Xây dựng một ứng dụng pureXML và JSON, Phần 1: Lưu trữ và truy vấn JSON với pureXML của DB2

Chấp nhận một ánh xạ JSON-thành-XML đơn giản

Nuno Job, Chuyên gia DB2, IBM

Susan Malaika, Chuyên viên kỹ thuật cao cấp, IBM Japan

Michael Schenker, Kỹ sư phần mềm, IBM

Tóm tắt: JavaScript Object Notation (JSON), một ký pháp văn bản phổ biến trong Web 2.0, được dùng để biểu diễn các đối tượng (hoặc cấu trúc dữ liệu) dưới dạng văn bản tuần tự hóa khi các trình khách và trình chủ trao đổi thông tin với nhau. Một số ứng dụng lợi dụng các đối tượng JSON lâu bền để duy trì trạng thái qua các phiên làm việc. Trong bài viết này, chúng ta hãy tìm hiểu cách pureXML® DB2® có thể lưu trữ, quản lý, và truy vấn JSON khi bạn chấp nhận một ánh xạ JSON-thành-XML đơn giản.

Giới thiệu

JavaScript (được định nghĩa trong Đặc tả Ngôn ngữ ECMAScript trong ECMA) là một ngôn ngữ kịch bản lệnh lần đầu tiên được thực hiện trong Netscape Navigator (một trình duyệt Web) để tăng cường việc xử lý của các trang Web. JSON (được định nghĩa trong RFC 4.627 tại IETF) là một định dạng biểu diễn các cấu trúc dữ liệu JavaScript, chẳng hạn như các đối tượng và mảng, dưới dạng văn bản tuần tự hóa.

Các thuật ngữ thường sử dụng

- Ajax: JavaScript + XML không đồng bộ (Asynchronous JavaScript + XML)
- API: Giao diện lập trình ứng dụng (Application programming interface)
- DBMS: Hệ Quản trị Cơ sở dữ liệu (Database Management System)
- DOM: Mô hình đối tượng tài liệu (Document Object Model)
- HTTP: Giao thức truyền siêu văn bản (Hypertext Transfer Protocol)
- IETF: Đơn vị đặc trách kỹ thuật Internet (Internet Engineering Task Force)
- RFC: Đề xuất để xin góp ý (Request For Comments)
- RSS: Dịch vụ Cung cấp thông tin đơn giản (Really Simple Syndication)
- SAX: API đơn giản cho XML (Simple API for XML)
- SOA: Kiến trúc Hướng dịch vụ (Service Oriented Architecture)
- W3C: Hiệp hội World Wide Web (World Wide Web Consortium)
- XHTML: Ngôn ngữ đánh dấu siêu văn bản mở rộng (Extensible HyperText Markup Language)
- XML: Ngôn ngữ đánh dấu mở rộng được (Extensible Markup Language)
- XSLT: Các chuyển đổi ngôn ngữ phiếu định kiểu mở rộng được (Extensible Stylesheet Language Transformation)

