

7 déc. 2023

Événement Online

# iBelieve

Présent et Futur de l'IBM i 2023

Un événement organisé par



avec la participation d'



Avec :



# Web Services/API exposer et consommer des données par SQL

Un événement organisé par



avec la participation d'



**iBelieve** 2023  
Présent et Futur de l'IBM i





# Présentation

Nathanaël BONNET

IBM i depuis 1999

Expert IBM i



GAIA / VOLUBIS

Formation (débutant, perfectionnement)

Expertise IBM i

Centre de Services



## Approche Data Centric

- Décloisonner les SI
- Circulation de la donnée

## Pourquoi DB2 SQL ?

- Standard
- Puissant
- Riche fonctionnellement (TR)

## Pourquoi Web Service / API web ?

- Standard
- Technologie simple (texte = agnostique)

## Peut-on faire autrement ?

- Accès SQL via JDBC/ODBC/Provider .Net
- Remote command - Appel de programme

# CRUD vs WS REST (RESTful)

CRUD, acronyme pour

- Create
- Read
- Update
- Delete

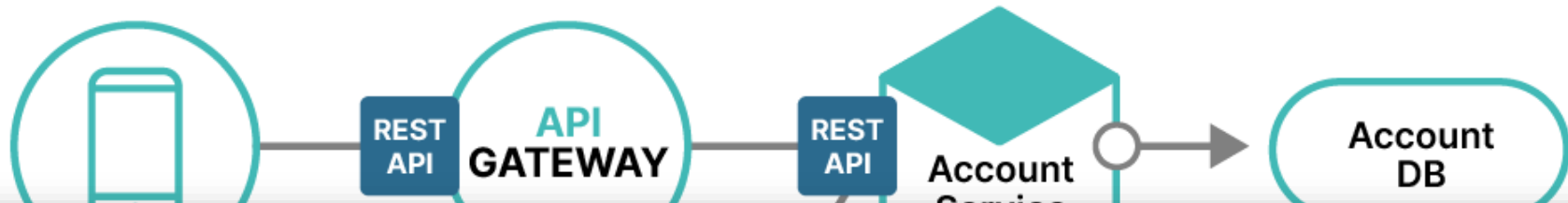
Opérations basiques de la BD

Web Service REST  
(REpresentational State Transfer)

- Style d'architecture
- Basé sur HTTP
- Non limité à CRUD

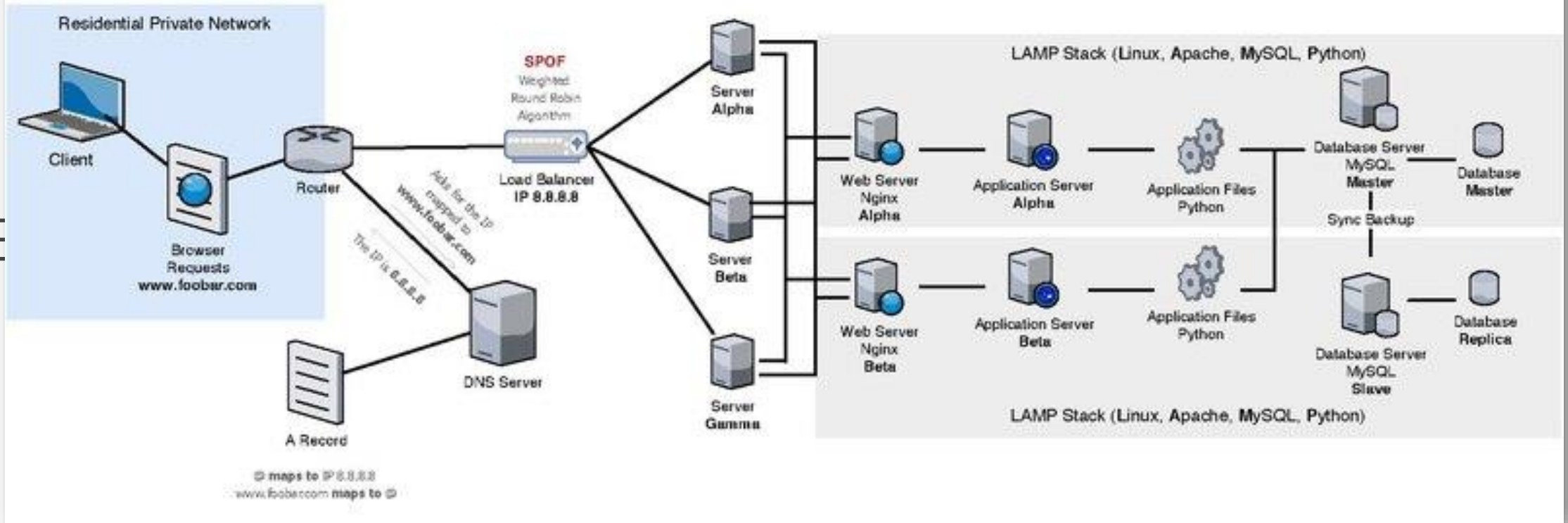
CRUD	HTTP	REST
Create	POST	/api/movie
Read	GET	/api/movie/{id}
Update	PUT	/api/movie
Delete	DELETE	/api/movie/{id}





istribuée

# 1. Distributed Web Infrastructure





# Exposition

Un événement organisé par



avec la participation d'



