



IBM Power Systems - IBM i

Modernisation, développement d'applications et DB2 sous IBM i
Technologies, outils et nouveautés 2013-2014

13 et 14 mai 2014 – IBM Client Center Paris, Bois-Colombes

S2 – Invoquez des Services Web en SQL

Mardi 13 mai – 14h00-15h30

Nathanaël BONNET – Gaïa Mini Systèmes

Pré requis

- IBM i 7.1
 - SF99701 (DB2) niveau 23
 - niveau actuel : 28
 - 5761-JV1 , option 11,12,14 ou 15
 - pour Java 1.6

- Ces fonctions sont livrées dans SYSTOOLS
 - « As is » : pas de support
 - Etape avant intégration définitive
 - Existent également dans la version LUW

DB2

- Base de Données
 - Stockage et restitution de données relationnelles
 - De nombreuses améliorations continues (cf. fonctionnalités livrées par TR au fil de l'eau)

- Dernièrement
 - XML natif dans la BD
 - XML est un autre formalisme de représentation des données

- HTTP
 - Protocole de communication internet
 - A priori loin du monde de la BD ...

DB2 en tant que middleware

- DB2 est utilisé également en tant que middleware
 - Pour faciliter l'interconnexion des applications
 - Depuis de nombreuses années par le biais des procédures cataloguées principalement

- A ce titre, l'ajout de fonctionnalités HTTP permet d'élargir les possibilités techniques
 - Avec le principal standard du monde informatique

HTTP - fonctionnement

- HyperText Transfer Protocol
 - Client = navigateur web
 - Protocole simple, basé sur un encodage caractères

- Principe
 - Le client envoie une requête
 - Une méthode (commande)
 - Une URL (Uniform Resource Locator)
 - `scheme://domain:port/path?query_string`
 - `http://www.ibm.fr`
 - `https://www.google.com/calendar/render?tab=mc`
 - Des paramètres supplémentaires
 - Le serveur renvoie une réponse
 - Un entête technique
 - Un corps contenant les données

HTTP - Exemple

- Depuis un navigateur, accès au site <http://www.gaia.fr>

- Requête

```
GET / HTTP/1.1
Host: www.gaia.fr
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip,deflate,sdch
Accept-Language: fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/34.0.1847.131 Safari/537.36
```

- Réponse

```
HTTP/1.1 200 OK
Cache-control: private, no-cache="set-cookie", must-revalidate
Content-Length: 2685
...

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<base href="http://www.gaia.fr/" />
<title>Gaia &bull; Accueil</title>
...
```

HTTP - Usage

- HTTP fonctionne sur les réseaux TCP
 - Tous les LAN, WAN, internet ...
 - C'est donc le protocole le plus répandu
- DB2 nous permet maintenant d'utiliser en direct des ressources « web »
 - Accéder à des documents de tous types
 - Textes : pages web, flux XML, flux RSS ...
 - Binaires : photos, pdf ...
 - A des fonctionnalités accessibles via HTTP
 - Services Web, SOAP ou REST
- DB2 est un client HTTP au même titre qu'un navigateur

Retrouver une ressource web

- Reprenons notre exemple : <http://www.gaia.fr>
values SYSTOOLS.HTTPGETCLOB('http://www.gaia.fr' , '') ;
- Avec
 - HTTPGETCLOB
 - Fonction SQL provoquant la commande GET
 - 1^{er} paramètre : URL sur pour commande GET
 - 2nd paramètre : valeurs de l'entête HTTP
 - Valeur de retour : CLOB
- Résultat

```
00001
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN""http://www.w3.org/TR
R/xhtml11/DTD/xhtml11-strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml"...
```


Retrouver une ressource web

- Insertion en BD

```
CREATE TABLE MESPAGES (  
ID          INTEGER      GENERATED ALWAYS AS IDENTITY ,  
ACCUEIL    CLOB (2M)    CCSID 1208 ,  
STAMP      TIMESTAMP   GENERATED ALWAYS FOR EACH ROW  
                ON UPDATE AS ROW CHANGE TIMESTAMP ) ;
```

```
insert into MESPAGES( accueil )  
values(SYSTOOLS.HTTPGETCLOB( 'http://www.know400.fr' , '' )) ;
```

