



# REST APIs for Direct Db2 Access

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

Architect Application  
Development IBM i



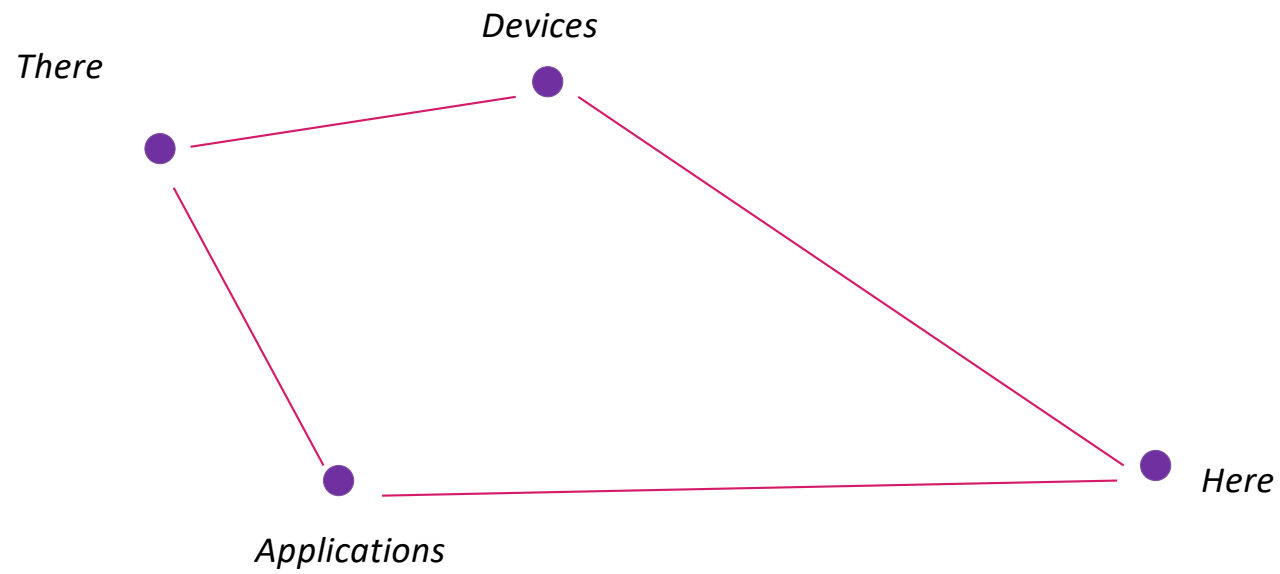


# What is an API - Agenda

- 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**pplication

**P**rogramming

**I**nterface

# 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

REpresntational