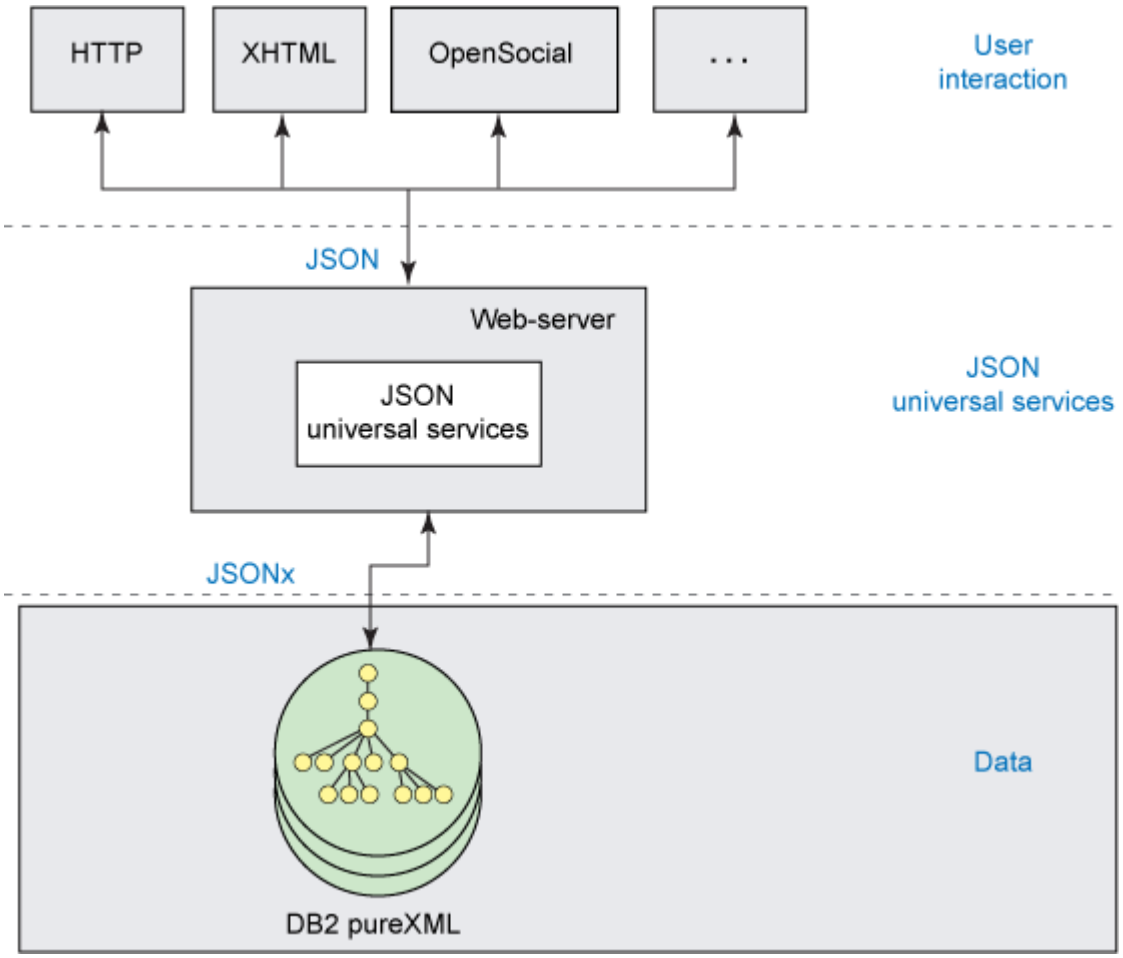
Trong khi XML (được định nghĩa trong đặc tả kỹ thuật XML 1.0 tại W3C) là một biểu diễn phổ biến để trao đổi thông báo giữa các trình chủ và trình khách được ghép lỏng, JSON thường được sử dụng để đạt được cùng một hiệu quả. Một trong những lý do để chấp nhận JSON là ở chỗ các đối tượng JSON dễ dàng được thao tác hơn bằng cách sử dụng các ngôn ngữ kịch bản như JavaScript, Python, hoặc Ruby.

Ý tưởng về việc lưu trữ và truy vấn XML trong các cơ sở dữ liệu đã nảy ra sau khi việc sử dụng XML để trao đổi dữ liệu trở nên phổ biến. Tương tự như vậy, thao tác và trao đổi JSON đã trở nên phổ biến, nhưng lưu trữ thì chưa. Tuy nhiên, các hệ quản trị cơ sở dữ liệu hướng tài liệu JSON đã bắt đầu xuất hiện. Thí dụ, Apache CouchDB được tuân thủ JSON chặt chẽ thông qua các giao diện và cách lưu trữ của nó

Trong bài này, chúng tôi giới thiệu khái niệm về sự trao đổi JSON như là XML để tận dụng các bộ xử lý XML, thiết bị XML, lưu trữ XML (thường có trong các cơ sở dữ liệu XML như pureXML DB2), và các công nghệ XML khác chẳng hạn như XQuery và XSLT. Để đạt được điều đó, chúng tôi sẽ giới thiệu một khuôn dạng XML có tên là JSONx, mô tả một ký pháp JSON-thành-XML thân thiện, và giải thích những sự khác nhau giữa JSONx và các khuôn dạng thân thiện đó.

Bằng cách làm theo các bước trong bài viết và phần tải về kèm theo này, bạn có thể xây dựng nên một kho lưu giữ JSON có chỉ mục và có thể truy vấn được dựa trên cơ sở dữ liệu mẫu pureXML DB2. Bài này là bài viết đầu tiên trong loạt ba bài minh họa cách xây dựng một ứng dụng pureXML ba tầng dựa trên JSON bằng cách kết hợp JSON, các dịch vụ web, và các tiểu trình OpenSocial (OpenSocial gadgets).