**iBelieve** 2023  
Présent et Futur de l'IBM i





# IWS – Integrated Web Services

## Fourni par IBM

- Gratuit
- Maintenu par PTF/TR

## Le serveur permet l'exposition de web services / APIs web depuis

- Des requêtes SQL
- Du code ILE (programme / programme de service)

## Stack

- Java

Référence : <https://www.ibm.com/support/pages/integrated-web-services-ibm-i-web-services-made-easy>



## Exposition d'instructions SQL classiques

- SELECT
- UPDATE
- INSERT
- DELETE

En tant que Web Service

```
set current schema = ibelieve23 ;

select * from EMPLOYEE ;
select * from EMPLOYEE where empno = '000010' ;
select * from EMPLOYEE where workdept = 'A00' ;
select * from EMPLOYEE where upper(lastname) like upper('%haa%') ;

delete from employee where empno = '990010' ;
```

Démo



1. Préparez vos instructions
  - Par exemple avec ACS
2. Préparez votre instance IWS
  - Via l'administration HTTP : <http://lpar:2001/HTTPAdmin>
3. Déployez vos services

- Type de service : REST obligatoire pour SQL
- Paramètres JDBC
  - Database : poste de BD Relationnelle (cf WRKRDBDIRE)
  - Default schema
  - Naming convention : \*SQL (.) ou \*SYS (/) (influe sur la résolution des objets non qualifiés)
  - Library List : \*LIBL de l'utilisateur

The screenshot shows the IBM Web Administration interface for server 'ibelieve23 - V2.6 (web services)'. The 'Deploy New Service' wizard is active, showing the following configuration:

- Specify Web service type: **REST**
- Specify Web service implementation: **SQL as a Web service**
- Specify database properties that will be used to process SQL statements:
  - Database system: localhost
  - Default schema: IBELIEVE23
  - Naming convention: \*SQL
  - Library list: \*LIBL



- Resource name
  - Apparaît dans l'URL
- Service Description
  - Uniquement pour IWS
- URI path template
  - Complément de l'URL

[ibelleve23](#) > [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 incoming HTTP requests.

Resource name:

Service description:

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

- Associer un nom de « procédure » REST à une instruction SQL
  - Passage de paramètre admis
- Plusieurs « procédures » supportées



# Exposition SQL - 1

## 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"/> readByNo	select * from EMPLOYEE where empno = ? order by empno		
	empno	input	CHAR
<input type="checkbox"/> readByName	select * from EMPLOYEE where upper(lastname) like upper('%'    ?    '%') order by lastname, firstnme		
	name	input	VARCHAR
<input type="checkbox"/> readByDept	select * from EMPLOYEE where workdept = ? order by workdept, lastname, firstnme		
	dept	input	CHAR
<input type="checkbox"/> create	insert into employee(EMPNO,FIRSTNME,MIDINIT,LASTNAME,WORKDEPT,PHONENO,HIREDATE,JOB,EDLEVEL,SEX,BIRTHDATE,SALARY,BONUS,COMM) values(?,?,?,?,?,?,?,?,?,?,?,?,?)		
	EMPNO	input	CHAR
	FIRSTNME	input	VARCHAR
	MIDINIT	input	CHAR
	LASTNAME	input	VARCHAR
	WORKDEPT	input	CHAR
	PHONENO	input	CHAR
	HIREDATE	input	DATE
	JOB	input	CHAR
	EDLEVEL	input	SMALLINT
	SEX	input	CHAR
	BIRTHDATE	input	DATE
	SALARY	input	DECIMAL
	BONUS	input	DECIMAL
	COMM	input	DECIMAL

# Exposition SQL - 1

<input type="checkbox"/>	delete	delete from employee where empno = ?		
		empno	input	CHAR
<input type="checkbox"/>	updateIdentity	UPDATE employee SET (FIRSTNME, MIDINIT, LASTNAME, PHONENO, EDLEVEL, SEX, BIRTHDATE) = (?, ?, ?, ?, ?, ?, ?) where empno = ?		
		FIRSTNME	input	VARCHAR
		MIDINIT	input	CHAR
		LASTNAME	input	VARCHAR
		PHONENO	input	CHAR
		EDLEVEL	input	SMALLINT
		SEX	input	CHAR
		BIRTHDATE	input	DATE
		empno	input	CHAR
<input type="checkbox"/>	updateJob	UPDATE employee SET (WORKDEPT, HIREDATE, JOB, SALARY, BONUS, COMM) = (?, ?, ?, ?, ?, ?) where empno = ?		
		WORKDEPT	input	CHAR
		HIREDATE	input	DATE
		JOB	input	CHAR
		SALARY	input	DECIMAL
		BONUS	input	DECIMAL
		COMM	input	DECIMAL
		empno	input	CHAR