Sate



Transfer



# 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



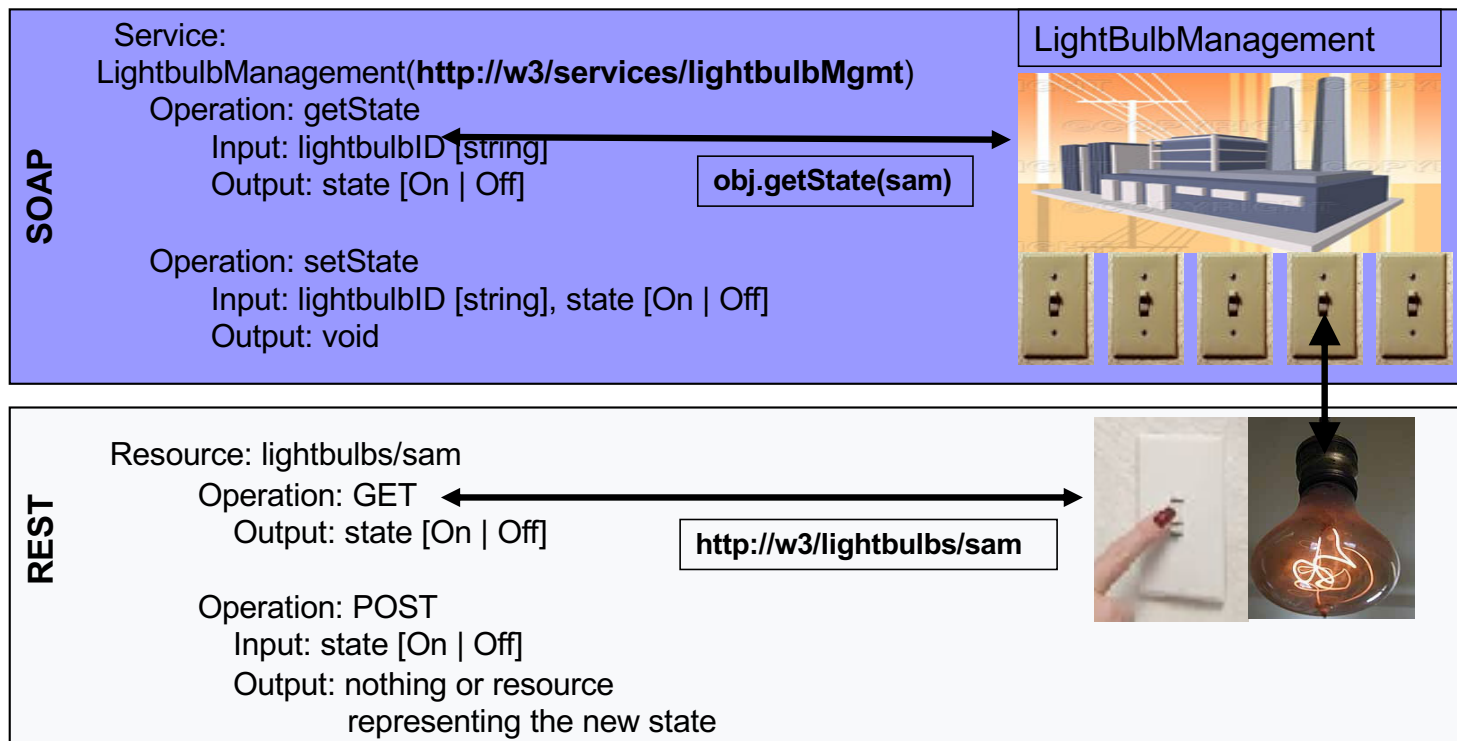
## Rest

- 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>
```

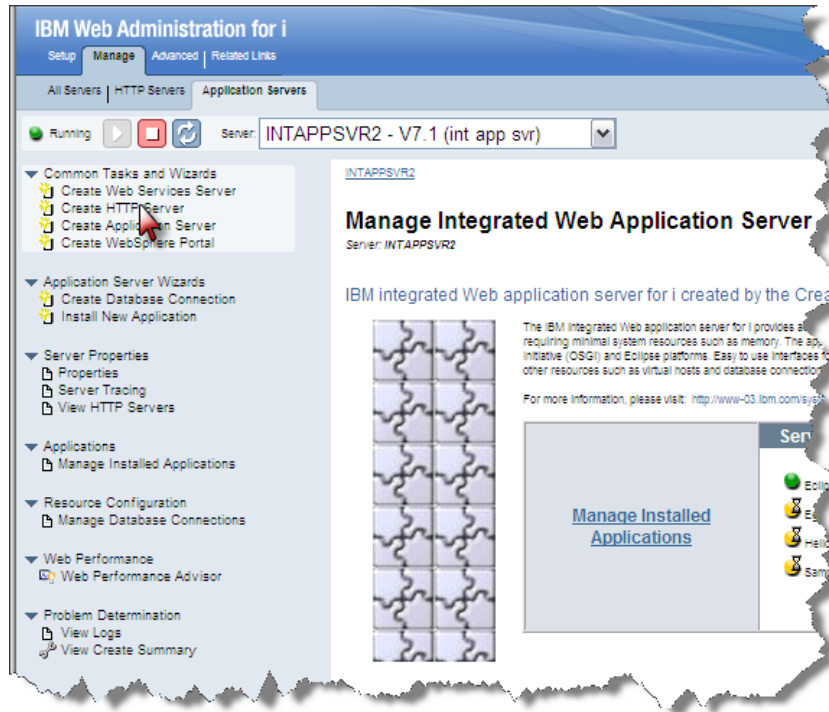
## REST request

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

