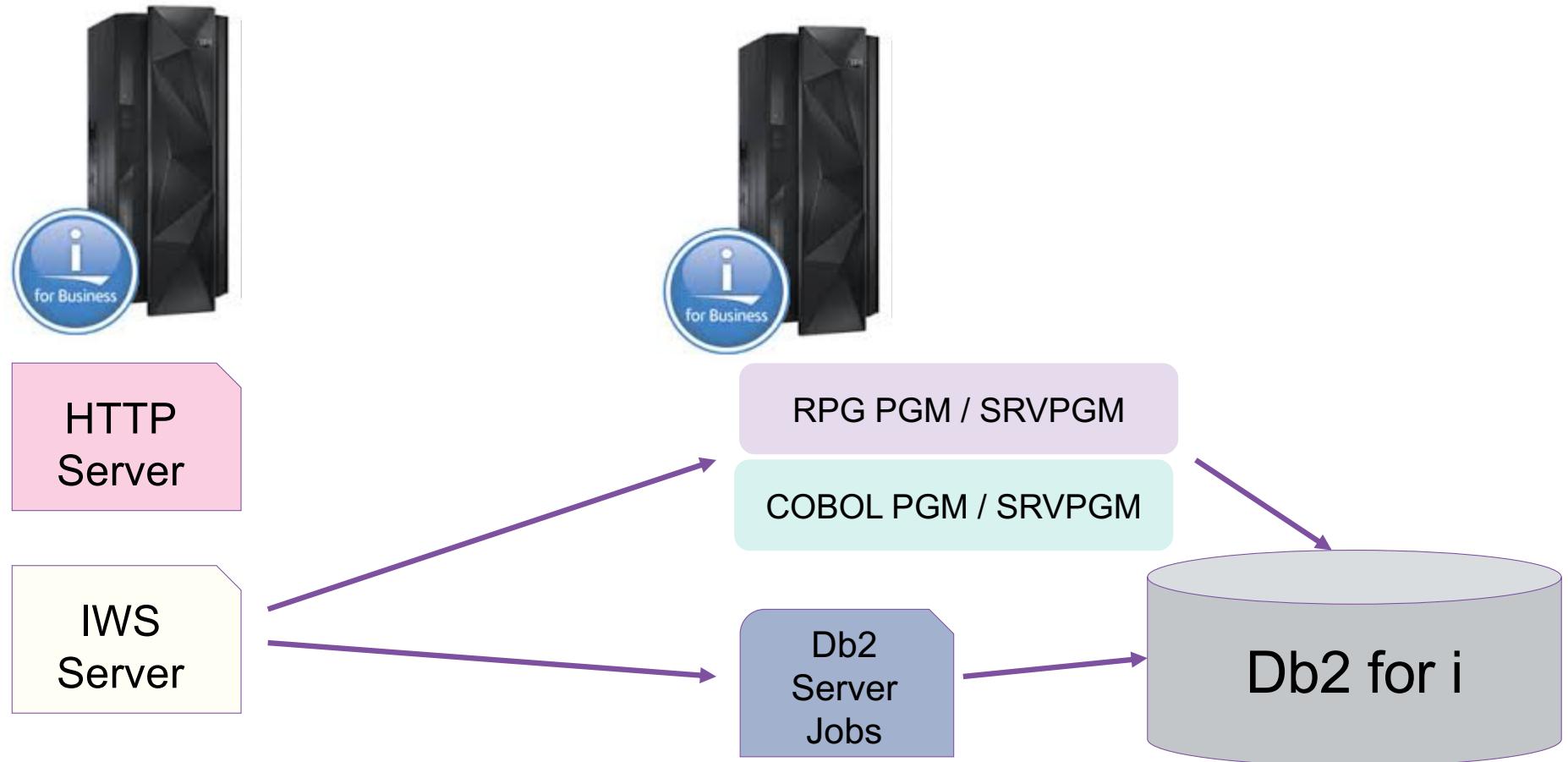


# Rest Apis – New SQL Access Using REST





# Multi – Tier





## What are all the parts...

- HTTP Apache Server
  - Connector to the IAS server
- IAS Server (Liberty)
  - JAX-RS (REST)
  - JAX-WS (SOAP)
  - Java program
    - Handles Inputs
    - Calls the backend ILE Program
    - Converts Output back to Web format





# About integrated web services server REST support

- Uses JAX-RS
  - Java API for RESTful Web Services
- Two ways to deploy a REST service
  - IBM Web Administration GUI updated
    - Deploying a REST service will require more user input than when deploying a SOAP service
  - QShell script installWebService.sh updated to support REST
  - Qshell Scripts for virtually all functions within IWS, everything can be programmatically configured



## What to get Prepared Ahead of Time

- Identify the RPG / Cobol / Java pgm
- Identify the SQL statements to use
- Figure out the HTTP methods
  - GET - read activities
  - POST - create entries
  - PUT - update an entry
  - DELETE – remove
- Determine the URI - Identifiers
  - Use Nouns vs Verbs
  - Keep it simple
- What incoming content types need to be supported
- What type of data is to be returned



# Student Registration Management



Create APIs using SQL to do basic application function

- Register new students
- Edit registered student information
- List registered students
- Get information about a student
- Remove student registrations



Thanks Nadir!