- SQL result type
  - Permet aussi la gestion des médias : pdf, images
- Gestion des erreurs
  - Interface entre le monde DB2 et HTTP

## 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 re

Procedure name: readByNo  
SQL Statement: select \* from EMPLOYEE where empno = ? order by empno

SQL result type: Single-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: 200 or... ▼

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



- Méthode HTTP
- URI path template
  - Indique les paramètres reçus sur l'URL
- Format
  - Des données reçues (POST/PUT) ou renvoyées

## 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 by mapping an HTTP

Procedure name: readByDept  
URI path template for resource: /  
HTTP request method: GET  
URI path template for method: /dept/{dept} or...  
HTTP header information: \*NONE  
Allowed input media types: \*ALL or...  
Returned output media types: \*JSON  
Identifier for input wrapper element: readByDeptInput or...  
Identifier for output wrapper element: readByDeptResult or...  
 Wrap input parameters  
Input parameter mappings:

Parameter name	Data type	Input source	Identifier	Default Value
dept	CHAR	*PATH_PARAM or...	dept or...	*NONE or...

- Pour les valeurs reçues
  - Possibilités de « wrap » dans du JSON ou XML (contrôle du nom de la racine dans ce cas)

## 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.

Procedure name:	create
URI path template for resource:	/
HTTP request method:	POST ▼
URI path template for method:	*NONE or... ▼
HTTP header information:	*NONE
Allowed input media types:	*JSON or... ▼
Returned output media types:	*JSON ▼
Identifier for input wrapper element:	createInput or... ▼
Identifier for output wrapper element:	createResult or... ▼
<input checked="" type="checkbox"/> Wrap input parameters	




# Utilisation

- IWS génère le Swagger pour faciliter l'utilisation

## Manage Deployed Services

Data current as of 5 déc. 2023 22:43:00.

Find service:

Service name ^	Status	Type	Startup type	Service definition
ConvertTemp	<span style="color: green;">●</span> Running	SOAP	Automatic	 View WSDL
employeeCRUD	<span style="color: green;">●</span> Running	REST	Automatic	 View Swagger
employeeCRUDPS	<span style="color: green;">●</span> Running	REST	Automatic	 View Swagger

Total 3 items

```
{} employeeCRUD.json X
C:\Users\...> Users > nbonnet > Documents > Doc > Evenements > 2023-12-07 - iBelieve 2023 > prépa >
1
2  "swagger": "2.0",
3  "info": {
4    "title": "employeeCRUD APIs",
5    "description": "APIs available for employeeCRUD",
6    "version": "1.0.0"
7  },
8  "host": "itest9:10011",
9  "schemes": [ "http" ],
10 "basePath": "/web/services/employeeCRUD",
11 "tags": [
12   {
13     "name": "employeeCRUD APIs",
14     "description": "APIs available for employeeCRUD"
15   }
16 ],
17 "definitions": {
18   "SQLException": {
19     "type": "object",
20     "properties": {
21       "SQLState": {
22         "type": "string",
23         "maxLength": 5
24       },
25       "errorCode": {
26         "type": "integer"
27       },
28       "message": {
29         "type": "string"
30       }
31     }
32   },
33   "SQLStateInfo": {
34     "type": "object",
35     "properties": {
36       "rowsAffectedCounts": {
37         "type": "string"
38       },
39       "SQLError": {
40         "$ref": "#/definitions/SQLException"
41       },
42       "SQLWarnings": {
43         "type": "array",
44         "items": {
45           "$ref": "#/definitions/SQLException"
46         }
47       }
48     }
49   }
50 }
```



Démo

# Appel via Postman

The screenshot shows the Postman interface for an API call. The top navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore'. The user is logged in as 'iBelieve 2023'. The left sidebar shows a collection named 'employeeCRUD APIs' with several folders and a 'GET read All' endpoint selected. The main area shows the details of the 'GET read All' endpoint, including the URL 'employeeCRUD APIs / read All' and the method 'GET'. The 'Headers' tab is active, showing a table of headers:

Key	Value
Accept	*/*
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
Accept	application/json