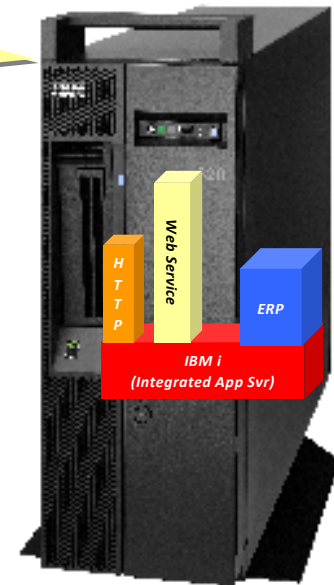
# IBM i Integrated Web Services Environment



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



**Included with  
IBM i**



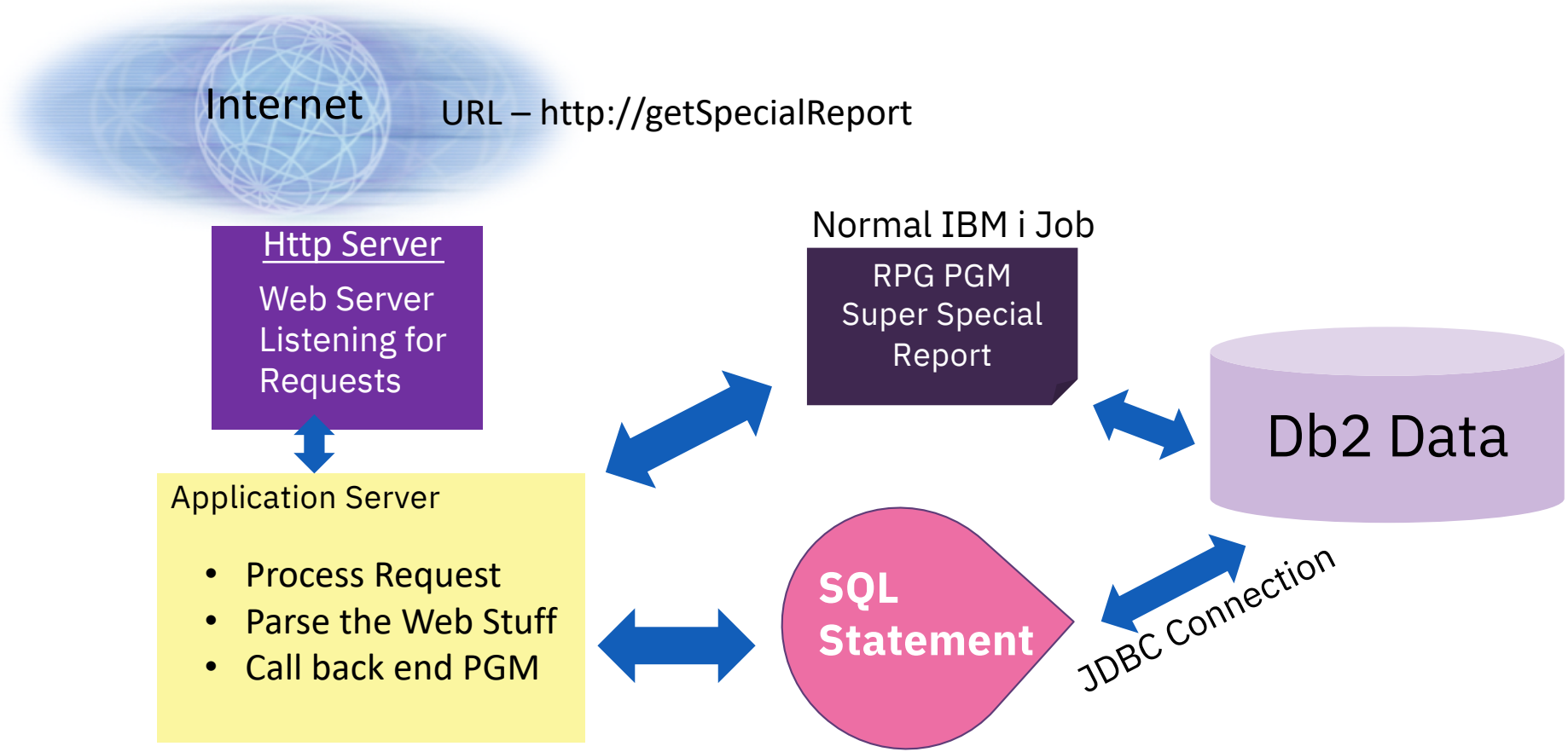




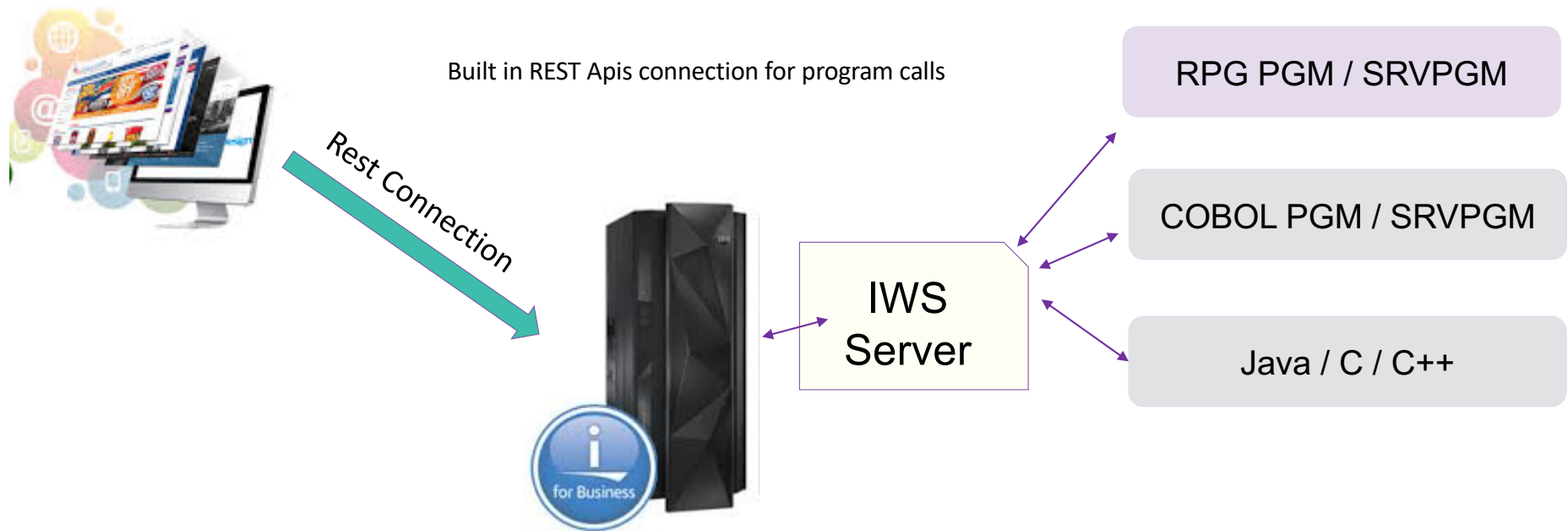
## About Integrated Web Services

- 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