. Tổng quan về kiến trúc Các dịch vụ Phổ dụng (Universal Services)



Các dịch vụ phổ dụng là một tập hợp đơn giản nhưng cố định các hoạt động (các phép toán) cơ sở dữ liệu cho phép truy vấn và sửa đổi dữ liệu XML, được lưu trong một cột pureXML của một cơ sở dữ liệu DB2. Các phép toán cơ sở dữ liệu này cho phép bạn chèn vào, cập nhật, xóa, và truy vấn dữ liệu, được trưng ra như các dịch vụ web thông qua cơ chế dịch vụ web dữ liệu. Xem phần Tài nguyên để biết thêm thông tin về Các dịch vụ phổ dụng dùng cho pureXML. Các dịch vụ phổ dụng JSON cho pureXML cũng đưa ra cùng các phép toán cơ sở dữ liệu đó cho trình khách, tuy nhiên bằng JSON chứ không phải bằng XML, trong khi tiếp tục làm việc với XML bên phía trình chủ. Ứng dụng khách không biết được rằng JSON đến và đi được chuyển thành XML ở phía trình chủ trong cơ sở dữ liệu.

Các bài khác trong loạt bài này

- Phần 2: Tạo Các Dịch vụ Phổ dụng cho pureXML để đưa ra JSON
- Phần 3: Tạo các tiểu trình OpenSocial cho pureXML

Bài viết cũng khảo sát các lựa chọn mà bạn có thể thực hiện để biểu diễn JSON như là XML và sau đó tiếp tục mô tả một số kịch bản và cách áp dụng pureXML DB2 của IBM trong những kịch bản như vậy. Bài viết kèm theo một phần tải về (tên là *JSONx bundle* - gói JSONx) mà bạn có thể sử dụng để xây dựng một cơ sở dữ liệu pureXML JSONx mẫu. Nó có thể tạo nên một nền tảng cho các ứng dụng JSONx. Trong phần tải về có hai hàm DB2 do người sử dụng định nghĩa, chúng chuyển đổi qua lại giữa JSON và JSONx.

JSON

JSON là một khuôn dạng dựa văn bản, con người có thể đọc được, dùng để trao đổi dữ liệu giữa các trình khách và trình chủ. Nó cung cấp cho các nhà phát triển một khuôn dạng trao đổi, ánh xạ trực tiếp đến các cấu trúc dữ liệu mà họ sử dụng. Để có hiệu quả đó, JSON định nghĩa các cấu trúc dữ liệu chính sau đây: số, chuỗi ký tự, logic (đúng và sai), mảng (một dãy có trình tự các giá trị), đối tượng (tập hợp các cặp giá trị khoá), và rỗng (null).

Liệt kê 1 minh hoạ một đối tượng JSON mô tả một khách hàng. Lồng bên trong đối tượng customerinfo là hai đối tượng để xác định mã nhận dạng khách hàng (cid) và tên. Đối tượng customerinfo cũng chứa hai đối tượng có cấu trúc để xác định địa chỉ và số điện thoại khách hàng.

Liệt kê 1. Thông tin khách hàng Kathy Smith trong JSON

```
{
  "customerinfo" : {
        "cid" : 1000 ,
        "name" : "Kathy Smith" ,
        "addr" : {
          "country"    : "Canada" ,
          "street"     : "5 Rosewood" ,
          "city"       : "Toronto" ,
          "prov-state" : "Ontario" ,
          "pcode-zip"  : "M6W 1E6"
        } ,
        "phone" : {
          "work" : "416-555-1358"
        }
  }
}
```

Một kịch bản JSON điển hình

Một ca sử dụng điển hình dành cho JSON là có một ứng dụng Web hoán đổi dữ liệu cho nhau với một API chẳng hạn như Các Dịch vụ Web Yahoo hoặc API Twitter. Trong kịch bản này, một ứng dụng Web sử dụng các yêu cầu JavaScript không đồng bộ (Ajax) để trao đổi thông tin JSON với dịch vụ Web đã đưa ra API đó.