## Getting Started



# Planning

- Specify media types (e.g. XML, JSON, etc.) the procedure will accept
- Specify media types the procedure will return
- Optionally specify what values to inject in procedure input parameters
  - Path segment (e.g. /accounts/{**id**})
  - Matrix parameters (e.g. /cars;**color=blue**)
  - Query parameters (e.g. /cars?b**color=blue**)
  - Form data
  - HTTP headers
  - HTTP Cookies
- Optionally designate response code and HTTP header output parameters

# HTTP Methods and URI Mappings



| HTTP Method | URI                         | Description                      |
|-------------|-----------------------------|----------------------------------|
| GET         | /context-root/students      | Return all student registrations |
| GET         | /context-root/students/{id} | Return student registration      |
| POST        | /context-root/students      | Register a new student           |
| PUT         | /context-root/students      | Update registered student        |
| DELETE      | /context-root/students/{id} | Remove registered student        |



# Lets Define the SQL

| URI                         | Procedure identifier | SQL statement   |
|-----------------------------|----------------------|---|
| /context-root/students      | GETALL               | SELECT * from STUDENTDB   |
| /context-root/students/{id} | GETBYID              | SELECT * from STUDENTDB<br>WHERE "studentID" = ?  |
| /context-root/students      | ADD                  | INSERT INTO STUDENTDB<br>("studentID", "firstName", "lastName", "gender")<br>VALUES(?, ?, ?, ?)   |
| /context-root/students      | UPDATE               | UPDATE STUDENTDB SET<br>"firstName" = ?, "lastName" = ?,<br>"gender" = ?<br>WHERE "studentID" = ? |
| /context-root/students/{id} | REMOVE               | DELETE FROM STUDENTDB<br>WHERE "studentID" = ?  |



# Lets Setup the Database

## Open Run SQL Scripts in ACS

### Create the Library

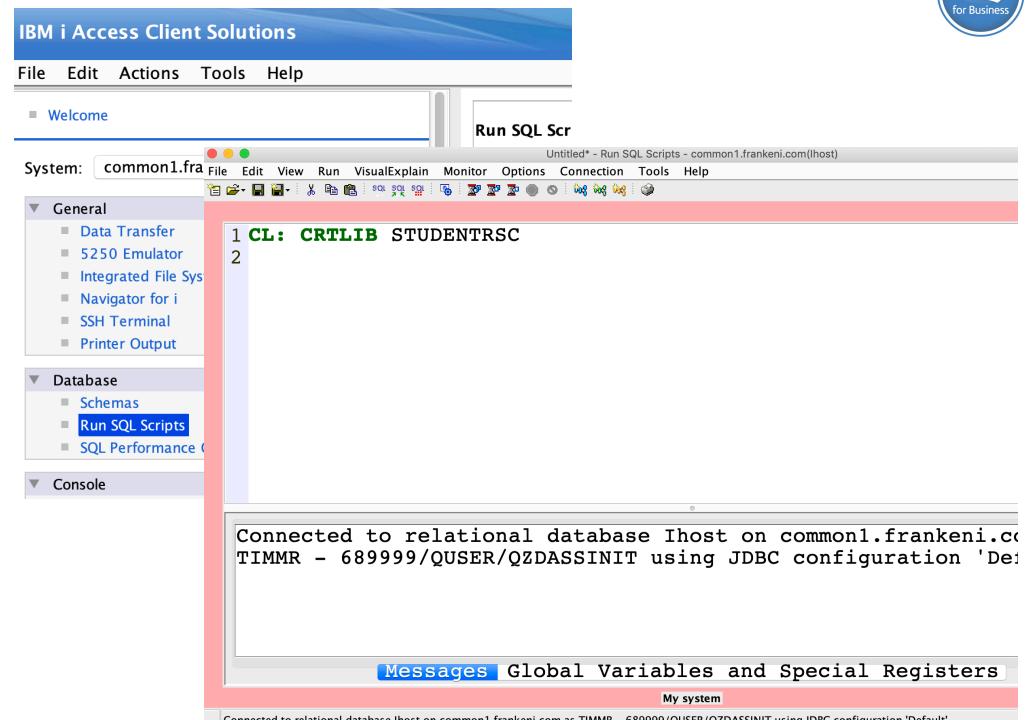
```
CL: CRTLIB STUDENTRSC
```

### Create the Database

```
CREATE TABLE STUDENTRSC/STUDENTDB
("studentID"  CHAR (9) NOT NULL,
 "firstName"   CHAR (50) NOT NULL,
 "lastName"    CHAR (50) NOT NULL,
 "gender"      CHAR (10) NOT NULL,
 PRIMARY KEY ("studentID"))
```

### Populate the Database

```
INSERT INTO STUDENTRSC/STUDENTDB
("studentID", "firstName", "lastName", "gender")
VALUES ('823M934LA', 'Nadir', 'Amra', 'Male'),
       ('826M660CF', 'John', 'Doe', 'Male'),
       ('747F023ZX', 'Jane', 'Amra', 'Female')
```



# Setup



In order to insure the service works, need to authorize the user profile for the service to the Db2 table

```
CL: CHGAUT OBJ('/qsys.lib/studentrsc.lib/studentdb.file')
    USER(QWSERVICE) DTAAUT(*RWX)
```

A screenshot of the IBM i Navigator interface. The title bar reads "Untitled\* - Run SQL Scripts - common1.frankeni.com(lhost)". The menu bar includes File, Edit, View, Run, VisualExplain, Monitor, Options, Connection, Tools, and Help. The toolbar below has icons for file operations like Open, Save, Print, and Database. The main area is a script editor with a pink background. It contains four numbered lines of CL code:

```
1 CL: CHGAUT OBJ('/qsys.lib/studentrsc.lib/studentdb.file')
2     USER(QWSERVICE) DTAAUT(*RWX)
3
4
```



# Web Integration Permissions

In the past, any user wanting to use Web Admin they were required to have \*ALLOBJ and \*IOSYSCFG special authority!



System Security policy just does not allow this!