Retrouver une ressource web

- En SQL embarqué, il est possible de stocker la valeur retournée directement dans un fichier stream

```
// déclaration CLOB
dcl-s fichierHTML SQLTYPE( CLOB_FILE ) ;

// créer un fichier .html sur l'IFS
fichierHTML_name = '/home/NB/gaia.html' ;
fichierHTML_nl   = %len( %trim( fichierHTML_name ) ) ;
fichierHTML_fo   = SQFCRT ;

// lecture de données HTML
exec sql values SYSTOOLS.HTTPGETCLOB('http://www.gaia.fr' , '')
           into :fichierHTML ;
if ( SqlCode = 0 ) ; // Erreur ?
```

HTTP - Fonctions

- DB2 fourni un ensemble de fonctions permettant les actions HTTP
 - HEAD, GET, POST, PUT, DELETE

- 2 ensembles de fonctions
 - HTTP* et HTTP*VERBOSE
 - HTTP*
 - UDF - Fonctions scalaires (i.e. renvoie une valeur résultat)
 - HTTP*VERBOSE
 - UDTF - Fonctions table (i.e. renvoie une table contenant une ligne avec une colonne pour la valeur de l'entête HTTP et une colonne pour la valeur résultat)

HTTP - Fonctions

■ Les fonctions

- HTTP[Action][TypeRes]
- Où
 - Action = {PUT, POST, DELETE, GET}
 - TypeRes = {BLOB, CLOB}
- Par exemple
 - HTTPGETCLOB
 - Permet de demander une ressource par la méthode HTTP GET, le résultat étant sous la forme d'un CLOB (Character Large Object)

■ Fonctions génériques

- Ces fonctions sont redondantes avec les précédentes
 - HTTPBLOB et HTTPCLOB
 - Elles admettent le nom de la méthode HTTP en paramètre
 - HTTPGETCLOB(....) est identique à HTTPCLOB('GET' , ...)

HTTP - Fonctions

- Liste des fonctions

- HEAD

- Ne demande que les informations sur la ressource sans demander la ressource elle-même

- GET

- Demande une ressource

- POST

- Transmets des données en vue d'un traitement d'une ressource (formulaire par exemple)

- PUT

- Ajout ou remplacement de ressources

- DELETE

- Supprime une ressource

HTTP - Fonctions

■ Fonctions utilitaires

- URLENCODE et URLDECODE
 - Gestion de l'encodage des URL
- BASE64ENCODE et BASE64DECODE
 - Gestion de l'encodage en base6
 - Permet de transmettre des données binaires dans un format texte

URLENCODE

- Les URL peuvent transmettre des données
 - Recherche Google : "IBM i"
 - <https://www.google.fr/#q=%22IBM+i%22&safe=off>
 - L'encodage des données sur l'URL est normalisé
 - RFC 1738 : <http://www.ietf.org/rfc/rfc1738.txt>
 - En résumé, une URL se compose
 - De caractères ASCII affichables
 - Les autres caractères doivent être « échappés » %XX avec XX code hexadécimal du caractère

- Usage

```
URLENCODE(value          VARCHAR(2048) CCSID 1208,
           encoding VARCHAR(20)   CCSID 1208)
```

 - Renvoie VARCHAR(4096) CCSID 1208
 - Value : valeur à encoder
 - Encoding : jeu de caractères à utiliser(défaut UTF-8)

URLENCODE

■ Exemples

```
values systools.urlencode('chaîne à encoder :  
nbonnet@gaia.fr', '') ;
```

➔ cha%C3%A9ne+%C3%A0+encoder+%3A+nbonnet%40gaia.fr

-- Avec un encodage ASCII :

```
values systools.urlencode('chaîne à encoder :  
nbonnet@gaia.fr', 'ASCII') ;
```

➔ cha%3Fne+%3F+encoder+%3A+nbonnet%40gaia.fr

Retrouver une ressource binaire

- Par exemple une photo

```
values SYSTOOLS.HTTPGETBLOB(
  'http://www.gaia.fr/ressources/nath.jpg' , '' ) ;
```

- Toujours possibilité d'insérer dans une table contenant une colonne de type BLOB et utilisation dans SQL embarqué

00001
FFD8FFE000104A46494600010200006400640000FFEC00114475636B790001...

- Usage de la fonction VERBOSE

```
select *
from table( SYSTOOLS.HTTPGETBLOBVERBOSE(
  'http://www.gaia.fr/ressources/nath.jpg' , '' ) ) rs ;
```

- Résultat

RESPONSEMSG	RESPONSEHTTPHEADER
FFD8FFE000104A464946...	<?xml version="1.0" encoding="UTF-8" ?><httpHeader responseCode="200"><responseMessage>OK</r...