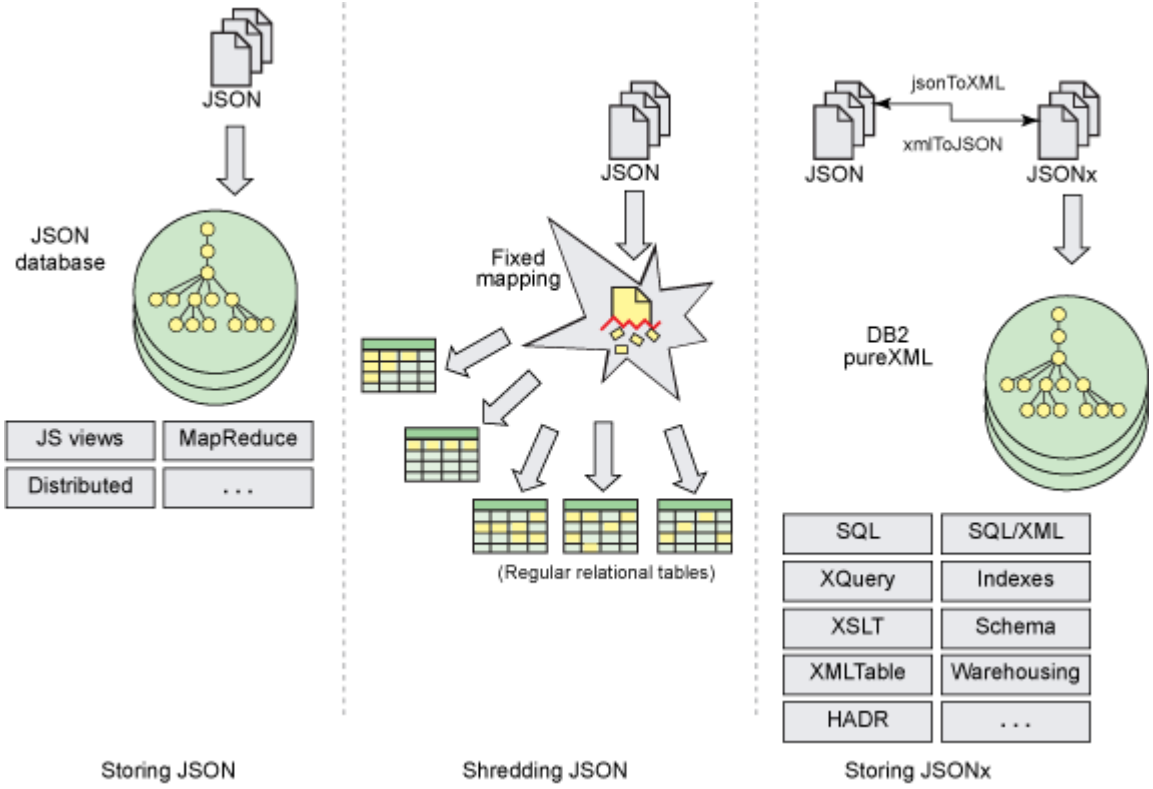
Thường thì một API như vậy cho phép ứng dụng chọn ra khuôn dạng hoán đổi. Các khuôn dạng được hỗ trợ thông dụng gồm XML, các chuẩn XML được định nghĩa trước chẳng hạn như RSS và Atom, và JSON. Để tìm thí dụ về các khuôn dạng như vậy xin tham khảo Phụ lục A.

Việc tự do lựa chọn khuôn dạng dùng để giao tiếp giữa ứng dụng và dịch vụ Web cho phép các nhà phát triển tăng tốc quy trình phát triển. Tuy nhiên, điều này phát sinh ra các câu hỏi liên quan đến việc duy trì một cơ sở hạ tầng hỗ trợ đồng thời nhiều khuôn dạng. Danh sách dưới đây bao gồm một số tùy chọn để lưu trữ dữ liệu:

- Sử dụng một cơ sở dữ liệu JSON hướng tài liệu (document-centric) chẳng hạn như Apache CouchDB.
- Sử dụng một cơ sở dữ liệu quan hệ bằng cách chia nhỏ và tái xây dựng JSON cho từng yêu cầu.
- Sử dụng lưu trữ XML nguyên sinh (lưu trữ JSONx) và cung cấp các giao diện để đưa ra và xử lý JSON.

Hình 2 minh họa các thí dụ cho các tùy chọn lưu trữ dữ liệu này. (Xem Hình 2 lớn hơn.)