## 'Permissions' Support

- Now a **\*USER** granted 'permission' can use the GUI
- Group profiles are now supported





# Web Integration Permissions

## — Developers can use Web Admin

- No longer need \*ALLOBJ special authority
- Administrators can grant users ‘Permission’
- Empowering the User
- Group Profile support

## — Two Permissions Available

- Operator – Start & Stop servers
- Developer – All functions

*Integrated GUI interface now available to Developers and Operators without compromising your system security*

The screenshot shows the 'Manage Permissions' page in the IBM Web Administration for i interface. The top navigation bar includes tabs for 'Setup', 'Manage', 'Advanced' (which is selected), and 'Related Links'. Below the navigation is a breadcrumb trail: 'Settings | Internet Users and Groups | Permissions'. The main content area is titled 'Manage Permissions' and contains a descriptive text about managing and creating Web-related servers. It features two tabs: 'By User' (selected) and 'By Server'. A table lists permissions for two users: LDHTEST and MDSLOW. The table columns are 'User ID', 'Server Type/Name', and 'Role'. The data is as follows:

| User ID | Server Type/Name   | Role      |
|---------|--|-----------|
| LDHTEST | Integrated Web Application Server<br>Create server - Enabled<br>*ALL   | Developer |
|         | Integrated Web Services Server   |           |
|         | WebSphere Application Server<br>Create server - Enabled<br>*ALL  | Developer |
|         | IBM HTTP Server<br>Create server - Enabled<br>*ALL   | Developer |
| MDSLOW  | Integrated Web Application Server<br>Integrated Web Services Server<br>WebSphere Application Server<br>IBM HTTP Server |           |

At the bottom of the table are buttons for 'Add', 'Modify', 'Remove', 'Expand All', 'Collapse All', and 'Refresh'.

IBMi 7.1 & Higher



# Create web services server

Access Web Admin <http://hostname:2001/HTTPAdmin>

The screenshot shows the 'IBM Web Administration for i' interface. In the top left, there's a navigation bar with 'Setup' selected, followed by 'Manage | Advanced | Related Links'. To the right are 'WebSphere' and 'IBM' logos. On the left, a sidebar titled 'Common Tasks and Wizards' lists four items: 'Create Web Services Server', 'Create HTTP Server', 'Create Application Server', and 'Create WebSphere Portal'. A red arrow points from the text above to the 'Create Web Services Server' link. The main content area is titled 'IBM Web Administration for i' and contains a 'Getting started' message: 'Create and learn about the servers needed to run your Web content.' Below this, three links are listed with icons: 'Create a New Web Services Server' (with a cloud icon), 'Create a New HTTP Server' (with a globe icon), and 'Create a New Application Server' (with a server icon). A red arrow points from the text above to the 'Create a New Web Services Server' link.

Click on the Create New Web Services Server link

# How do you test things ?



The screenshot shows the SoapUI homepage. At the top left is the SoapUI logo with 'by SMARTBEAR'. Top navigation links include 'Why SoapUI', 'Blog', 'Download', and 'Getting Started'. A banner at the top says 'SoapUI users unite at SmartBear Connect! Learn more' with a link icon. The main content area has a green background with the text 'Build Better | Test Smarter' and 'The Most Advanced REST & SOAP Testing Tool in the World'. A button labeled 'Download SoapUI NG Pro' is visible. The central part of the page features a construction-themed illustration with a crane, a large 'S' logo, and various icons related to software development and testing.

Free Download

<https://www.soapui.org>



# Deployment demo



The screenshot shows the 'Deploy New Service' wizard in the IBM Web Administration interface. The left sidebar contains navigation links for Common Tasks and Wizards, Web Services Wizards, Server Properties, Services, and Problem Determination. The main panel displays the first step of the wizard, titled 'Specify Web service type - Step 1 of 8'. It includes a description of what a Web service is, a 'Specify Web service type:' dropdown set to 'REST', and a 'Specify Web service implementation:' dropdown set to '\*SQL'. A large blue arrow points from the text 'This panel has been updated so one can indicate whether web service is based on SQL statements or ILE programs/service programs' towards the '\*SQL' dropdown.

## Notes:

1. This panel has been updated so one can indicate whether web service is based on SQL statements or ILE programs/service programs



# Create the Web Services Server Container

## Create Web Services Server

Specify Web services server name - Step 1 of 4

Welcome to the Create Web Services Server wizard. A Web services server provides a secure and easy way to configure environment for hosting Web services that are based on IBM i objects such as RPG and COBOL programs and SQL statements. This wizard creates everything needed to run Web services.

For more information, please visit: <http://www.ibm.com/support/docview.wss?uid=isg3T1026868>

### Specify a unique name for this server

Server name:

Server description:

Create HTTP server

**Back**

**Next**

**Cancel**



# Create the Web Services Server Container

## Create Web Services Server

*Specify network attributes for server - Step 2 of 4*

Your server may listen for requests on specific IP addresses or on all IP addresses of the system. A comma separates multiple IP addresses. You can also specify a port for the server.

### Specify internet addresses and ports for server ?

Specify server command port:

10259

Specify internet address and port for the server

IP address:

All IP addresses ▾

Port:

10258

Specify internet address and port for the HTTP server

IP address:

All IP addresses ▾

Port:

10268

**Back**

**Next**

**Cancel**



# Create the Web Services Server Container

## Create Web Services Server

### Specify User ID for Server - Step 3 of 4

The server requires an IBM i user ID to run the server's jobs. It is recommended that a special user ID is used for the server's jobs since this user ID is given authority to all of the server's objects, such as files and directories.