Retrouver un document XML sur le web

- C'est une ressource comme déjà traité
- Mais DB2 sait manipuler nativement des données XML
 - Insertion dans la BD avec validation, syntaxique et XSD optionnelle
 - Accès aux données contenues dans le document XML
- Par exemple, accès au fichier européen des terroristes
 - http://ec.europa.eu/external_relations/cfsp/sanctions/list/version4/global/global.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<WHOLE Date="24/04/2014">
<ENTITY Id="1" Type="P" legal_basis="151/2012 (OJ L 49)" reg_date="2012-02-23" pdf_link="http://eur-lex.europa.eu/eur-lex-content/uris/uris.htm" />
<ENTITY Id="13" Type="P" legal_basis="1210/2003 (OJ L 169)" reg_date="2003-07-09" pdf_link="http://eur-lex.europa.eu/eur-lex-content/uris/uris.htm" />
<NAME Id="17" Entity_id="13" legal_basis="1210/2003 (OJ L 169)" reg_date="2003-07-09" pdf_link="http://eur-lex.europa.eu/eur-lex-content/uris/uris.htm" />
  <LASTNAME>Hussein Al-Tikriti</LASTNAME>
  <FIRSTNAME>Saddam</FIRSTNAME>
  <MIDDLENAME></MIDDLENAME>
  <WHOLENAME></WHOLENAME>
  <GENDER>M</GENDER>
  <TITLE></TITLE>
```

Retrouver un document XML sur le web

- Contrôler que l'on ne fournit pas de service à des terroristes

```
with terroristes as (
  SELECT res.*
  FROM XMLTABLE('$result/WHOLE/ENTITY/NAME'
  PASSING XMLPARSE(DOCUMENT
SYSTOOLS.HTTPGETCLOB('http://ec.europa.eu/external_relations/cfsp/sanctions/
list/version4/global/global.xml' , '')) AS "result" COLUMNS
  nom VARCHAR(128) PATH 'LASTNAME',
  prenom VARCHAR(128) PATH 'FIRSTNAME' ) AS RES )
```

```
select distinct nom, prenom
from terroristes
where nom <> ' ' and prenom <> ' '
order by nom, prenom ;
```

NOM	PRENOM
Kuznyecova	Natalja
Kyagulanyi	Alilabaki
Kyagulanyi	David
KADAFI	Aisza
KADAFI	Hassiba

HTTPHEAD

■ Exemple

```
Values SYSTOOLS.HTTPHEAD('http://www.know400.fr', '');
```

■ Résultat

```
<?xml version="1.0" encoding="UTF-8" ?>
<httpHeader responseCode="200">
  <responseMessage>OK</responseMessage>
  <header name="HTTP_RESPONSE_CODE" value="HTTP/1.1 200 OK" />
  <header name="X-UA-Compatible" value="IE=contours" />
  <header name="Server" value="Apache" />
  <header name="Keep-Alive" value="timeout=5, max=100" />
  <header name="Content-Encoding" value="gzip" />
  <header name="Date" value="Mon, 28 Apr 2014 15:36:40 GMT" />
  <header name="Content-Type" value="text/html; charset=utf-8" />
  ...
  <header name="Content-Length" value="20" />
  <header name="Pragma" value="no-cache" />
  <header name="Connection" value="Keep-Alive" />
</httpHeader>
```

```
00001
```

```
<?xml version="1.0" encoding="UTF-8" ?><httpHeader responseCode="200"><responseMessage>OK</resp...
```

HTTPHEAD

■ Shredding de l'entête

```
with httphead as (
  select XMLPARSE( DOCUMENT SYSTOOLS.HTTPHEAD('http://www.know400.fr', '' )
) as header_ from sysibm.sysdummy1 )
```