Hình 2. Các cách khác nhau để lưu trữ JSON



Tất cả các cách tiếp cận này đều có những điểm mạnh và điểm yếu. Bài này không phân tích các khác biệt đó mà tập trung vào một kịch bản, trong đó việc sử dụng pureXML là cách tiếp cận thuận tiện nhất đối với một ứng dụng cụ thể. Một số lý do có thể làm cho XML trở thành cách tiếp cận thuận tiện nhất gồm:

- Phần còn lại của cơ sở hạ tầng đã sử dụng XML và SOA từ trước
- Sự tồn tại của các công cụ XML (chẳng hạn như Websphere® DataPower® của IBM) và các công nghệ XML có thể mở rộng được để làm việc với JSONx

Đối chiếu JSON với XML

Khi các nhà phát triển sử dụng XML để trao đổi, họ thường dẫn hướng trong XML thông qua DOM XML hay SAX. Khả năng làm việc với cùng một cấu trúc dữ liệu JSON để thao tác và để trao đổi sẽ làm đơn giản quy trình phát triển bằng việc cung cấp cách truy cập qua khai báo tới các đối tượng trong ngôn ngữ lập trình chủ của ứng dụng.

Việc tập trung vào XML là để cung cấp một ký pháp trao đổi tự định nghĩa, có thể kết hợp nếu muốn với một lược đồ nghiêm ngặt mà chính lược đồ này có thể trao đổi được. XML cũng cung cấp một loạt các đặc tính, chẳng hạn như vùng tên, nén, chữ ký số, và an ninh, cùng với các ngôn ngữ khai báo chẳng hạn như XQuery và XSLT mà có thể triệu gọi từ các ngôn ngữ lập trình C, Java™, và Ruby để thao tác một hoặc nhiều tài liệu XML.

JSON không có nhiều đặc tính mà XML có.

Biểu diễn JSON bằng XML

Tiêu điểm của bài viết này là để chỉ ra cách làm thế nào để tạo ra một khuôn dạng XML đẳng cấu (có cấu trúc giống hệt) với bất kỳ tài liệu JSON nào, nói một cách khác là cách làm thế nào để tạo ra một ánh xạ phổ quát giữa bất kỳ tài liệu JSON và tài liệu XML nào.

Để đạt được một ánh xạ hiệu quả giữa bất kỳ tài liệu JSON và XML nào, bạn phải xem xét sự khác nhau giữa hai khuôn dạng. Phần này khảo sát một sự ánh xạ khá trực quan nhưng không đẳng cấu giữa XML và JSON, sao cho bạn có thể hiểu được những khác biệt như vậy. Bảng 1 mô tả một thí dụ chưa đầy đủ của ánh xạ như vậy.

Bảng 1. Lập ánh xạ khả dĩ đối với XML thân thiện

Mẫu	JSON	XML	Mô tả
1	{"foo" : "bar"}	<foo> bar </foo>	Đối tượng với kiểu giá trị chuỗi ký tự
2	{"foo" : true}	<foo> <true/> </foo>	Đối tượng với kiểu giá trị true (đúng)
3	{"foo" : { "true" : null }}	<foo> <true/> </foo>	Đối tượng với một Đối tượng lồng bên trong kiểu giá trị null (rỗng)
4	{"foo bar!" : true}	Error: "foo bar!" is not a valid QName	Đối tượng với kiểu giá trị false (sai) không chuyển đổi thành công vì khoá chính chứa các ký tự không hợp lệ đối với một QName XML (thí dụ dấu cách và dấu chấm than)
5	{"foo" : null}	<foo/>	Đối tượng với kiểu giá trị null (rỗng)
6	{"foo": { "bar" : null}}	<foo> <bar/> </foo>	Các đối tượng lồng nhau với nút lá có kiểu null (rỗng)

7	{"foo": { "bar" : [null, false]}}	<foo> <bar> <null/> <false/></bar></foo>	Đối tượng với mảng lồng bên trong

Chúng tôi đề cập đến một ký pháp ánh xạ sử dụng các tên phần tử và tên thuộc tính đặc thù cho ứng dụng, chứ không phải các tên JSONx chính tắc, giống như khuôn dạng *thân thiện* của JSON. Nhiều thiếu sót vẫn còn trong cách ánh xạ này. Các Mẫu 2 và 3 có cấu trúc khác nhau và đã được tuần tự hóa thành thành tài liệu XML như nhau, làm cho không thể xây dựng lại thành chính tài liệu JSON đã có từ XML đã được tạo ra. Khoá đối tượng của Mẫu 4 chứa ký tự không hợp lệ đối với một QName XML. Một số kỹ thuật có thể được sử dụng để vượt qua các hạn chế này, tuy nhiên chúng sẽ không thể hiện cùng một sức mạnh như một ký pháp đẳng cấu chẳng hạn như JSONx.

JSONx

JSON XML (JSONx) chính tắc được đưa vào như là một khuôn dạng đẳng cấu đối với JSON. Do đó các vấn đề đã ghi trong phần trước với các ký pháp thân thiện không xảy ra trong JSONx.