The 'Body' tab is also active, showing the response in 'Pretty' format:

```
1 {
2   "employeeCRUD_ReadAll_R": [
3     {
4       "EMPNO": "000010",
5       "FIRSTNME": "CHRISTINE",
6       "MIDINIT": "I",
7       "LASTNAME": "HAAS",
8       "WORKDEPT": "A00",
9       "PHONENO": "3978",
10      "HIREDATE": "1965-01-01",
11      "JOB": "PRES",
12      "EDLEVEL": 18,
13      "SEX": "F",
14      "BIRTHDATE": "1933-08-24",
15      "SALARY": 52750.00,
16      "BONUS": 1000.00,
17      "COMM": 4220.00
18    },
19    {
20      "EMPNO": "000020",
21      "FIRSTNME": "MICHAEL",
22      "MIDINIT": "L",
23      "LASTNAME": "THOMPSON",
24      "WORKDEPT": "B01",
25      "PHONENO": "3476",
26      "HIREDATE": "1973-10-10",
27      "JOB": "MANAGER",
28      "EDLEVEL": 18,
29      "SEX": "M",
30      "BIRTHDATE": "1948-02-02",
31      "SALARY": 41250.00,
```



iBelieve 2023    New    Import    Overview    GET read All    GET read By Name    +    ...

employeeCRUD APIs / name / {name} / read By Name

GET    {{baseUrl}}/name/:name

Params    Authorization    Headers (8)    Body    Pre-request Script    Tests    Settings

Query Params

Key	Value
Key	Value

Path Variables

Key	Value
name	ha

Body    Cookies    Headers (4)    Test Results

Pretty    Raw    Preview    Visualize    JSON    ...

```
1  [
2  "employeeCRUDPS_ReadByName_R": [
3    {
4      "EMPNO": "000010",
5      "FIRSTNAME": "CHRISTINE",
6      "MIDINIT": "I",
7      "LASTNAME": "HAAS",
8      "WORKDEPT": "A00",
9      "PHONENO": "3978",
10     "HIREDATE": "1965-01-01",
11     "JOB": "PRES",
12     "EDLEVEL": 18,
13     "SEX": "F",
14     "BIRTHDATE": "1933-08-24",
15     "SALARY": 52750.00,
16     "BONUS": 1000.00,
17     "COMM": 4220.00
18   }
19 ]
20 ]
```

# Exposition SQL - 1



# Exposition SQL - 1

Overview GET read All GET read By Name POST create

HTTP employeeCRUDPS APIs / / / create

POST {{baseUrl}}/

Params Authorization Headers (11) Body Pre-request Script Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL  JSON

```
1  [
2  · "NEW_EMPNO": "990010",
3  · "NEW_FIRSTNAME": "NATHANAEL",
4  · "NEW_MIDINIT": " ",
5  · "NEW_LASTNAME": "BONNET",
6  · "NEW_WORKDEPT": "D11",
7  · "NEW_PHONENO": "0090",
8  · "NEW_HIREDATE": "2023-12-07",
9  · "NEW_JOB": "ANALYST",
10 · "NEW_EDLEVEL": "12",
11 · "NEW_SEX": "M",
12 · "NEW_BIRTHDATE": "1977-07-08",
13 · "NEW_SALARY": "30000",
14 · "NEW_BONUS": "1500",
15 · "NEW_COMM": "1000"
16 ]
```

## Conclusion

- Très simple
- On expose directement le modèle de données
- Peu de contrôles possibles
- Application des contraintes (référentielles, domaine de valeur, clés)

Une instruction SQL permet également l'appel de procédure ou fonction

- Permet des contrôles / application règles de gestion
  - Contrôles brutes sur la donnée (format, domaine de valeur ...), y compris des contrôles dynamiques (date du jour + 30 jours)
  - Code : règle de gestion, calcul, appel à d'autres fonctions/procédures, accès à la base de données ...
- De masquer le modèle de donnée
  - Masquer la complexité = facilité l'usage
  - Ne pas exposer les détails (historiques) de l'implémentation
    - Une table employee contenant l'ensemble des employés, présents ou non
    - Les informations des employés répartis dans plusieurs fichiers





## On obtient

```
CREATE OR REPLACE PROCEDURE IBELIEVE23.ADD_ROW_TO_EMPLOYEE (  
  IN NEW_EMPNO CHARACTER(6) CCSID 297,  
  IN NEW_FIRSTNME VARCHAR(12) CCSID 297,  
  IN NEW_MIDINIT CHARACTER(1) CCSID 297,  
  IN NEW_LASTNAME VARCHAR(15) CCSID 297,  
  IN NEW_WORKDEPT CHARACTER(3) CCSID 297,  
  IN NEW_PHONENO CHARACTER(4) CCSID 297,  
  IN NEW_HIREDATE DATE,  
  IN NEW_JOB CHARACTER(8) CCSID 297,  
  IN NEW_EDLEVEL SMALLINT,  
  IN NEW_SEX CHARACTER(1) CCSID 297,  
  IN NEW_BIRTHDATE DATE,  
  IN NEW_SALARY DECIMAL(9, 2),  
  IN NEW_BONUS DECIMAL(9, 2),  
  IN NEW_COMM DECIMAL(9, 2)  
)
```