```
select x.*
from httphead, xmltable('$d/httpHeader/header' passing header_ as "d"
columns
  nom varchar(64) path '@name' ,
  val varchar(64) path '@value' ) as x ;
```

NOM	VAL
HTTP_RESPONSE_CODE	HTTP/1.1 200 OK
X-UA-Compatible	IE=contours
Server	Apache
Keep-Alive	timeout=5, max=100
Content-Encoding	gzip
Date	Mon, 28 Apr 2014 15:38:22 GMT
Content-Type	text/html; charset=utf-8
Set-Cookie	domus_data=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; path=
Vary	Accept-Encoding

Entête personnalisée

- Dans certains cas, il est nécessaire de modifier l'entête de la requête
 - Principalement dans le cas d'accès à des ressources protégées
 - Nécessitant une authentification HTTP

- Entête personnalisé

- Sous la forme d'un flux XML

```
<httpHeader>  
  <header name="Accept" value="application/atom+xml" />  
</httpHeader>
```

- Ré-encodé ensuite par les fonctions HTTP*

- Exemple

```
values SYSTOOLS.HTTPGETCLOB( 'http://www.gaia.fr',  
'<httpHeader><header name="Accept"  
value="application/atom+xml"/></httpHeader>' ) ;
```

Entête personnalisée

- Dans le cas de GMail qui permet de récupérer le contenu de sa boîte mail moyennant une authentification

```
values ( SYSTOOLS.HTTPGETCLOB(  
  'https://mail.google.com/mail/feed/atom/',  
  '<httpHeader> <header name="Authorization" value="Basic '  
  || SYSTOOLS.BASE64ENCODE( cast(  
  'nathanael.bonnet@mondomaine:motdepasse' as varchar(64) ccsid  
  1208))  
  || '"/> <header name="Accept" value="application/atom+xml"/>  
</httpHeader>' ) ) ;
```

– Retourne

```
00001  
<?xml version="1.0" encoding="UTF-8"?><feed version="0.3" xmlns="http://purl.org/atom/ns#"><title>Gm...
```

Entête personnalisée

```
<?xml version="1.0" encoding="UTF-8"?>
<feed version="0.3" xmlns="http://purl.org/atom/ns#">
  <title>Gmail - Inbox for nathanael.bonnet@<img alt="redacted" data-bbox="488 235 600 255"/></title>
  <tagline>New messages in your Gmail Inbox</tagline>
  <fullcount>6</fullcount>
  <link rel="alternate" href="http://mail.google.com/mail" type="text/html" />
  <modified>2014-04-29T13:56:54Z</modified>
  <entry>
    <title>Rappel : IBM - 13 et 14 mai 2014 - Evénement sur la
      modernisation IBM i + Annonces POWER8 et IBM i 7.2</title>
    <summary>Bonjour à toutes et à tous, Si vous n'êtes pas encore
      inscrit(e) à cet événement, voici trois</summary>
    <link rel="alternate"
      href="http://mail.google.com/mail?account_id=nathanael.bonnet@<img alt="redacted" data-bbox="650 530 750 550"/>&mess
      type="text/html" />
    <modified>2014-04-28T12:52:33Z</modified>
    <issued>2014-04-28T12:52:33Z</issued>
    <id>tag:gmail.google.com,2004:1466632437793931245</id>
    <author>
      <name>Nathanael BONNET</name>
      <email>nathanael.bonnet@gaia.fr</email>
    </author>
    <contributor>
      <name>Philippe Bourgeois1</name>
      <email>pbourgeois@fr.ibm.com</email>
    </contributor>
  </entry>
```


Entête personnalisée

- Qu'il est possible d'exploiter