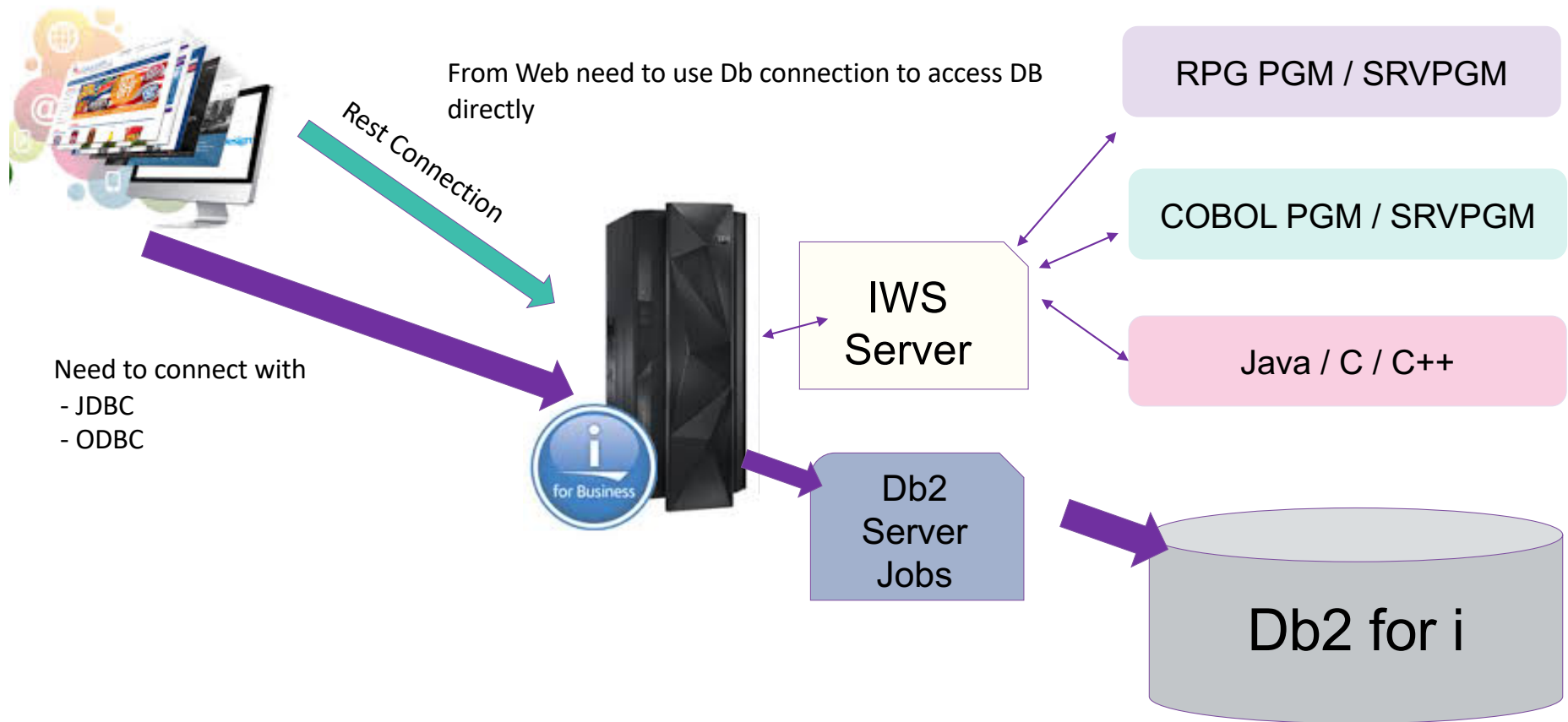
# Logistics



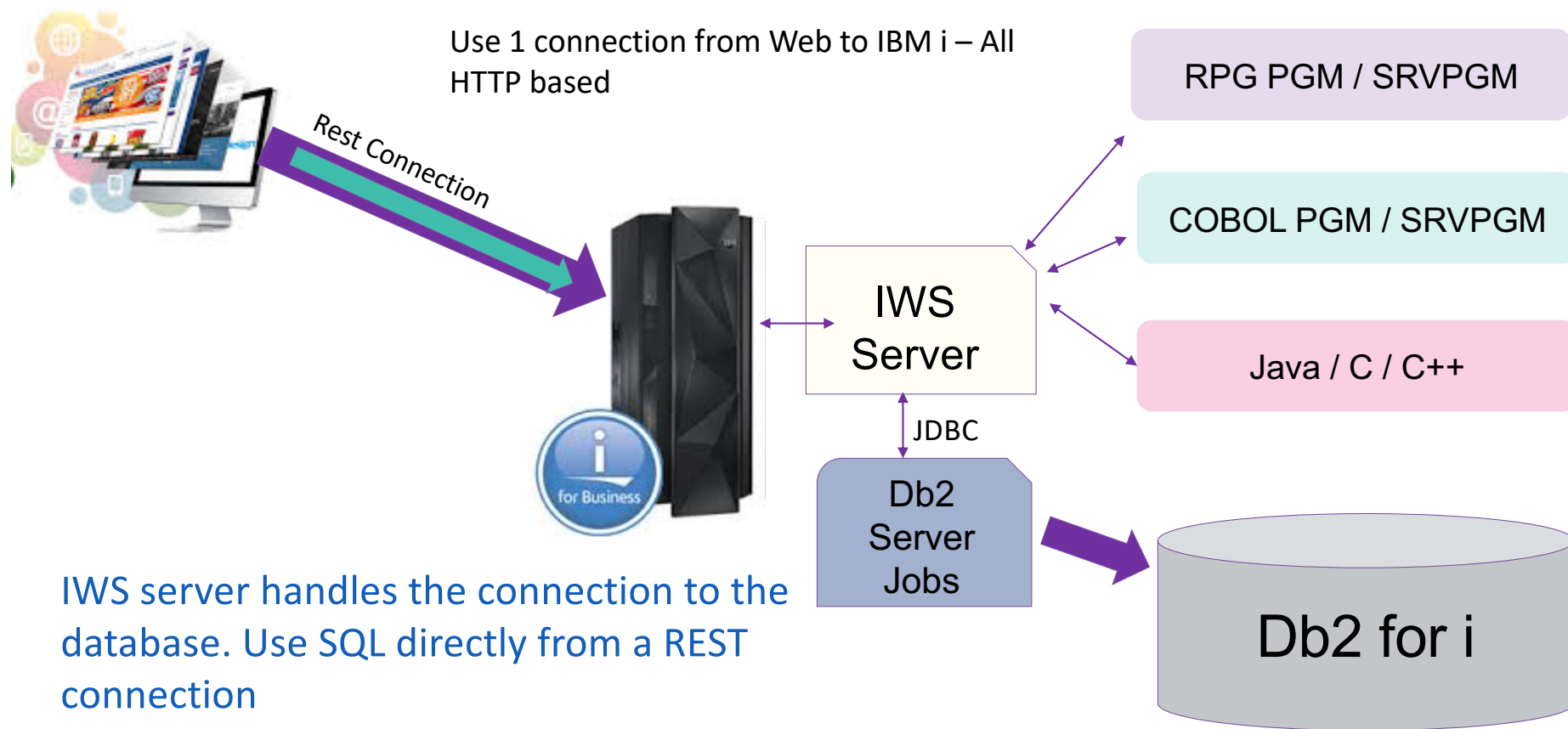
# Rest Apis - Today



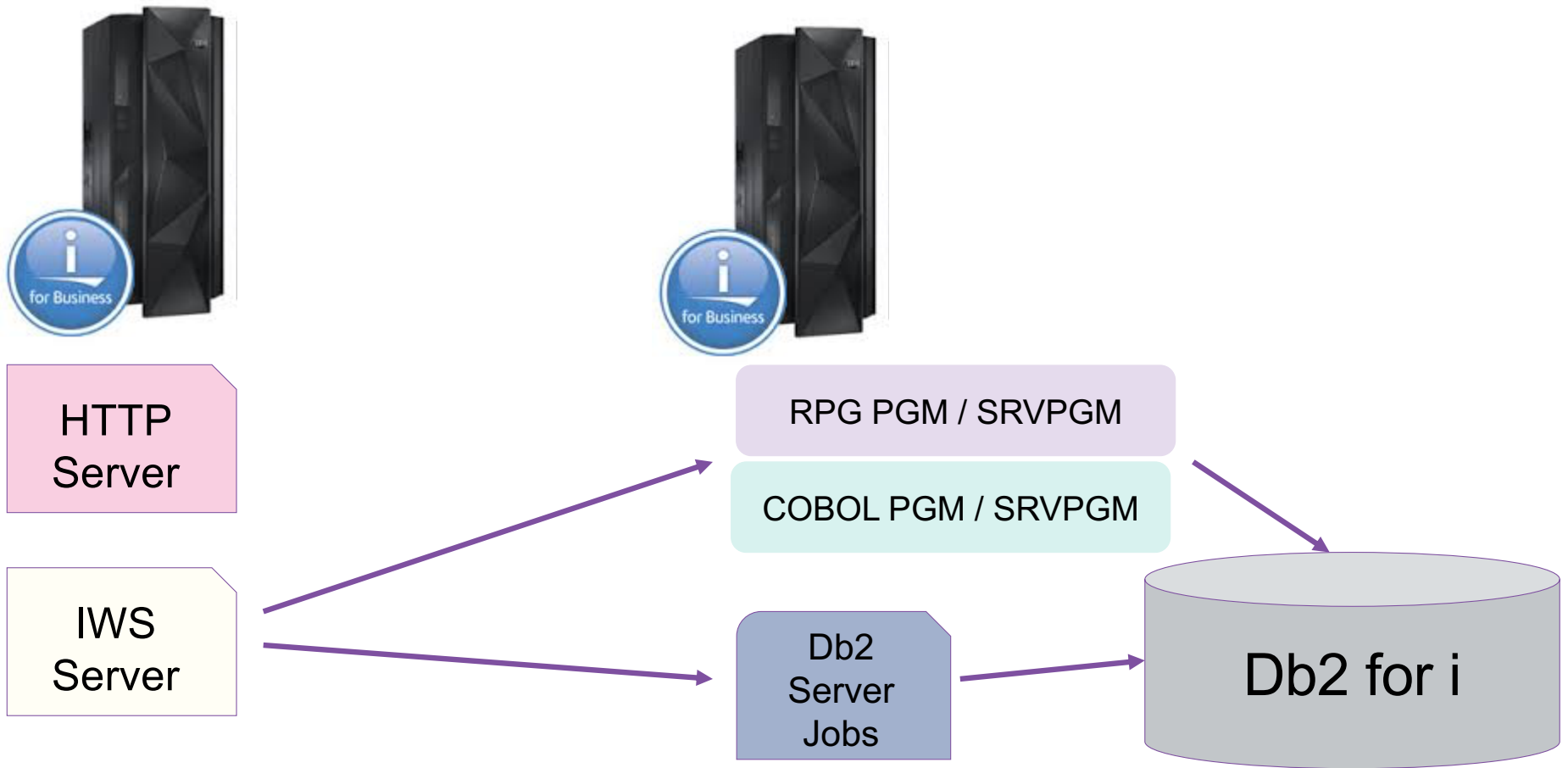
# Connect to the Database - Today



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

# Lets Setup the Database

## Open Run SQL Scripts in ACS

### Create