Specify user ID for this server: 

Use default user ID

**Note:** The default server user ID is QWSERVICE.

- Specify an **existing** user ID
- Create a **new** user ID

---

**Back**

**Next**

**Cancel**



# Create the Web Services Server Container

## Create Web Services Server

Summary - Step 4 of 4

**Servers**   **Service**

### Web Services Server Information

**Server name:** Virtual

**Server description:** Web services server created by the Create Web Services Server wizard.

**Port:** 10258

**Command port:** 10259

**Server root:** /www/Virtual

**Server URL:** http://common1.frankeni.com:10268

**User ID for server:** QWSERVICE

**Context root:** /web

### HTTP Server Information

**HTTP server name:** VIRTUAL

**HTTP server description:** Web services server created by the Create Web Services Server wizard.

**Port:** 10268

**Document root:** /www/Virtual/htdocs

**Server root:** /www/Virtual

**Server association:** Virtual

**Back**

**Finish**

**Cancel**



# Create the Web Services Server Container

IBM Web Administration for i

Setup **Manage** Advanced | Related Links

All Servers | HTTP Servers **Application Servers** Installations

Creating Server: Virtual - V2.6 (web services)

**Common Tasks and Wizards**

- [Create Web Services Server](#)
- [Create HTTP Server](#)
- [Create Application Server](#)

[Virtual](#)

## Manage Web Services Server

Server: **Virtual**

Web services server created by the Create Web Services Server wizard.

The IBM integrated Web services server provides a secure and easy way to configure an environment for hosting Web objects such as RPG and COBOL programs and Db2 files. An easy to use interface for managing the server in addition to other services is provided.

For more information, please visit: <http://www.ibm.com/support/docview.wss?uid=isg3T1026868>

---

Server "Virtual" is in the process of being created. To update the status, click the **Refresh** icon above.

---

**Note:** To update the status, click Refresh



# Container Created

All Servers | HTTP Servers **Application Servers** Installations

Running Server: Virtual - V2.6 (web services)

Common Tasks and Wizards  
 Create Web Services Server  
 Create HTTP Server  
 Create Application Server

Web Services Wizards  
 Deploy New Service  
 Configure SSL  
 Disable SSL

Server Properties  
 Properties  
 Server Tracing  
 View HTTP Servers

Services  
 Manage Deployed Services

Problem Determination  
 View Logs  
 Web Log Monitor  
 View Create Summary

Virtual

## Manage Web Services Server

Server: **Virtual**

Web services server created by the Create Web Services Server wizard.

The IBM integrated Web services server provides a secure and easy way to configure an environment for Web services that are based on IBM i objects such as RPG and COBOL programs and Db2 files. An extensive set of tools for managing the server in addition to installing and managing Web services is provided.

For more information, please visit: <http://www.ibm.com/support/docview.wss?uid=isg3T1026868>

**Manage Deployed Services**

Server: "Virtual"  
 ConvertTemp

Note: To update the status, click



# Deploy a SQL Based Service

[Virtual](#) > Manage Deployed Services

## Manage Deployed Services

Data current as of Jul 15, 2019 7:08:17 AM.

Deployed services: [?](#)

|  | Service name | Status  | Type | Startup type | Service definition        |
|--|--------------|---------|------|--------------|---------------------------|
|  | ConvertTemp  | Running | SOAP | Automatic    | <a href="#">View WSDL</a> |

[Deploy](#)

[Refresh](#)

[Close](#)



# Deploy a SQL Based Service

[Virtual](#) > [Manage Deployed Services](#) > Deploy New Service

## Deploy New Service

Specify Web service type - Step 1 of 9

Welcome to the Deploy New Service wizard. This wizard helps you create Web services using IBM i objects and data. A Web service is a self-contained software component with a well-defined interface that describes a set of operations accessible over the Internet and exchange XML messages that are based on the SOAP protocol. A REST-based Web service exposes resources, where client requests are handled by resource methods and the format of messages that are exchanged is defined by the resource itself.

Specify Web service type:

REST ▾

Specify Web service implementation ✓ \*PGM/\*SRVPGM  
\*SQL



Back

**Next**

Cancel



# Deploy a SQL Based Service

Virtual > [Manage Deployed Services](#) > Deploy New Service

## Deploy New Service

Specify Name for Service - Step 2 of 8

The Web service to be externalized is a resource. The URI path template identifies matching patterns for the path is relative to the context root and can be a simple string or one or more template parameters that expressions to further restrict what is allowed.



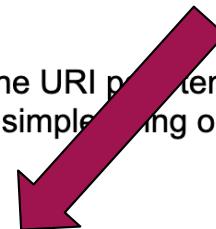
Resource name:

Service description:

URI path template:  e.g. /temperature, /temperature/{temp:\d+}

Specify a meaningful resource name,  
this service will be referenced now as

*/context-root/students*



[Back](#) [Next](#) [Cancel](#)



# Deploy a SQL Based Service

[Virtual](#) > [Manage Deployed Services](#) > Deploy New Service

## Deploy New Service

Specify Database Properties - Step 3 of 8

Specify database properties that will be used to process SQL statements.

Database system: localhost

or...

Default schema: studentrsc

or...

Naming convention: \*SQL

\*LIBL

Is the database Local or Remote ? ie, the database can be on a different IBM i

Specify the default schema for these Rest APIs. Only this Schema is used by default.

Do you want to use SQL based naming ?  
*schema.table*

Do you want to use System based naming ?  
*schema/table*

Specify the library list that is to be used to resolve unqualified stores procedure names or unqualified names

Back

Next

Cancel



# Deploy a SQL Based Service - Specify the SQL

Specify SQL statements that will be externalized as a Web service: [?](#)

| Procedure name                             | SQL statement/Parameter name                |
|--|---|
| <input checked="" type="checkbox"/> REMOVE | Delete From STUDENTDB Where "studentID" = ? |

Enter the Procedure name and the SQL statement. Notice, I have a parameter specified

Once you click 'Continue' the wizard will detect the parameter and you will have the opportunity to then view and modify