```
SELECT res.*
FROM XMLTABLE(XMLNAMESPACES(DEFAULT 'http://purl.org/atom/ns#'),
'$result/feed/entry'
PASSING XMLPARSE(DOCUMENT SYSTOOLS.HTTPGETCLOB(
'https://mail.google.com/mail/feed/atom/',
'<httpHeader><header name="Authorization" value="Basic ' ||
SYSTOOLS.BASE64ENCODE(cast( 'nathanael.bonnet@mondomaine:motdepasse' as
varchar(64) ccsid 1208)) || '" /><header name="Accept"
value="application/atom+xml" /></httpHeader>' ) ) AS "result"
COLUMNS
  title          VARCHAR(128)  PATH 'title',
  summary        VARCHAR(1024) PATH '*[local-name()='summary']',
  author_name    VARCHAR(255)  PATH '*[local-name()='author']/*[local-
name()='name']',
  author_email   VARCHAR(255)  PATH '*[local-name()='author']/*[local-
name()='email']' ) AS RES;
```

Entête personnalisé

TITLE	SUMMARY
Rappel : IBM - 13 et 14 mai 2014 - Evéneme...	Bonjour à toutes et à tous, Si vous n'êtes pas encore inscrit(...
KNOW400	Salutations J'ai mis à jour KNOW400 pour : - RPG format libre ...
la meme avce 2 corrections	Je suis decu par l'utilisation des tableaux Pour le moment c'e...
un petit grain sur les variables globales	Je vai essayé d'en faire un sur l'utilisation des tableaux A+ ...
Accès machine Christian	Christophe, j'ai des problèmes de stabilité de la connexion à ...
Révision de cours	Bonjour ma puce, pourrais-tu regarder le cours Utilisation de ...

AUTHOR_NAME	AUTHOR_EMAIL
Nathanael BONNET	nathanael.bonnet@gaia.fr
Nathanael BONNET	nathanael.bonnet@gaia.fr
Nathanael BONNET	nathanael.bonnet@gaia.fr
BERTHOIN Pierre-Lou.	pierre-louis.berthoin@prestataire.april.com
Christophe Nicolian	christophe.nicolian@wirtanen.net
Saad - Gaia	jmsaad@gaia.fr

HTTP et XML

■ Constat

- HTTP est le protocole de communication le plus répandu
- XML est le langage de représentation des données le plus répandu
- → Les Services Web SOAP utilisent ces deux éléments pour fournir des fonctionnalités distantes

■ DB2 dispose de HTTP et de XML

- Il est possible d'appeler par SQL des Services Web SOAP
- Jusqu'ici, il fallait utiliser
 - IWS Client
 - HTTPAPI
 - Java, PHP, ...
- Nécessite du code, très technique, ainsi que de la compilation

Service Web SOAP

- Les Services Web SOAP sont définis par une norme
 - Les messages échangés sont codés en langage SOAP, qui est un langage XML
 - Un certain nombre d'éléments sont normalisés pour l'interopérabilité
- Ils sont très répandus
 - Sur le net
 - Dans les communications entre les applications
 - Sur nos IBM i ...

ConvertTemp

- Un serveur de services web est fourni sur l'IBM i
 - Il contient un service web de test
 - Conversion de température
- Appel par un outil externe
 - Requête

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://converttemp.wsbeans.iseries/xsd">
  <soapenv:Header/>
  <soapenv:Body>
    <xsd:converttemp>
      <xsd:args0>
        <xsd:_TEMPIN>56</xsd:_TEMPIN>
      </xsd:args0>
    </xsd:converttemp>
  </soapenv:Body>
</soapenv:Envelope>
```

ConvertTemp

– Réponse

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:converttempResponse
      xmlns:ns="http://converttemp.wsbeans.iseries/xsd">
      <ns:return
        xsi:type="ns:CONVERTTEMPResult"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ns:_TEMPOUT>13.33</ns:_TEMPOUT>
      </ns:return>
    </ns:converttempResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

ConvertTemp

■ Par SQL

```
values ( SYSTOOLS.HTTPPOSTCLOB(
'http://10.3.1.1:10020/web/services/ConvertTemp.ConvertTempHttpSoap11Endpoint/' ,
'<httpHeader><header name="Accept" value="application/atom+xml"/></httpHeader>' ,
XMLSERIALIZE(
XMLDOCUMENT(
XMLELEMENT(
NAME "soapenv:Envelope",
XMLNAMESPACES( 'http://schemas.xmlsoap.org/soap/envelope/' AS "soapenv",
'http://converttemp.wsbeans.iseries/xsd' AS "xsd" ),
XMLELEMENT(
NAME "soapenv:Body",
XMLELEMENT(
NAME "xsd:converttemp",
XMLELEMENT(
NAME "xsd:args0",
XMLELEMENT(
NAME "xsd:_TEMPIN" , '45' ) ) ) ) ) ) AS CLOB(1M) ) ) ) ;
```