MODIFIES SQL DATA

PROGRAM TYPE SUB

SPECIFIC IBELIEVE23.ADD\_ROW\_TO\_EMPLOYEE

SET OPTION

```
  COMMIT = *NONE,  
  DYNUSRPRF = *USER,  
  USRPRF = *USER
```

-- Contrôles

```
  if (NEW_EMPNO = ' ' or NEW_EMPNO = '000000') then  
    signal sqlstate '22601' set message_text = 'Numéro incorrect' ; -- Classe 22 = Data Exception  
  end if;
```

```
  if (NEW_SALARY <= 0) then  
    signal sqlstate '22602' set message_text = 'Salaire incorrect' ; -- Classe 22 = Data Exception  
  end if;
```

```
BEGIN  
  DECLARE EXIT HANDLER FOR SQLEXCEPTION  
  BEGIN  
    DECLARE LOCAL_SQLCODE INTEGER;  
    DECLARE LOCAL_SQLSTATE CHAR(5) FOR SBCS DATA;  
    DECLARE LOCAL_MESSAGE_TEXT VARCHAR(200) FOR SBCS DATA;  
    DECLARE JOBLOG_MESSAGE VARCHAR(1000) FOR SBCS DATA;  
    GET DIAGNOSTICS CONDITION 1  
      LOCAL_SQLCODE = DB2_RETURNED_SQLCODE,  
      LOCAL_SQLSTATE = RETURNED_SQLSTATE,  
      LOCAL_MESSAGE_TEXT = MESSAGE_TEXT;  
    SET JOBLOG_MESSAGE = 'IBELIEVE23.ADD_ROW_TO_EMPLOYEE  
      CONCAT ' SQLCODE=' CONCAT LOCAL_SQLCODE  
      CONCAT ' SQLSTATE=' CONCAT LOCAL_SQLSTATE  
      CONCAT ' MESSAGE_TEXT=' CONCAT LOCAL_MESSAGE_TEXT;  
    CALL SYSTOOLS.LPRINTF(JOBLOG_MESSAGE);  
    resignal ;  
  END;
```

```
INSERT INTO IBELIEVE23.EMPLOYEE (  
  EMPNO, /* EMPNO CHARACTER(6) */  
  FIRSTNME, /* FIRSTNME VARCHAR(12) */  
  MIDINIT, /* MIDINIT CHARACTER(1) */  
  LASTNAME, /* LASTNAME VARCHAR(15) */  
  WORKDEPT, /* WORKDEPT CHARACTER(3) */  
  PHONENO, /* PHONENO CHARACTER(4) */  
  HIREDATE, /* HIREDATE DATE */  
  JOB, /* JOB CHARACTER(8) */  
  EDLEVEL, /* EDLEVEL SMALLINT */  
  SEX, /* SEX CHARACTER(1) */  
  BIRTHDATE, /* BIRTHDATE DATE */  
  SALARY, /* SALARY DECIMAL(9,2) */  
  BONUS, /* BONUS DECIMAL(9,2) */  
  COMM /* COMM DECIMAL(9,2) */  
)  
VALUES (  
  NEW_EMPNO, /* CHARACTER(6) */  
  NEW_FIRSTNME, /* VARCHAR(12) */  
  NEW_MIDINIT, /* CHARACTER(1) */  
  NEW_LASTNAME, /* VARCHAR(15) */  
  NEW_WORKDEPT, /* CHARACTER(3) */  
  NEW_PHONENO, /* CHARACTER(4) */  
  NEW_HIREDATE, /* DATE */  
  NEW_JOB, /* CHARACTER(8) */  
  NEW_EDLEVEL, /* SMALLINT */  
  NEW_SEX, /* CHARACTER(1) */  
  NEW_BIRTHDATE, /* DATE */  
  NEW_SALARY, /* DECIMAL(9,2) */  
  NEW_BONUS, /* DECIMAL(9,2) */  
  NEW_COMM /* DECIMAL(9,2) */  
);  
END;
```



## Appel

```
CALL IBELIEVE23.ADD_ROW_TO_EMPLOYEE (  
  NEW_EMPNO      => '600010', /* IN CHARACTER(6) */  
  NEW_FIRSTNAME => 'NATHANAEL', /* IN VARCHAR(12) */  
  NEW_MIDINIT    => ' ', /* IN CHARACTER(1) */  
  NEW_LASTNAME   => 'BONNET', /* IN VARCHAR(15) */  
  NEW_WORKDEPT  => 'D11', /* IN CHARACTER(3) */  
  NEW_PHONENO   => '0099', /* IN CHARACTER(4) */  
  NEW_HIREDATE  => current date, --'2023-12-07', /* IN DATE */  
  NEW_JOB       => 'ANALYST', /* IN CHARACTER(8) */  
  NEW_EDLEVEL   => 15, /* IN SMALLINT */  
  NEW_SEX       => 'M', /* IN CHARACTER(1) */  
  NEW_BIRTHDATE => current date, --'1977-08-07', /* IN DATE */  
  NEW_SALARY    => 30000, /* IN DECIMAL(9,2) */  
  NEW_BONUS     => 1500, /* IN DECIMAL(9,2) */  
  NEW_COMM      => 2000 /* IN DECIMAL(9,2) */  
);
```