Add    Remove    Remove All    Continue    Insert from example

Back    Next    Cancel



# Deploy a SQL Based Service - Specify the SQL

## Deploy New Service

Specify SQL Statements - Step 4 of 8

Specify SQL statements that will be externalized as a Web service: 

|                          | Procedure name | SQL statement/Parameter name                | Usage | Data type |
|--------------------------|----------------|---|-------|-----------|
| <input type="checkbox"/> | REMOVE         | Delete From STUDENTDB Where "studentID" = ? |       |           |
|                          |                | PARM00001                                   | input | CHAR      |

**Add** **Remove All**

Click on the entry to 'open' it back up for update to modify the parameter name from the default value to something customized



# Deploy a SQL Based Service - Specify the SQL

Specify SQL statements that will be externalized as a Web service: [?](#)

| Procedure name                             | SQL statement/Parameter name                | Usage                        | Data type  |
|--|---|------------------------------|------------|
| <input checked="" type="checkbox"/> REMOVE | Delete From STUDENTDB Where "studentID" = ? | Updated to correct value for | input CHAR |

A red arrow points from the text "studentID" in the "Parameter name" column to the input field where "studentID" is typed.



# Deploy a SQL Based Service - Specify the SQL

Specify SQL statements that will be externalized as a Web service: [?](#)

|                          | Procedure name | SQL statement/Parameter name  | Usage | Data type |
|--------------------------|----------------|---|-------|-----------|
| <input type="checkbox"/> | REMOVE         | Delete From STUDENTDB Where "studentID" = ?   |       |           |
|                          |                | studentID   | input | CHAR      |
| <input type="checkbox"/> | UPDATE         | UPDATE STUDENTDB SET<br>"firstName" = ?, "lastName" = ?,<br>"gender" = ?<br>WHERE "studentID" = ? |       |           |
|                          |                | PARM00001   | input | CHAR      |
|                          |                | PARM00002   | input | CHAR      |
|                          |                | PARM00003   | input | CHAR      |
|                          |                | PARM00004   | input | CHAR      |
| <input type="checkbox"/> | ADD            | INSERT INTO STUDENTDB<br>("studentID", "firstName", "lastName", "gender")<br>VALUES(?,?,?,?,?)    |       |           |
|                          |                | PARM00001   | input | CHAR      |
|                          |                | PARM00002   | input | CHAR      |
|                          |                | PARM00003   | input | CHAR      |
|                          |                | PARM00004   | input | CHAR      |
| <input type="checkbox"/> | GETBYID        | SELECT * from STUDENTDB<br>WHERE "studentID" = ?  |       |           |
|                          |                | studentID   | input | CHAR      |
| <input type="checkbox"/> | GETALL         | SELECT * from STUDENTDB   |       |           |

[Add](#) [Remove All](#)



# Deploy a SQL Based Service – SQL Information

## Deploy New Service

### Specify SQL Information - Step 5 of 8

Customize how each procedure will process SQL statements. For query statements, this includes the type of result sets that may be returned and how pagination is handled for result sets.: [?](#)

|                                    |   |  |
|------------------------------------|---|--|
| Procedure name:                    | REMOVE                                      | Set any SQL warnings as an error condition |
| SQL Statement:                     | Delete From STUDENTDB Where "studentID" = ? |  |
| SQL result type:                   | Multi-row result set                        |  |
| Trim mode for output fields:       | Trailing                                    |  |
| SQL state information in response: | On errors                                   |  |
| Treat warnings as SQL Errors:      | Yes   |  |
| User-defined error message:        | <input type="text"/>                        |  |
| HTTP status code on SQL success:   | 204   | or... <a href="#">?</a>                    |
| HTTP status code on SQL failure:   | 500   | or... <a href="#">?</a>                    |

Set any SQL warnings as an error condition

The SQL code 204 (no content) , if that code is produced, then treat as a success

[Back](#) [Next](#) [Cancel](#)



# Deploy a SQL Based Service – SQL Information

## Deploy New Service

### Specify SQL Information - Step 5 of 8

Customize how each procedure will process SQL statements. For query statements, this includes the type of pagination is handled for result sets.: [?](#)

Procedure name: UPDATE  
SQL Statement:  
UPDATE STUDENTDB SET  
"firstName" = ?, "lastName" = ?, "gender" = ?  
WHERE "studentID" = ?

SQL result type: Multi-row result set

Trim mode for output fields: Trailing

SQL state information in response: On errors

Treat warnings as SQL Errors: Yes

User-defined error message:

HTTP status code on SQL success: 204

or 1

HTTP status code on SQL failure: 500

or...

The SQL code 204 (no content) , if that code is produced, then treat as a success



# Deploy a SQL Based Service – SQL Information

## Deploy New Service

### Specify SQL Information - Step 5 of 8

Customize how each procedure will process SQL statements. For query statements, this includes the type of pagination is handled for result sets.: [?](#)

Procedure name: ADD

SQL Statement:  
INSERT INTO STUDENTDB  
("studentID", "firstName", "lastName", "gender")  
VALUES(?, ?, ?, ?)

SQL result type: Multi-row result set

Trim mode for output fields: Trailing

SQL state information in response: On errors

Treat warnings as SQL Errors: Yes

User-defined error message:

HTTP status code on SQL success: 201 or... 1

HTTP status code on SQL failure: 500 or...

The SQL code 201 (Created) , if that code is produced, then treat as a success



# Deploy a SQL Based Service – SQL Information

## Deploy New Service

### Specify SQL Information - Step 5 of 8

Customize how each procedure will process SQL statements. For query statements, this includes the type of result set returned, how results are paginated, and how trailing blanks are handled.

Procedure name:

GETBYID

SQL Statement:

```
SELECT * from STUDENTDB  
WHERE "studentID" = ?
```

SQL result type:

Single-row result set

1

Return only a single entry. This will ensure the results are not an array of objects

Trim mode for output fields:

Trailing

2

Trim trailing blanks. This improves performance and only returns the actual data

SQL state information in response:

On errors

Treat warnings as SQL Errors:

Yes

User-defined error message:

HTTP status code on SQL success:

200

or...