the Library

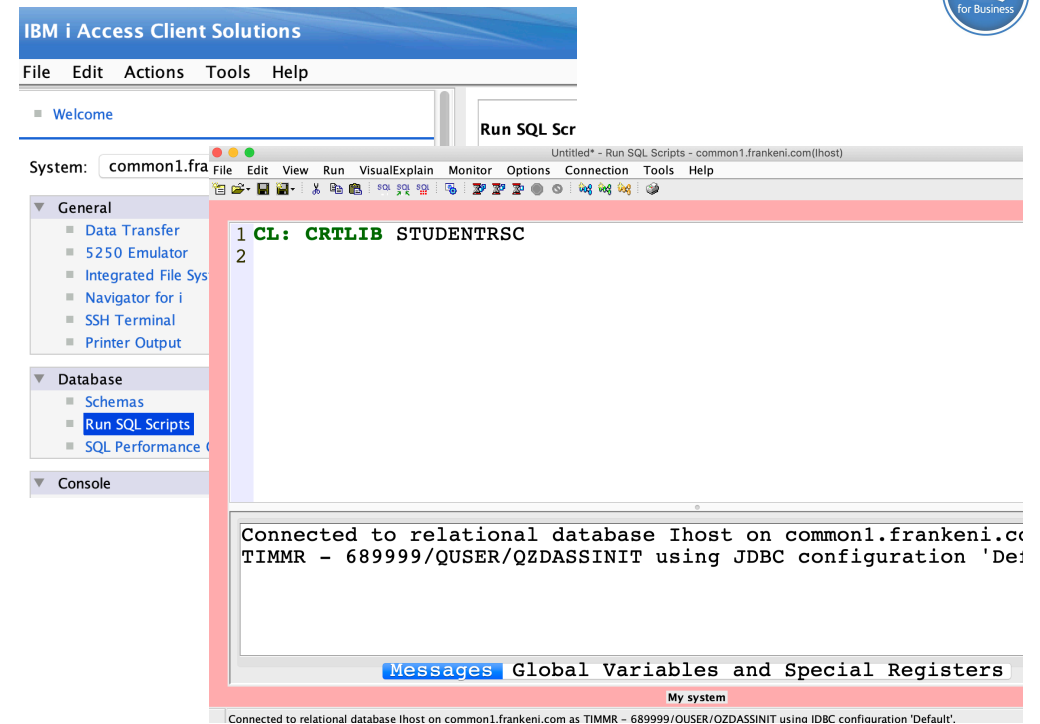
```
CL: CRTLIB STUDENTRSC
```

### 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 a SQL script editor window. 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 contains various icons for file operations and SQL execution. The main text area shows the following SQL command:

```
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!

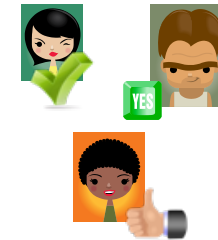


System Security policy just does not allow this!



## '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*

IBM Web Administration for i  
Setup | Manage | **Advanced** | Related Links  
Settings | Internet Users and Groups | **Permissions**

### Manage Permissions

By default, only users with \*ALLOBJ & \*IOSYSCFG special authorities can manage and create Web related servers on the system through the use of IBM Web Administration for i. The Web related server include IBM HTTP Server, WebSphere Application Server, Integrated Application Server and Integrated Web Services Server. If a user without the necessary IBM i special authorities is required to manage or create Web based servers, an administrator can grant that user permission to a server or group of servers.

By User | **By Server**

Data current as of Apr 22, 2010 8:19:16 AM.

User ID	Server Type/Name	Role
LDHTEST	Integrated Web Application Server	
	Create server - Enabled	
	*ALL	Developer
	Integrated Web Services Server	
MDSL0W	WebSphere Application Server	
	Create server - Enabled	
	*ALL	Developer
	IBM HTTP Server	
MDSL0W	Create server - Enabled	
	*ALL	Developer
	Integrated Web Application Server	
	Integrated Web Services Server	
MDSL0W	WebSphere Application Server	
	IBM HTTP Server	

Add | Modify | Remove | Expand All | Collapse All | Refresh

# IBM i 7.1 & Higher

# Create web services server



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

The screenshot shows the IBM Web Administration for i interface. The top navigation bar includes 'Setup', 'Manage', 'Advanced', and 'Related Links'. The left sidebar lists 'Common Tasks and Wizards' with sub-items: 'Create Web Services Server', 'Create HTTP Server', 'Create Application Server', and 'Create WebSphere Portal'. A red arrow points to the 'Create Web Services Server' link. The main content area is titled 'IBM Web Administration for i' and contains a 'Getting started' section with three links: 'Create a New Web Services Server', 'Create a New HTTP Server', and 'Create a New Application Server'. Each link has a small icon and an information icon.

**IBM Web Administration for i**

Getting started - Create and learn about the servers needed to run your Web content.

- Create a New Web Services Server**  
Create Web Services Server Wizard provides a convenient way to externalize existing programs running on IBM i, such as RPG or COBOL, as Web services. This allows Web service clients to interact with IBM i program based services from the Internet or intranet using Web service based industry standard communication protocols such as SOAP.
- Create a New HTTP Server**  
Create a new HTTP Server (powered by Apache) to run your HTTP Web content. This wizard will create everything you need to get started with simple Web serving.
- Create a New Application Server**  
Create a new application server to run dynamic Web applications. Create either an IBM integrated Web application server for i or a WebSphere Application Server.

Click on the Create New Web Services Server link

# How do you test things ?



The image shows a screenshot of the SoapUI website banner. At the top left is the SoapUI logo, which consists of a stylized 'S' inside a circle followed by the text 'SoapUI by SMARTBEAR'. To the right of the logo are navigation links: 'Why SoapUI' and 'Blog'. Below these are 'Download' and 'Getting Started' links. A dark green bar contains the text 'SoapUI users unite at SmartBear Connect! Learn more' with a right-pointing arrow. The main banner area has a green background with the text 'Build Better | Test Smarter' and 'The Most Advanced REST & SOAP Testing Tool in the World'. A white button with green text says 'Download SoapUI NG Pro'. The background features various icons: a crane lifting a large 'S' logo, a clock, a code symbol '</>', gears, a truck, and laboratory glassware.



Free Download

<https://www.soapui.org>

# Deployment demo



IBM Web Administration for i  
Setup **Manage** Advanced | Related Links

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

Running [stop] [refresh] Server: IWSDB2 - V2.6 (web services) ▾

IWSDB2 > [Manage Deployed Services](#) > Deploy New Service

### Deploy New Service

Specify Web service type - Step 1 of 8

Welcome to the Deploy New Service wizard. This wizard helps you create Web services using IBM i objects and data. A SOAP-based Web service is a self-contained software component with a well-defined interface that describes a set of operations that are 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: \*SQL ▾

**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 an 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 the server.