## Exposition via IWS

- Identique
- Analyse automatique des paramètres

Support des result sets en sortie

### 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"/>	readByNo	CALL GET_BY_NUM_FROM_EMPLOYEE (?)		
		P_EMPNO	input	CHAR
<input type="checkbox"/>	readByName	CALL GET_BY_NAME_FROM_EMPLOYEE (?)		
		P_LASTNAME	input	VARCHAR
<input type="checkbox"/>	readByDept	CALL GET_BY_DEPT_FROM_EMPLOYEE (?)		
		P_WORKDEPT	input	CHAR
<input type="checkbox"/>	readAll	CALL GET_ALL_FROM_EMPLOYEE()		
<input type="checkbox"/>	create	CALL ADD_ROW_TO_EMPLOYEE (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)		
		NEW_EMPNO	input	CHAR
		NEW_FIRSTNME	input	VARCHAR
		NEW_MIDINIT	input	CHAR
		NEW_LASTNAME	input	VARCHAR
		NEW_WORKDEPT	input	CHAR
		NEW_PHONENO	input	CHAR
		NEW_HIREDATE	input	DATE
		NEW_JOB	input	CHAR
		NEW_EDLEVEL	input	SMALLINT
		NEW_SEX	input	CHAR
		NEW_BIRTHDATE	input	DATE
		NEW_SALARY	input	DECIMAL
		NEW_BONUS	input	DECIMAL
		NEW_COMM	input	DECIMAL

## IWS supporte différents mécanismes de sécurité

- Certificat SSL/TLS
- Authentification basique : permet de régler pour chaque service/méthode les droits d'accès par utilisateur
- Application des droits de l'utilisateur : soit un utilisateur générique soit un utilisateur authentifié (authentification basique)
- TAI (Trust Association Interface) : code Java personnalisable déclenché par le serveur avant l'exécution du service (permet de gérer une authentification OAuth ou JWT par exemple)

## D'autres technologies à votre disposition

- APACHE + CGI (CGIDEV2, HTTPAPI, ...)
- NodeJS
- PHP
- Java
- ILEAstic
- ...



# Consommation

Un événement organisé par



avec la participation d'



**iBelieve** 2023  
Présent et Futur de l'IBM i



## SQL

- Intégration de fonctions liées à la données
  - JSON / XML
- Intégration de fonctions techniques
  - HTTP, base64, encodage URL
- ➔ permet l'appel de web services (ressources web au sens large)

## SQL embarqué

- Toutes les instructions SQL sont supportées

## Facilité de mise au point

- Facilité de mise au point des requêtes avec ACS



# Syntaxe

## SQL

00001

```
{"employeeCRUD_ReadByName_R" : [{"EMPNO" : "000150", "FIRSTNME" : "BRUCE", "MIDINIT" : "", "LASTNAME" : "ADAMSON", "WORKDEPT" : "D11", "PHON...
```

```
-- Liste des Employées dont le nom contient 'ON'  
values qsys2.http_get('http://itest9:10011/web/services/employeeCRUD/name/' || trim('on'), '' );
```

```
select *  
from json_table( qsys2.http_get('http://itest9:10011/web/services/employeeCRUD/name/' || trim('on'), ''),  
                'lax $.employeeCRUD_ReadByName_R[*]' columns( EMPNO      VARCHAR(6)  PATH 'lax $.EMPNO',  
                                                             FIRSTNME  VARCHAR(30) PATH 'lax $.FIRSTNME',  
                                                             MIDINIT   CHAR(1)     PATH 'lax $.MIDINIT',  
                                                             LASTNAME  VARCHAR(30) PATH 'lax $.LASTNAME',  
                                                             WORKDEPT  CHAR(3)     PATH 'lax $.WORKDEPT',  
                                                             PHONENO   CHAR(4)     PATH 'lax $.PHONENO',  
                                                             HIREDATE  DATE        PATH 'lax $.HIREDATE',  
                                                             JOB       VARCHAR(20) PATH 'lax $.JOB',  
                                                             EDLEVEL   SMALLINT   PATH 'lax $.EDLEVEL',  
                                                             SEX       CHAR(1)     PATH 'lax $.SEX',  
                                                             BIRTHDATE DATE        PATH 'lax $.BIRTHDATE',  
                                                             SALARY    DEC(9,2)   PATH 'lax $.SALARY',  
                                                             BONUS     DEC(9,2)   PATH 'lax $.BONUS',  
                                                             COMM      DEC(9,2)   PATH 'lax $.COMM' ) ) emp;
```

JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
DESIGNER	16	M	1947-05-17	25280.00	500.00	2022.00
FIELDREP	16	M	1926-05-17	23840.00	500.00	1907.00
ANALYST	15	M	2023-12-04	30000.00	1500.00	2000.00
MANAGER	16	F	1941-05-15	29750.00	600.00	2380.00
CLERK	14	M	1935-05-30	22180.00	400.00	1774.00
CLERK	16	F	1936-10-05	17250.00	300.00	1380.00
DESIGNER	17	M	1953-02-23	18270.00	400.00	1462.00
CLERK	17	M	1954-03-31	28760.00	600.00	2301.00
CLERK	14	M	1942-10-18	29250.00	600.00	2340.00
MANAGER	18	M	1948-02-02	41250.00	800.00	3300.00
FIELDREP	14	F	1941-07-18	25370.00	500.00	2030.00

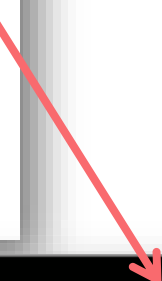
000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15
000230	JAMES	J	JEFFERSON	D21	2094	1966-11-21
000260	SYBIL	P	JOHNSON	D21	8953	1975-09-11
000210	WILLIAM	T	JONES	D11	0942	1979-04-11
200240	ROBERT	M	MONTEVERDE	D21	3780	1979-12-05
000120	SEAN		O'CONNELL	A00	2167	1963-12-05
000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10
200330	HELENA		WONG	E21	2103	1976-02-23



# Syntaxe

```
values json_object(  
  'EMPNO'      value '990010',  
  'FIRSTNME'  value 'NATHANAEL',  
  'MIDINIT'   value ' ',  
  'LASTNAME'  value 'BONNET',  
  'WORKDEPT'  value 'D11',  
  'PHONENO'   value '0090',  
  'HIREDATE'  value '2023-12-07',  
  'JOB'       value 'ANALYST',  
  'EDLEVEL'   value 12,  
  'SEX'       value 'M',  
  'BIRTHDATE' value '1977-07-08',  
  'SALARY'    value 30000,  
  'BONUS'     value 1500,  
  'COMM'     value 1000 );
```

```
values qsys2.http_post('http://itest9:10011/web/services/employeeCRUD/',  
  json_object(  
    'EMPNO'      value '990010',  
    'FIRSTNME'  value 'NATHANAEL',  
    'MIDINIT'   value ' ',  
    'LASTNAME'  value 'BONNET',  
    'WORKDEPT'  value 'D11',  
    'PHONENO'   value '0090',  
    'HIREDATE'  value '2023-12-07',  
    'JOB'       value 'ANALYST',  
    'EDLEVEL'   value 12,  
    'SEX'       value 'M',  
    'BIRTHDATE' value '1977-07-08',  
    'SALARY'    value 30000,  
    'BONUS'     value 1500,  
    'COMM'     value 1000 ) ,  
    {'headers': {'Content-Type': 'application/json'}}' );
```



```
{  
  "EMPNO": "990010",  
  "FIRSTNME": "NATHANAEL",  
  "MIDINIT": " ",  
  "LASTNAME": "BONNET",  
  "WORKDEPT": "D11",  
  "PHONENO": "0090",  
  "HIREDATE": "2023-12-07",  
  "JOB": "ANALYST",  
  "EDLEVEL": 12,  
  "SEX": "M",  
  "BIRTHDATE": "1977-07-08",  
  "SALARY": 30000,  
  "BONUS": 1500,  
  "COMM": 1000  
}
```

## SQLRPGLE

- On modifie un peu pour permettre une meilleure maintenance
- Indentification des erreurs

```
// Variables globales
dcl-s httpstatus int(10) ;
dcl-s resp_msg  sqltype(clob: 1000000) ccsid 1208;
dcl-s resp_head varchar(25000) ;

dcl-ds emp Qualified;
  empno char(6) ;
  prenom varchar(30) ;
  nom    varchar(30) ;
end-ds ;
```

```
// Récupérer les employés !
exec sql
  select r.RESPONSE_MESSAGE,
         r.RESPONSE_HTTP_HEADER,
         h.code
  into :resp_msg, :resp_head, :httpstatus
  from table( qsys2.http_get_verbose(
             'http://itest9:10011/web/services/employeeCRUD/',
             '{"headers":{"Content-Type":"application/json"}}' ) ) as r,
         json_table(RESPONSE_HTTP_HEADER,
                   '$' columns (code int path '$.HTTP_STATUS_CODE') ) as h;
```