3

The SQL code 200 (OK) , if that code is produced, then treat as a success

HTTP status code on SQL failure:

500

or...



# Deploy a SQL Based Service – Resource Information

## Deploy New Service

### Specify Resource Method Information - Step 6 of 8

Procedures are mapped to resource methods. Each resource method needs to be defined to handle client requests. Map the HTTP request method to a resource method.: [?](#)

Procedure name: REMOVE

URI path template for resource: /

HTTP request method:  1

URI path template for method: /{id} 2 or... 3

HTTP header information:

Allowed input media types: \*ALL or... 4

Returned output media types: \*JSON 3

Whether to wrap input parameters:

- Wrap input parameters
- Do not wrap input parameters

Input parameter mappings:

| Parameter name | Data type | Input source  | Identifier | Default Value |
|----------------|-----------|---------------|------------|---------------|
| studentID      | CHAR      | *PATH_PARAM 4 | id         | *NONE or...   |

Specify the HTTP Method

URI path template, since we have to pass in a parameter, need to define that

Return only JSON

Specify the input source as a Path Parameter



# Deploy a SQL Based Service – Resource Information

## Deploy New Service

### Specify Resource Method Information - Step 6 of 8

Procedures are mapped to resource methods. Each resource method needs to be defined to handle client requests. Map an HTTP request method to a resource method.: [?](#)

Procedure name: UPDATE

URI path template for resource: /

HTTP request method:  1

URI path template for method: \*NONE  2

HTTP header information: 3

Allowed input media types: \*JSON 2  4

Returned output media types: \*JSON 3

Whether to wrap input parameters:

Wrap input parameters 4

Do not wrap input parameters

Specify the HTTP Method

Send and Return only JSON

Since the request will be in the payload of the client request, we specify that the parameters should be wrapped



# Deploy a SQL Based Service – Resource Information

## Deploy New Service

### Specify Resource Method Information - Step 6 of 8

Procedures are mapped to resource methods. Each resource method needs to be defined to handle an HTTP request method to a resource method.: [?](#)

Procedure name: ADD

URI path template for resource: /

HTTP request method:

POST 1

URI path template for method:

\*NONE or...

HTTP header information:

Allowed input media types:

\*JSON 2 or...

Returned output media types:

\*JSON 3

Whether to wrap input parameters:

Wrap input parameters 4

Do not wrap input parameters

Specify the HTTP Method

Send and Return only JSON

Since the request will be in the payload of the client request, we specify that the parameters should be wrapped



# Deploy a SQL Based Service – Resource Information

## Deploy New Service

### Specify Resource Method Information - Step 6 of 8

Procedures are mapped to resource methods. Each resource method needs to be defined to handle client requests. Map the HTTP request method to a resource method.: [?](#)

| Procedure name:                   | GETBYID  | Specify the HTTP Method  |            |                |
|-----------------------------------|--|--|------------|----------------|
| URI path template for resource:   | /  | URI path template, since we have to pass in a parameter, need to define that |            |                |
| HTTP request method:              | GET<br>1   |  |            |                |
| URI path template for method:     | /{id}<br>2   |  |            |                |
| HTTP header information:          |  |  |            |                |
| Allowed input media types:        | *ALL<br>3  | Send and Return only JSON  |            |                |
| Returned output media types:      | *JSON<br>4   |  |            |                |
| Whether to wrap input parameters: | <input type="radio"/> Wrap input parameters<br><input checked="" type="radio"/> Do not wrap input parameters | Specify the input source as a Path Parameter                                 |            |                |
| Input parameter mappings:         |  |  |            |                |
| Parameter name                    | Data type  | Input source   | Identifier | Default Value  |
| studentID                         | CHAR   | *PATH_PARAM<br>5   | id         | *NONE<br>or... |



# Deploy a SQL Based Service – Resource Information

## Deploy New Service

Specify Resource Method Information - Step 6 of 8

Procedures are mapped to resource methods. Each resource method needs to be defined to handle client requests. Map an HTTP request method to a resource method.: ?

Procedure name: GETALL

URI path template for resource: /

HTTP request method:

GET

1

Specify the HTTP Method

URI path template for method:

\*NONE

or...

or...

HTTP header information:

or...

or...

Default Input of \*ALL is fine, and Return only JSON

Allowed input media types:

\*ALL

2

or...

or...

Returned output media types:

\*JSON

3



# Deploy a SQL Based Service – User Profile

## Deploy New Service

### *Specify User ID for this Service - Step 7 of 8*

The service requires an IBM i user ID to run the Web service business logic. The user ID must have the resources that the Web service requires.

Specify User ID for this Service: [?](#)

- Use **server's** user ID
- Specify an **existing** user ID
- Use **authenticated** user ID

User Profile for the Service. Can be different than the profile for the server. This profile must have the necessary authority to access the data in the database in order to function correctly.



# Deploy a SQL Based Service – Create Service

## Deploy New Service

Summary - Step 8 of 8

When you click **Finish** the web service is deployed.

**Service**   **JDBC Properties**   **Methods**

**Resource name:** students

**Resource description:** SQL

**Service install path :** /www/wservice/webservices/services/students

**URI path template:** /

**User ID for service:** \*SERVER (QWSERVICE)

**Back**

**Finish**

**Cancel**



# Service Created and Running

Running Server: Virtual - V2.6 (web services)

Common Tasks and Wizards

- Create Web Services Server
- Create HTTP Server
- Create Application Server

Web Services Wizards

- Deploy New Service
- Configure SSL
- Disable SSL

Server Properties

- Properties
- Server Tracing
- View HTTP Servers

Services

- Manage Deployed Services

[Virtual > Manage Deployed Services](#)

## Manage Deployed Services

Data current as of Jul 15, 2019 11:13:31 AM.