### Specify internet addresses and ports for server ?

Specify server command port:

Specify internet address and port for the server

IP address:

Port:

Specify internet address and port for the HTTP server

IP address:

Port:

**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 these 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 W objects such as RPG and COBOL programs and Db2 files. An easy to use interface for managing the server in addit 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 services that are based on IBM i objects such as RPG and COBOL programs and Db2 files. An environment 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>

<b>Manage Deployed Services</b>	Server: "Virtual"
	ConvertTemp

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

# 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 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

\*SQL



Back

Next

Cancel


# Deploy a SQL Based Service



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

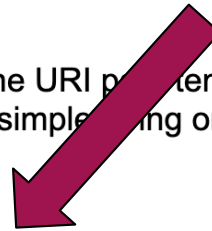
## Deploy New Service

*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. The path is relative to the context root and can be a simple string or one or more template parameters that use curly braces and expressions to further restrict what is allowed. 

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

*/context-root/students*



Resource name:

Service description:

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

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  
Library list: \*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

# Deploy a SQL Based Service - Specify the SQL



## Deploy New Service

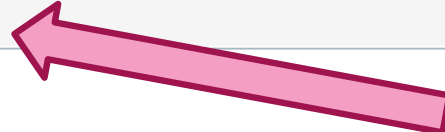
Specify SQL Statements - Step 4 of 8

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" = ?		
		<div style="border: 1px solid black; padding: 5px; display: inline-block;">Updated to correct value for</div>		
		<input type="text" value="studentID"/>	input	CHAR

# 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 "firstName" = ?, "lastName" = ?, "gender" = ? WHERE "studentID" = ?		
		PARM00001	input	CHAR
		PARM00002	input	CHAR
		PARM00003	input	CHAR
		PARM00004	input	CHAR
<input type="checkbox"/>	ADD	INSERT INTO STUDENTDB ("studentID", "firstName", "lastName", "gender") VALUES(?,?,?,?)		
		PARM00001	input	CHAR
		PARM00002	input	CHAR
		PARM00003	input	CHAR
		PARM00004	input	CHAR
<input type="checkbox"/>	GETBYID	SELECT * from STUDENTDB 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
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... ▾
HTTP status code on SQL failure:	500 or... ▾

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

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

1

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...
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 pagination is handled for result sets. ?

Procedure name:	GETBYID
SQL Statement:	SELECT * from STUDENTDB WHERE "studentID" = ?
SQL result type:	Single-row result set <b>1</b>
Trim mode for output fields:	Trailing <b>2</b>
SQL state information in response:	On errors
Treat warnings as SQL Errors:	Yes
User-defined error message:	
HTTP status code on SQL success:	200 <b>3</b>
HTTP status code on SQL failure:	500

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

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

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

# 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. Specify the HTTP request method to a resource method. ?

Procedure name: REMOVE

URI path template for resource: /

HTTP request method: DELETE

URI path template for method: /{id}

HTTP header information:

Allowed input media types: \*ALL

Returned output media types: \*JSON

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	id	*NONE

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. Specify the HTTP request method to a resource method. ?

Procedure name: UPDATE

URI path template for resource: /

HTTP request method: PUT **1**

URI path template for method: \*NONE or...

HTTP header information:

Allowed input media types: \*JSON **2**

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**

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. Specify the HTTP request method to a resource method. ?

Procedure name: GETBYID

URI path template for resource: /

HTTP request method: GET 1

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

HTTP header information:

Allowed input media types: \*ALL 3 or...

Returned output media types: \*JSON 4

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 5	id	*NONE or...

Specify the HTTP Method

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

Send and 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. Specify the HTTP request method to a resource method. ?

Specify the HTTP Method

Procedure name: GETALL

URI path template for resource: /

HTTP request method: GET **1**

URI path template for method: \*NONE or...

HTTP header information:

Allowed input media types: \*ALL **2** or...

Returned output media types: \*JSON **3**

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



# 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.

<b>Service</b>	<b>JDBC Properties</b>	<b>Methods</b>
<b>Resource name:</b> students		
<b>Resource description:</b> SQL		
<b>Service install path :</b> /www/wservice/webservices/services/students		
<b>URI path template:</b> /		
<b>User ID for service:</b> *SERVER (QWSERVICE)		
<b>Back</b>	<b>Finish</b>	<b>Cancel</b>

# 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	Running	SOAP	Automatic	<a href="#">View WSDL</a>
<input checked="" type="radio"/>	students	Running	REST	Automatic	<a href="#">View Swagger</a>

- 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



```
▼ students_GETALL_R:  
  ▼ 0:  
    studentID: "823M934LA"  
    firstName: "Nadir"  
    lastName: "Amra"  
    gender: "Male"  
  ▼ 1:  
    studentID: "826M660CF"  
    firstName: "John"  
    lastName: "Doe"  
    gender: "Male"  
  ▼ 2:  
    studentID: "747F023ZX"  
    firstName: "Jane"  
    lastName: "Amra"  
    gender: "Female"  
  —
```