Trong Liệt kê 2, bạn có thể xem các thông tin khách hàng về Kathy Smith, trước đây được biểu diễn bằng JSON trong Liệt kê 1, dưới dạng JSONx.

Liệt kê 2. Thông tin khách hàng về Kathy Smith bằng JSON XML chính tắc (JSONx)

```
<json:object xmlns:json="http://www.ibm.com/xmlns/prod/2009/jsonx">
  <json:object name="customerinfo">
        <json:number name="cid">1000</json:number>
        <json:string name="name">Kathy Smith</json:string>
        <json:object name="addr">
          <json:string name="country">Canada</json:string>
          <json:string name="street">5 Rosewood</json:string>
          <json:string name="city">Toronto</json:string>
          <json:string name="prov-state">Ontario</json:string>
          <json:string name="pcode-zip">M6W 1E6</json:string>
        </json:object>
        <json:object name="phone">
          <json:string name="work">416-555-1358</json:string>
```

```
        </json:object>
  </json:object>
</json:object>
```

JSONx là một khuôn dạng thích hợp dùng cho các hệ thống xử lý dữ liệu XML nhưng cần được mở rộng với sự hỗ trợ cho JSON. Tuy nhiên, nó không là một giải pháp một-kích-cỡ-vừa-với-tất-cả để chuyển đổi JSON thành XML. Trong thí dụ trước, người ta có thể chỉ rõ các thông tin ban đầu về Kathy Smith với một khuôn dạng thân thiện như sau:

Liệt kê 3. Thông tin khách hàng về Kathy Smith với một khuôn dạng XML thân thiện khả dĩ

```
<customerinfo>
  <cid>1000</cid>
  <name>Kathy Smith</name>
  <addr>
        <country>Canada</country>
        <street>5 Rosewood</street>
```

```
        <city>Toronto</city>
        <prov-state>Ontario</prov-state>
        <pcode-zip>M6W 1E6</pcode-zip>
  </addr>
  <phone>
        <work>416-555-1358</work>
  </phone>
</customerinfo>
```

Việc sử dụng JSONx chứ không phải khuôn dạng thân thiện thúc đẩy các khả năng tái sử dụng mã, sử dụng công cụ phổ biến, chuyển đổi, và lọc mà thường gắn liền với việc sử dụng một tiêu chuẩn. Người ta vẫn có thể chọn tạo ra một khung nhìn trưng ra tài liệu với khuôn dạng khác thân thiện hơn bằng cách sử dụng các công nghệ XML chẳng hạn như XSLT và XQuery.

JSONx cho phép việc sử dụng cơ sở hạ tầng XML hiện có mà không phải tùy biến giải pháp để xử lý và lưu trữ JSON.

Gói JSONx

Phần này sẽ cung cấp một tổng quan ngắn gọn về một gói JSONx (một tập hợp các kịch bản lệnh và mã) dùng cho pureXML DB2, nó chỉ cho bạn cách:

- Chuyển đổi JSON thành JSONx
- Chuyển đổi JSONx thành JSON
- Nhập khẩu các tài liệu JSON như là JSONx
- Lưu trữ JSONx
- Lập chỉ mục JSONx để đạt được hiệu năng
- Trưng ra JSONx với một khuôn dạng quan hệ
- Chuyển đổi JSONx thành một ký pháp XML thân thiện
- Nối các tài liệu JSONx (dùng các vị từ quan hệ, nếu cần thiết) bằng cách sử dụng XMLQuery

Để có giải thích chi tiết hơn về những thứ trong gói JSONx, xin xem tệp tin readme kèm trong gói này.

Các điều kiện tiên quyết

Để cài đặt và chạy gói JSONx, cần phải có DB2 phiên bản v9.5 hoặc mới hơn đã cài đặt. Bạn có thể tải về DB2 Express-C, ấn bản mở của DB2 có gộp cả pureXML, làm cho nó thành một máy chủ dữ liệu XML và quan hệ, có đầy đủ chức năng (xem phần Tài nguyên để có liên kết tới địa chỉ tải về).

Dù gói này có thể chạy trong các hệ điều hành khác, nó đã được kiểm thử với Ubuntu Jaunty Jackalope 9.04 và Microsoft® Windows® XP SP2.