ConvertTemp

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:converttempResponse xmlns:ns="http://converttemp.wsbeans.iseries/xsd">
      <ns:return xsi:type="ns:CONVERTTEMPResult"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ns:_TEMPOUT>7.22</ns:_TEMPOUT>
      </ns:return>
    </ns:converttempResponse>
  </soapenv:Body>
</soapenv:Envelope>
```


Liste des aéroports

- Le site <http://www.webservices.net> fournit des web services gratuits
- Par exemple, la liste des aéroports par pays

The screenshot displays a SOAP request and response in a browser window. The address bar shows the URL `http://www.webservices.net/airport.asmx`. The request (left pane) is a SOAP envelope with a body containing a `GetAirportInformationByCountry` call with the parameter `FRANCE`. The response (right pane) is a SOAP envelope with a body containing a `GetAirportInformationByCountryResponse` with a `GetAirportInformationByCountryResult` that returns a table of airport data for France. The first row of the table is highlighted, showing details for LME (LE MANS ARNAGE).

```
Request 1
http://www.webservices.net/airport.asmx

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
2   <soapenv:Header/>
3   <soapenv:Body>
4     <web:GetAirportInformationByCountry>
5       <web:country>FRANCE</web:country>
6     </web:GetAirportInformationByCountry>
7   </soapenv:Body>
8 </soapenv:Envelope>

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   <soap:Body>
3     <GetAirportInformationByCountryResponse xmlns="http://www.webserviceX.NET">
4       <GetAirportInformationByCountryResult><![CDATA[<NewDataSet>
5         <Table>
6           <AirportCode>LME</AirportCode>
7           <CityOrAirportName>LE MANS ARNAGE</CityOrAirportName>
8           <Country>France</Country>
9           <CountryAbbrviation>FR</CountryAbbrviation>
10          <CountryCode>427</CountryCode>
11          <GMTOffset>-1</GMTOffset>
12          <RunwayLengthFeet>4626</RunwayLengthFeet>
13          <RunwayElevationFeet>194</RunwayElevationFeet>
14          <LatitudeDegree>48</LatitudeDegree>
15          <LatitudeMinute>0</LatitudeMinute>
16          <LatitudeSecond>0</LatitudeSecond>
17          <LatitudeNpeerS>N</LatitudeNpeerS>
18          <LongitudeDegree>1</LongitudeDegree>
19          <LongitudeMinute>15</LongitudeMinute>
20          <LongitudeSeconds>0</LongitudeSeconds>
21          <LongitudeEperW>E</LongitudeEperW>
22        </Table>
23      </GetAirportInformationByCountryResult>
24    </GetAirportInformationByCountryResponse>
25  </soap:Body>
26 </soap:Envelope>
```

Liste des aéroports

– Par SQL

```

with cte as (
select xmlparse( document SYSTOOLS.HTTPPOSTCLOB(
'http://www.webserviceX.net/airport.asmx' ,
'<httpHeader><header name="Accept-Encoding" value="gzip,deflate"/><header name="Content-Type" value="text/xml"/><header
name="action" value="http://www.webserviceX.NET/GetAirportInformationByCountry"/> </httpHeader>' ,
XMLSERIALIZE( XMLDOCUMENT( XMLELEMENT(NAME "soap:Envelope", XMLNAMESPACES( 'http://schemas.xmlsoap.org/soap/envelope/' AS
"soap", 'http://www.webserviceX.NET' AS "web" ),
XMLELEMENT(NAME "soap:Body", XMLELEMENT(NAME "web:GetAirportInformationByCountry", XMLELEMENT(NAME "web:country", 'FRANCE' ) )
) ) ) AS CLOB(1M) ) ) )
  as resp_
from sysibm.sysdummy1 ) ,

lst_airprt as (
  select list_airport
  from cte , XMLTABLE(
    '$result/*:Envelope/*:Body/*:GetAirportInformationByCountryResponse/*:GetAirportInformationByCountryResult'
    PASSING resp_ AS "result"
    COLUMNS list_airport clob(1M) PATH '.' ) as res ) ,

lst_airprt_xml as (
  select xmlparse( document list_airport ) as xml_airprt
  from lst_airprt )