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

# Test the service



SoapUI 5.4.0

Empty SOAP REST Import Save All Forum Trial Preferences Proxy

Search Forum

Projects

- REST Project 1
  - http://common1.frankeni.com:10268
    - Students [/web/services/stud
      - Students 1
        - Request 1

swagger.json

Request 1

Name	Value	Style	Level

Required:  Sets if parameter is

Media Type: application/json

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

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

Create the input data source for the new student



# Test the service



The screenshot shows a web browser window with the address bar displaying `common1.frankeni.com:10268/web/services/students/`. The browser's developer tools are open, showing the response for a `students_GETALL_R:` request. The response is a JSON array of five student objects. The first object is expanded, showing its fields: `studentID`, `firstName`, `lastName`, and `gender`.

```
students_GETALL_R:  
  0:  
    studentID: "823M934LA"  
    firstName: "Nadir"  
    lastName: "Amra"  
    gender: "Male"  
  1:  
    studentID: "826M660CF"  
    firstName: "John"  
    lastName: "Doe"  
    gender: "Male"  
  2:  
    studentID: "747F023ZX"  
    firstName: "Jane"  
    lastName: "Amra"  
    gender: "Female"  
  3:  
    studentID: "123M45678"  
    firstName: "Tim"  
    lastName: "Rowe"  
    gender: "Male"  
  4:  
    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

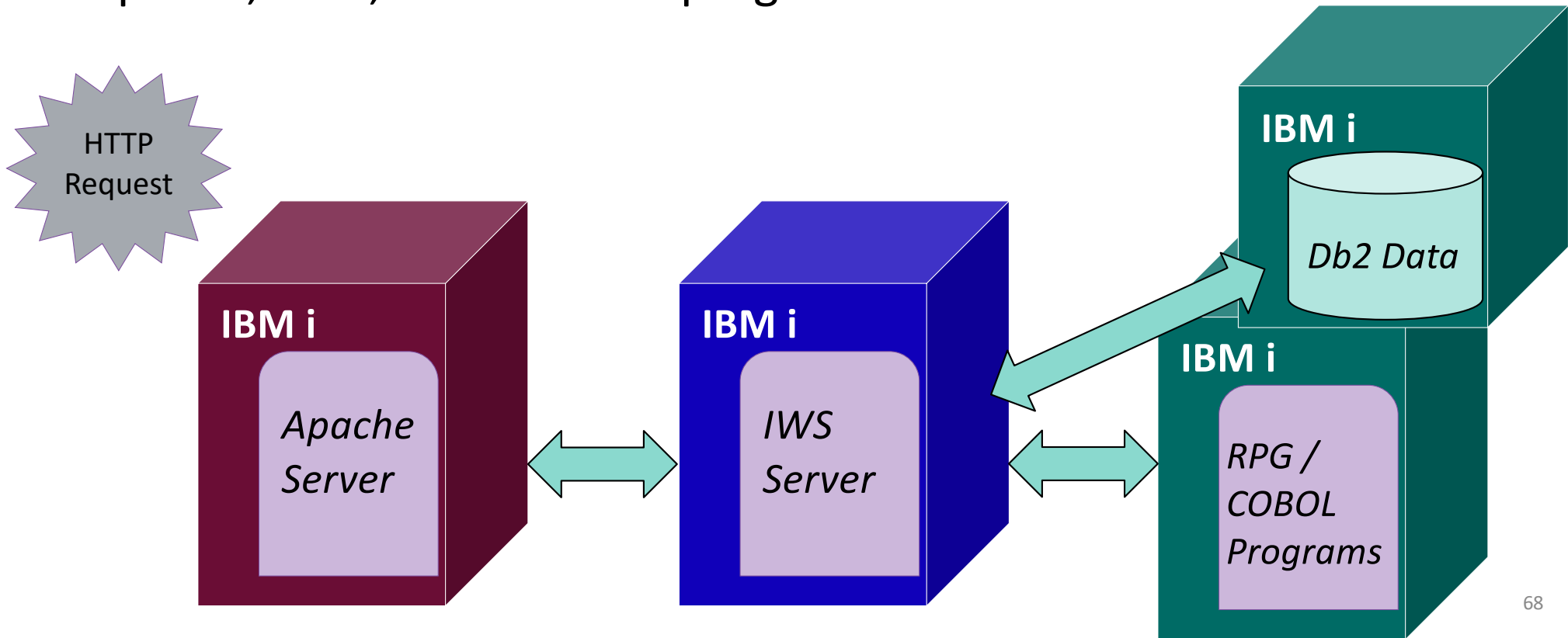
<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/dW%20IBM%20Integrated%20Web%20Services%20for%20i>

# 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/>
- <http://www.ibm.com/developerworks/ibmi/library/i-rest-web-services-server2/>
- <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: <a href="http://www.ibm.com/systems/i">www.ibm.com/systems/i</a></p> <p>IBM 30<sup>th</sup> Anniversary <a href="http://ibmi30.mybluemix.net/">http://ibmi30.mybluemix.net/</a></p> <p>IBM Systems Magazine IBM i Edition: <a href="http://ibmsystemsmag.com/ibmi/">http://ibmsystemsmag.com/ibmi/</a></p> <p>Support Life Cycle: <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: <a href="https://www-01.ibm.com/support/docview.wss?uid=nas8N1022087">https://www-01.ibm.com/support/docview.wss?uid=nas8N1022087</a></p>	<p><a href="#">@IBMSystems</a> <a href="#">@COMMONug</a> <a href="#">@IBMChampions</a> <a href="#">@IBMSystemsISVs</a> <a href="#">@IBMiMag</a> <a href="#">@ITJungleNews</a> <a href="#">@SAPonIBMi</a> <a href="#">@SiDforIBMi</a></p>	<p>#IBMi30 #PowerSystems #IBMi #IBMAIX #POWER8 #LinuxonPower #OpenPOWER #HANAonPower #ITInfrastructure #OpenSource #HybridCloud #BigData</p>

# For More Information:



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