Cấu trúc bảng

Trong phần trước, bạn đã được giới thiệu thông tin khách hàng của Kathy Smith, bằng cả JSON và JSONx. Trong Bảng 2, bạn nhìn thấy bảng khách hàng, nơi thông tin được lưu trữ:

Bảng 2: Bảng JSONXMLADMIN.CUSTOMER

Tên cột	Kiểu dữ liệu	Mô tả
CID	INTEGER	Mã nhận dạng khách hàng
INFO	XML	Một bản ghi chứa thông tin cá nhân về khách hàng
COMMENT	VARCHAR(256)	Một mã nhận dạng ngắn, kiểu văn bản liên quan đến khách hàng

Hai bảng khác có trong gói JSONx: Product (Sản phẩm) và PurchaseOrder (Đơn đặt hàng).

Bảng sản phẩm chứa thông tin sản phẩm trong một cột XML tên là DESCRIPTION (Mô tả), cũng như một mã nhận dạng duy nhất cho sản phẩm đó:

Bảng 3: Bảng JSONXMLADMIN.PRODUCT

Tên cột	Kiểu dữ liệu	Mô tả
PID	INTEGER	Mã nhận dạng sản phẩm
Mô tả	XML	Một bản ghi chứa thông tin cá thể về sản phẩm

PurchaseOrder mô tả một giao dịch khi một khách hàng mua một hay nhiều sản phẩm. Một chi tiết thú vị về bảng này là ở chỗ mã nhận dạng khách hàng không chứa trong tệp tin gốc JSON mà nằm trong một cột quan hệ riêng. Bằng cách sử dụng JSONx và pureXML DB2, bạn có thể nối dữ liệu từ hai (hoặc nhiều) tài liệu JSON khác biệt nhau, và cũng có thể nối các bản ghi quan hệ với các tài liệu JSON. Để có một thí dụ cụ thể về cách thực hiện một phép nối như vậy, xin xem mã thủ tục chứa trong gói JSONx này.

Bảng 4: Bảng JSONXMLADMIN.PURCHASEORDER

Tên cột	Kiểu dữ liệu	Mô tả
POID	INTEGER	Mã nhận dạng đơn đặt hàng
CUSTID	INTEGER	Mã nhận dạng khách hàng liên đới
PORDER	XML	Một bản ghi chứa thông tin về đơn đặt hàng

Chuyển đổi JSON thành JSONx bằng các hàm Java do người sử dụng định nghĩa

Gói JSONx sẽ đăng ký hai hàm Java do người sử dụng định nghĩa (UDFs) trong DB2 cho phép chuyển đổi JSON thành JSONx và ngược lại. Liệt kê 4 minh họa một lệnh triệu gọi rất đơn giản, gọi hàm do người sử dụng định nghĩa để chuyển đổi JSON thành JSONx.

Liệt kê 4. Gọi hàm Java JSONTOXML do người sử dụng định nghĩa

```
SELECT JSONXML.JSONTOXML('{"foo": "bar"}') FROM
SYSIBM.SYSDUMMY1
```

Liệt kê 5 cho thấy kết quả từ lệnh SELECT trong Liệt kê 4.

Liệt kê 5. Kết quả tạo ra từ lệnh gọi hàm trong Liệt kê 4

```
<json:object xmlns:json="http://www.ibm.com/xmlns/prod/2009/jsonx">
 <json:string name="foo"> bar </json:string>
</json:object>
```

Để chuyển đổi JSONx thành JSON, hãy gọi thủ tục lưu sẵn XMLTOJSON. Liệt kê 6 cho thấy một thí dụ khi cùng một tài liệu JSON được sử dụng trong Liệt kê 4 được tuần tự hóa thành JSONx và sau đó đổi trở lại JSON.

Liệt kê 6. JSON thành JSONx và sau đó trở về JSON

```
SELECT JSONXML.XMLTOJSON(JSONXML.JSONTOXML('{"foo": "bar"}'))
FROM SYSIBM.SYSDUMMY1
```

Nối các tài liệu JSONx

Một trong những tính năng thú vị có trong pureXML DB2 là sự đa dạng của các tuỳ chọn để ghép nối dữ liệu. Việc lưu trữ JSONx trong pureXML DB2 có nghĩa là bây giờ bạn có thể nối các dữ liệu JSON của bạn chính xác như bạn đã thực hiện với các tài liệu XML của bạn. Liệt kê 7 cho thấy cách làm thế nào sử dụng một phép nối với cả vị từ quan hệ và vị từ XML và trả về, đối với từng Đơn Đặt hàng, POID (mã nhận dạng đơn đặt hàng), và tên khách hàng.