select distinct res.nom_airport
from lst_airprt_xml ,
  XMLTABLE( '$result/NewDataSet/Table/CityOrAirportName'
    PASSING xml_airprt AS "result"
    COLUMNS nom_airport varchar( 64 ) PATH '.' ) as res

order by 1 ;

```

Liste des aéroports

■ Résultat

NOM_AIRPORT
AGEN LA GARENNE
AJACCIO CAMPO DELL
ALBI SEQUESTRE
ALPE D'HUEZ
ANGERS
ANGOULEME BEL AIR
ANNECY MEYTHE
AUBAGNE AGORA HLPD
AUBENAS VALS LANAS
AURILLAC TRONQUIERE
AVIGNON CAUMONT
BARCELONNETTE
BASTIA PORETTA
BEAUVAIS TILLE
BELFORT FONTAINE
BERGERAC ROUMANIERE
BEZIERS VIAS
BIARRITZ BAYON ANGL
BORDEAUX MERIGNAC
BOURGES
BREST GUIPAVAS
BRIVE-LA-GAILLARDE LAROCHE
CAEN CARPIQUET
CAHORS LALBENQUE
CALAIS MARCK
CALVI CATHERINE
CANNES CROISETTE
CANNES MANDELIEU
CARCASSONNE SALVAZA
CASTRES MAZAMET
CHAMBERY A.L. BAINS
CHATEAUROUX
CHERBOURG MAUPERTUS
CHOLET LE PONTREA
CLERMONT-FERRAND AULNAT
COGNAC
COLMAR HOUSSEN
COURCHEVEL

HTTP et JSON

- Aujourd'hui, de nombreux services web utilisent une notation JSON au lieu de XML
 - JavaScript Object Notation
 - <http://www.w3.org/TR/json-ld/>
 - <http://www.ietf.org/rfc/rfc4627.txt>
 - JSON sert généralement à encoder les valeurs de retour du service web

- JSON
 - Comme XML, information structurée en mode caractère
 - Ne comporte pas de balises (XML) mais des étiquettes

HTTP et JSON

■ Exemple

XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<clients nb="2">
  <client id="1">Client 1</client>
  <client id="2">
    <name>Client 2</name>
    <adresse>
      <voie>69 Rue Gorge de Loup</voie>
      <cp>69009</cp>
      <ville>LYON</ville>
    </adresse>
  </client>
</clients>
```

JSON

```
{
  "clients": {
    "-nb": "2",
    "client": [
      {
        "-id": "1",
        "#text": "Client 1"
      },
      {
        "-id": "2",
        "name": "Client 2",
        "adresse": {
          "voie": "69 Rue Gorge de Loup",
          "cp": "69009",
          "ville": "LYON"
        }
      }
    ]
  }
}
```

HTTP et JSON

- JSON est plus adapté aux traitements AJAX (Asynchronous JavaScript and XML)
 - Traitements asynchrones dans une page web permettant la mise à jour partielle de l'affichage sans action de l'utilisateur
 - Le langage est plus léger et moins formel que XML
 - Le nombre de caractères est comparable
- Aujourd'hui JSON non supporté par DB2 for i
 - Devrait bientôt arriver ? Disponible pour DB2 LUW ...

HTTP et JSON

■ Exemple

- Open Data du Grand Lyon
 - <http://smartdata.grandlyon.com/>
- SQL

values

```
SYSTOOLS.HTTPGETCLOB('https://download.data.grandlyon.com/ws/grandlyon/adr_voie_lieu.adrcommune/nomreduit.json?compact=true',  
'') ;
```

- Résultat

```
1 {  
2   "fields": ["nomreduit"],  
3   "nb_results": 59,  
4   "values": ["ST-PRIEST",  
5             "LISSIEU",  
6             "ST-GENIS-LES-O",  
7             "FLEURIEU",  
8             "CHARBONNIERES",  
9             "JONAGE",  
10            "CAILLOUX",  
11            "FONTAINES-ST-M",  
12            "ALBIGNY",  
13            "ST-DIDIER",  
14            "VILLEURBANNE",  
15            "LA-TOUR",  
16            "LIMONEST",  
17            "SATHONAY-V",  
18            "POLEYMIEUX".
```