## SQLRPGLE

```
// Erreur SQL ?
if SqlCode < 0 ;
  dsply ('Erreur. SQLCODE :' + %char( SqlCode ) ) ;
  return ;
endif ;
// Erreur HTTP ?
if httpstatus <> 200 ;
  dsply ('Erreur. Status HTTP : ' + %char(httpstatus) ) ;
  return ;
endif ;
```

```
// Extraire les employés
exec sql
declare c_emp cursor for
select *
from json_table( :resp_msg,
                 'lax $.employeeCRUD_ReadAll_R[*]'
                 columns( EMPNO      VARCHAR(6)   PATH 'lax $.EMPNO',
                          FIRSTNAME  VARCHAR(30)  PATH 'lax $.FIRSTNAME',
                          LASTNAME   VARCHAR(30)  PATH 'lax $.LASTNAME') ) emp;

exec sql open c_emp ;
if SqlCode < 0 ;
  dsply 'Erreur ouverture curseur' ;
  return ;
endif ;

EXEC SQL FETCH NEXT FROM c_emp INTO :emp;

Dow SqlCode >=0 and SqlCode < 100;
  dsply ('Employé : ' + emp.EMPNO);
  dsply ('..... prénom : ' + emp.prenom);
  dsply ('..... nom : ' + emp.nom);
  EXEC SQL
  FETCH NEXT FROM c_emp INTO :emp;
ENDDO;

EXEC SQL CLOSE c_emp;
```



## SQLRPGLE

- Résultat

```
4 > call list
DSPLY  Employé : 000010
DSPLY  ..... prénom : CHRISTINE
DSPLY  ..... nom      : HAAS
DSPLY  Employé : 000020
DSPLY  ..... prénom : MICHAEL
DSPLY  ..... nom      : THOMPSON
DSPLY  Employé : 000030
DSPLY  ..... prénom : SALLY
DSPLY  ..... nom      : KWAN
```

L'usage de SQL permet également de faciliter l'envoi de donnée ou l'usage de la donnée reçue

- Gestion XML/JSON
- Stockage
- Jointure avec d'autres données

## Démo

## Exemple

```

-- Liste des certifications
create table certif (
  titre varchar(128),
  nom   varchar(30),
  prenom varchar(30) );
insert into certif values('Administrator IBM i 7.5', 'JOHNSON', 'SYBIL'), ('Developer IBM i', 'BONNET', 'NATHANAEL'), ('Prix Nobel de la Paix', 'DUNANT', 'HENRY');

-- Liste des employés qui ont une certification
select emp.*, cert.titre
from json_table( qsys2.http_get('http://itest9:10011/web/services/employeeCRUD/', ''),
  'lax $.employeeCRUD_ReadAll_R[*]' columns( EMPNO    VARCHAR(6)  PATH 'lax $.EMPNO',
                                             FIRSTNME  VARCHAR(30) PATH 'lax $.FIRSTNME',
                                             LASTNAME   VARCHAR(30) PATH 'lax $.LASTNAME' ) ) emp
join certif cert on (upper(cert.nom), upper(cert.prenom)) = (upper(emp.LASTNAME), upper(emp.FIRSTNME));

```

EMPNO	FIRSTNME	LASTNAME	TITRE
000260	SYBIL	JOHNSON	Administrator IBM i 7.5
990010	NATHANAEL	BONNET	Developer IBM i



## Exemple

```
-- Générer cette liste en JSON
with emp_json as (
select json_object('num' value emp.empno, 'prenom' value emp.firstname, 'nom' value emp.lastname, 'titre' value cert.titre) as ej
from json_table( qsys2.http_get('http://itest9:10011/web/services/employeeCRUD/', ''),
                'lax $.employeeCRUD_ReadAll_R[*]' columns( EMPNO      VARCHAR(6)  PATH 'lax $.EMPNO',
                                                         FIRSTNME  VARCHAR(30) PATH 'lax $.FIRSTNME',
                                                         LASTNAME  VARCHAR(30) PATH 'lax $.LASTNAME' ) ) emp
   join certif cert on (upper(cert.nom), upper(cert.prenom)) = (upper(emp.LASTNAME), upper(emp.FIRSTNME)) )
select json_object('employe_certif' value json_arrayagg( ej format json ) )
from emp_json ;
```

00001

```
{"employe_certif":[{"num":"000260","prenom":"SYBIL","nom":"JOHNSON","titre":"Administrateur IBM i 7.5"}, {"num":"990010","prenom":"NATHANAEL","nom":"BONNET","titre":"Developer IBM i"}]}
```

```
{
  "employe_certif": [{
    "num": "000260",
    "prenom": "SYBIL",
    "nom": "JOHNSON",
    "titre": "Administrateur IBM i 7.5"
  }, {
    "num": "990010",
    "prenom": "NATHANAEL",
    "nom": "BONNET",
    "titre": "Developer IBM i"
  }
]
}
```

## En restant dans les standards

- cURL
- AXIS ILE
- PHP, Java, NodeJS, Python, ...

DB2 permet de faciliter l'exposition et la consommation des données au travers de web services / APIs web

- Peu de compétences supplémentaires
- Outillage déjà maîtrisé



# Q/R

Un événement organisé par



avec la participation d'



**iBelieve** 2023  
Présent et Futur de l'IBM i



MERCI

Un événement organisé par



avec la participation d'



**iBelieve** 2023  
Présent et Futur de l'IBM i