Deployed services:

|                                  | Service name | Status               | Type | Startup type | Service definition |
|----------------------------------|--------------|----------------------|------|--------------|--------------------|
| <input type="radio"/>            | ConvertTemp  | <span>Running</span> | SOAP | Automatic    | View WSDL          |
| <input checked="" type="radio"/> | students     | <span>Running</span> | REST | Automatic    | View Swagger       |

Deploy Stop Properties Uninstall Redeploy Refresh

Properties to change JDBC setting, User Profile, Edit Swagger Doc, and more

Updates to the service can be made using Re-deploy

View the Swagger document, this describes the details of the service



# Test the service

A screenshot of a web browser window. The address bar shows the URL: "common1.frankeni.com:10268/web/services/students/". The browser interface includes standard navigation buttons (back, forward, refresh, home), a toolbar with links like "Most Visited" and "Getting Started", and tabs for "JSON", "Raw Data", and "Headers". Below the tabs are buttons for "Save" and "Copy".

The JSON response is as follows:

## ▼ students\_GETALL\_R:

### ▼ 0:

```
studentID:      "823M934LA"  
firstName:      "Nadir"  
lastName:       "Amra"  
gender:         "Male"
```

The properties page and the Swagger doc have the URL to call. The default with no parameters specified returns all entries in the database table

### ▼ 1:

```
studentID:      "826M660CF"  
firstName:      "John"  
lastName:       "Doe"  
gender:         "Male"
```

### ▼ 2:

```
studentID:      "747F023ZX"  
firstName:      "Jane"  
lastName:       "Amra"  
gender:         "Female"
```

# Test the service



SoapUI 5.4.0

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Search Forum

Request 1

Method: POST Endpoint: http://common1.frankeni.com:10268 Resource: /web/services/students/

HTTP/1.1 201 Created  
Date: Mon, 15 Jul 2019 15:15:18 GMT  
Server: Apache  
X-Powered-By: IBM i  
Content-Length: 0  
Connection: close  
Content-Type: application/json  
Content-Language: en-US

Add a new entry to the database

Raw Request JSON XML HTML

Media Type: application/json

```
{  
    "PARM00001": "213F45678",  
    "PARM00002": "Megan",  
    "PARM00003": "Rowe",  
    "PARM00004": "Female"  
}
```

Create the input data source for the new student



# Test the service

Screenshot of a web browser displaying a JSON response from a service endpoint. The URL is common1.frankeni.com:10268/web/services/students/. The response shows five entries under the key 'students\_GETALL\_R:'.

```
studentID: "823M934LA"
firstName: "Nadir"
lastName: "Amra"
gender: "Male"

studentID: "826M660CF"
firstName: "John"
lastName: "Doe"
gender: "Male"

studentID: "747F023ZX"
firstName: "Jane"
lastName: "Amra"
gender: "Female"

studentID: "123M45678"
firstName: "Tim"
lastName: "Rowe"
gender: "Male"

studentID: "213F45678"
firstName: "Megan"
lastName: "Rowe"
gender: "Female"
```

The Multiple new entries added to the database



# Demo of REST based Methodology



## What have we done lately....

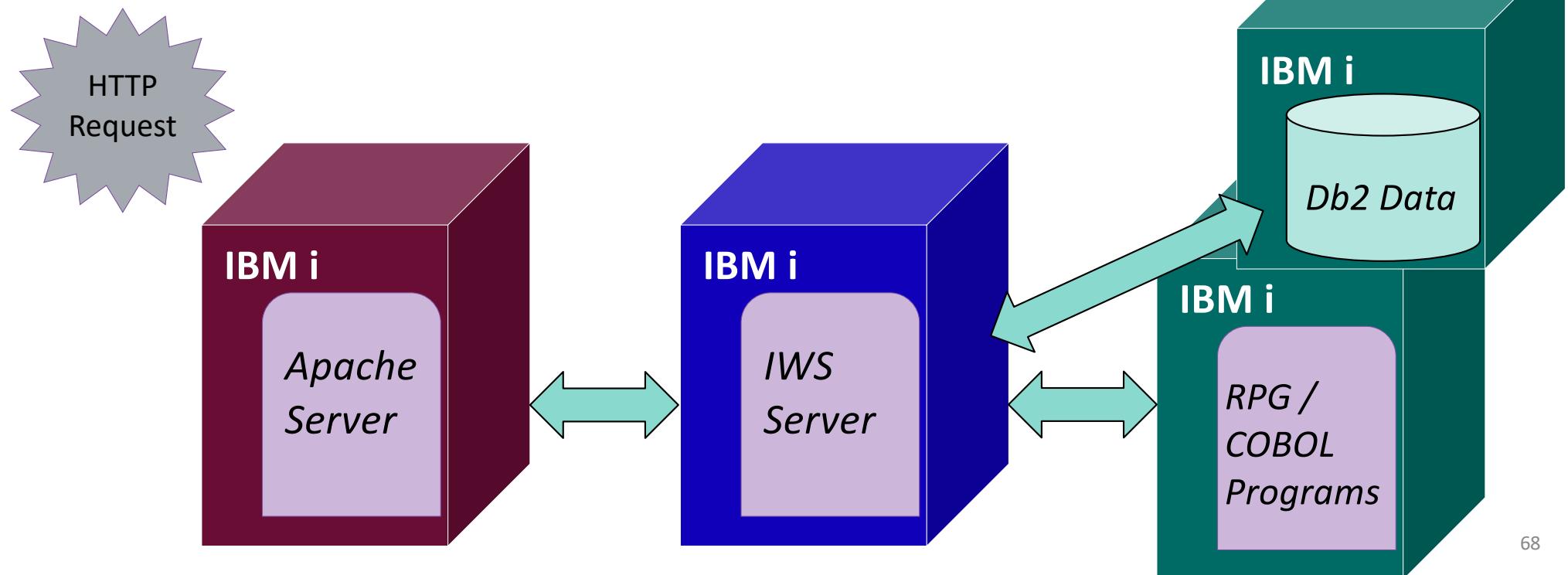
- 3 node support
    - HTTP on one node
    - Application Server on a node
    - Backend RPG on a node
  - Use Authenticated User
  - Services re-deploy
  - Connection pool pre-initialization
  - Variable length fields
  