Services Web REST

- REpresentational Stat Transfer
 - C'est un style d'architecture (ensemble de règles) et non un protocole (comme SOAP)
 - Architecture orientée ressource et non traitement
 - Ressource : tout ce qui peut être référencé par un lien

- Mécanisme
 - Séparation de l'interface et du stockage des données (évolutions)
 - Sans état (indépendance, load balancing)
 - Gestion du cache (réglable)
 - Interface uniforme (lisibilité des applications)
 - Verbes : GET , POST, PUT, DELETE
 - Hiérarchie définie dans l'URL
 - Libre

Services Web REST

■ Efficacité vs SOAP

- REST s'affranchit de couche d'abstraction, il est donc le plus efficace
- En syntaxe
- En performance
- Il n'est pas normé, et est donc plus souple (avantage ou non ?)
- De nombreux services web REST acceptent des données en entrée et en sortie au format XML et JSON
 - Le mode est généralement défini
 - Soit par l'URL
 - Soit par l'entête HTTP (Accept: application/xml ou Accept: application/json)

Services Web REST

■ Principe

- GET <http://masociete.fr/api/clients>
 - Retourne la liste des clients
- GET <http://masociete.fr/api/clients/25>
 - Retourne le client n° 25
 - On peut trouver GET <http://masociete.fr/api/clients?id=25>
- GET <http://masociete.fr/api/clients/fr/rhone>
 - Représentation de la hiérarchie
- DELETE <http://masociete.fr/api/clients/25>
 - Suppression du client n° 25

Services Web REST

■ Exemple

- <http://www.wunderground.com>
 - Propose des services web météo
- Nécessite un compte (gratuit pour test)
 - On récupère une clé de licence

GET YOUR API KEY

Analytics Key Settings Featured Applications Documentation Forums

Select a Key to Customize XXXXXXXXXX - Demo ▾

Edit API Key

Key ID

Project Name

Company Website

Contact Phone

Contact Email

[Create a New Key](#)

Regenerate API Key

Has your key been compromised? You can generate a new key.

Consequences

- You will need to change your apps to use the new key.
- Your statistics will be reset.
- This action cannot be undone.

I understand the consequences.

Billing Information

View your billing information:

- [Modify](#) • [Downgrade](#)
- [Upgrade](#) • [Cancel](#)

Customize a plan that suits your needs: TOTAL: \$0 USD per month

Services Web REST

■ Météo à Lyon

- Le service est documenté sur le site
- L'URL se construit de la façon suivante
 - <http://api.wunderground.com/api/{clé}/conditions/q/FR/Lyon.json>

– Appel

values

```
SYSTOOLS.HTTPGETCLOB('http://api.wunderground.com/api/licence/conditions/q/FR/Lyon.json' , '' ) ;
```

Services Web REST

– Résultat

```
{
  "response": {
    "version": "0.1",
    "termsOfService": "http://www.wunderground.com/weather/api/d/terms.html",
    "features": {
      "conditions": 1
    }
  },
  "current_observation": {
    "image": {
      "url": "http://icons-ak.wxug.com/graphics/wu2/logo_130x80.png",
      "title": "Weather Underground",
      "link": "http://www.wunderground.com"
    },
    "display_location": {
      "full": "Lyon, France",
      "city": "Lyon",
      "state": "",
      "state_name": "France",
      "country": "FR",
      "country_iso3166": "FR",
      "zip": "00000",
      "magic": "63",
      "wmo": "07480",
      "latitude": "45.75000000",
      "longitude": "4.84999990", ...
    }
  }
}
```

Conclusion

- Les possibilités HTTP de DB2
 - Permettent d'utiliser des ressources du SI ou d'internet qui ne sont pas stockées dans la BD
 - Sans code à développer et compiler
- La disponibilité de XML dans la base, conjointement à HTTP, permet l'utilisation de services web
 - Et l'utilisation directe dans une requête avec des données relationnelles
- La difficulté ne réside pas dans l'usage des fonctions
 - Mais dans la connaissance et la compréhension de HTTP, XML, des services web, SOAP, REST, des problématiques d'encodage, d'accès aux réseaux ...

Nous contacter

- Par mail

- nbonnet@gaia.fr
- contact@gaia.fr

- Nos sites

- www.gaia.fr
- www.know400.fr
- www.as400.fr