  - Many other updates as requested by the community
- <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/dW%20IBM%20Integrated%20Web%20Services%20for%20i>



# IWS Multi Node Server Support

## Multi nodes

- Apache, IWS, and backend programs ALL on different IBM i nodes



# Some Additional Light Reading



Developer Works – 3 Part Series on Rest for IBM i

- [https://www.ibm.com/developerworks\(ibmi\)/library/i-rest-web-services-server1/](https://www.ibm.com/developerworks(ibmi)/library/i-rest-web-services-server1/)
- [http://www.ibm.com/developerworks\(ibmi\)/library/i-rest-web-services-server2/](http://www.ibm.com/developerworks(ibmi)/library/i-rest-web-services-server2/)
- [https://www.ibm.com/developerworks\(ibmi\)/library/i-rest-web-services-server3/](https://www.ibm.com/developerworks(ibmi)/library/i-rest-web-services-server3/)

# Questions and Answers



# For More Information:



| Some Links You Need  | Twitter   | #Hashtags  |
|--|---|--|
| <p>IBM i Home Page:<br/><a href="http://www.ibm.com/systems/i">www.ibm.com/systems/i</a></p> <p>IBM 30<sup>th</sup> Anniversary<br/><a href="http://ibmi30.mybluemix.net/">http://ibmi30.mybluemix.net/</a></p> <p>IBM Systems Magazine IBM i Edition:<br/><a href="http://ibmsystemsmag.com/ibmi/">http://ibmsystemsmag.com/ibmi/</a></p> <p>Support Life Cycle:<br/><a href="https://www-01.ibm.com/software/support/ibmi/lifecycle/">https://www-01.ibm.com/software/support/ibmi/lifecycle/</a></p> <p>License Topics:<br/><a href="https://www-01.ibm.com/support/docview.wss?uid=nas8N1022087">https://www-01.ibm.com/support/docview.wss?uid=nas8N1022087</a></p> |  Twitter<br><br><a href="https://twitter.com/IBMSystems">@IBMSystems</a><br><a href="https://twitter.com/COMMONug">@COMMONug</a><br><a href="https://twitter.com/IBMChampions">@IBMChampions</a><br><a href="https://twitter.com/IBMSystemsISVs">@IBMSystemsISVs</a><br><a href="https://twitter.com/IBMiMag">@IBMiMag</a><br><a href="https://twitter.com/ITJungleNews">@ITJungleNews</a><br><a href="https://twitter.com/SAPonIBMi">@SAPonIBMi</a><br><a href="https://twitter.com/SiDforIBMi">@SiDforIBMi</a> | #IBMi30<br>#PowerSystems<br>#IBMi<br>#IBMAIX<br>#POWER8<br>#LinuxonPower<br>#OpenPOWER<br>#HANAonPower<br>#ITInfrastructure<br>#OpenSource<br>#HybridCloud<br>#BigData |

# For More Information:



| Blogs  |   |
|--|---|
| <p><b>IBM Blogs:</b></p> <p><a href="http://ibmsystemsmag.com/blogs/you-and-i/">IBM Systems Magazine You and i (Steve Will)</a><br/><a href="http://ibmsystemsmag.com/blogs/i-can/">IBM Systems Magazine i-Can (Dawn May)</a><br/><a href="http://ibmsystemsmag.com/blogs/open-your-i/">IBM Systems Magazine: Open your i (Jesse Gorzinski)</a><br/><a href="http://db2fori.blogspot.co.uk/">IBM DB2 for i (Mike Cain)</a><br/><a href="http://db2webqueryi.blogspot.co.uk/">IBM DB2 Web Query for i (Doug Mack)</a></p> <p><b>IBM Champion's Blogs:</b></p> <p><a href="http://ibmsystemsmag.com/blogs/idevelop/">IBM Systems Magazine: iDevelop (Jon Paris and Susan Gantner)</a><br/><a href="http://ibmsystemsmag.com/ibmi/trends/italk-with-tuohy/">IBM Systems Magazine: iTalk with Tuoy</a></p> | <p><a href="http://ibmsystemsmag.com/blogs/you-and-i/">http://ibmsystemsmag.com/blogs/you-and-i/</a><br/><a href="http://ibmsystemsmag.com/blogs/i-can/">http://ibmsystemsmag.com/blogs/i-can/</a><br/><a href="http://ibmsystemsmag.com/blogs/open-your-i/">http://ibmsystemsmag.com/blogs/open-your-i/</a><br/><a href="http://db2fori.blogspot.co.uk/">http://db2fori.blogspot.co.uk/</a><br/><a href="http://db2webqueryi.blogspot.co.uk/">http://db2webqueryi.blogspot.co.uk/</a></p> <p><a href="http://ibmsystemsmag.com/blogs/idevelop/">http://ibmsystemsmag.com/blogs/idevelop/</a><br/><a href="http://ibmsystemsmag.com/ibmi/trends/italk-with-tuohy/">http://ibmsystemsmag.com/ibmi/trends/italk-with-tuohy/</a></p> |



# Notices and disclaimers

- © 2019 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- **U.S. Government Users Restricted Rights – use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.**  
IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts.  
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”
- **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

# Notices and disclaimers continued



- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**
- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.
- IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)