Liệt kê 7. Nối JSONx bằng cả vị từ quan hệ lẫn vị từ XML

```
SELECT
```

```
  POID,
  XMLCAST (
          XMLQUERY(
           "declare default element namespace
"http://www.ibm.com/xmlns/prod/2009/jsonx";
           data(

          $INFO/object/object[@name="customerinfo"]/string[@name="name"]/text()
)"
           ) AS VARCHAR(32)
          ) AS CUSTNAME
FROM
  JSONXMLADMIN.PURCHASEORDER ,
  JSONXMLADMIN.CUSTOMER
WHERE
  XMLEXISTS(
          "declare default element namespace
"http://www.ibm.com/xmlns/prod/2009/jsonx";
          $INFO/object/object[@name="customerinfo"]/number[@name="cid"][. =
```

```
$CUSTID]"

 )
```

DB2 9.7

Gói JSONx cũng minh họa một số bổ sung mới nhất trong pureXML của DB2 v9.7. Khi tạo ra khung nhìn về một tài liệu JSONx với XMLTABLE, DB2 v9.7 sẽ chọn ra các chỉ mục XML áp dụng được, như vậy sẽ làm cho truy vấn nhanh hơn nhiều so với các phiên bản trước đây.

Liệt kê 8. Trưng ra JSON dưới dạng một khung nhìn quan hệ với XMLTable

```
CREATE VIEW JSONXMLADMIN.RELCUSTOMER(CID, NAME, PHONE)
AS

 SELECT  *

 FROM

  XMLTABLE(

   XMLNAMESPACES (DEFAULT
'http://www.ibm.com/xmlns/prod/2009/jsonx'),

   'db2-fn:xmlcolumn("JSONXMLADMIN.CUSTOMER.INFO")'
```

```
    COLUMNS
        "CID" INTEGER
        PATH
'/object/object[@name="customerinfo"]/number[@name="cid"]/xs:double(.)' ,
        "NAME" VARCHAR(32)
        PATH
'/object/object[@name="customerinfo"]/string[@name="name"]/text()' ,
        "PHONE" VARCHAR(32)
        PATH
'/object/object[@name="customerinfo"]/object[@name="phone"][1]/*[1]/text()'
    )
```

Cũng có thể tạo ra JSONx từ dữ liệu quan hệ bằng cách sử dụng các hàm xuất bản SQL/XML.

Một đặc tính mới nữa là sự hỗ trợ của kiểu dữ liệu XML trong các hàm do người sử dụng định nghĩa cho phép thao tác các tài liệu XML. Khi dùng DB2 v9.7 để chạy gói JSONx, bạn sử dụng hai đặc tính mới khác để tạo ra kết quả: lập chỉ mục trên khung nhìn và một hàm XML do người sử dụng định nghĩa.

Hàm do người dùng định nghĩa có trong gói JSONx sử dụng tiện ích Cập nhật XQuery (XQuery Update Facility) để chuyển đổi JSONx thành một khuôn dạng đủ để trình bày một tập con cụ thể của JSON mà sẽ được nhập khẩu vào khi việc

thực hiện cơ sở dữ liệu mẫu của JSONx hoàn tất. Mặc dù đây không phải là một thuật toán chuyển đổi tổng quát hoặc có hiệu quả, Liệt kê 9 cho thấy cách làm thế nào có thể lập trình một cách có hiệu quả các hàm XML do người sử dụng định nghĩa bằng cách sử dụng pureXML DB2:

Liệt kê 9. Ví dụ hàm XML do người dùng định nghĩa dùng để tạo ra XML thân thiện

```
CREATE FUNCTION JSONXMLADMIN.JSONXTOFRIENDLY(JSONX
XML)

RETURNS XML

BEGIN ATOMIC

  RETURN XMLQUERY('

        declare namespace json="http://www.ibm.com/xmlns/prod/2009/jsonx";

        copy $n := $JSONX

        modify(

          for $e in $n//json:*

          where $e/@name

          return (
```

```
                    do rename $e as replace($e/@name," ", "_") ,

                    do delete $e/@name

                )

            )

            return document { $n/json:object/* }

  ');

END
```

Liệt kê 10 cho thấy cách chuyển đổi một thông tin khách hàng đã được viết bằng ký pháp thân thiện trước đó trở lại thành JSONx.

Liệt kê 10. Chuyển đổi thông tin khách hàng Kathy Smith dạng thân thiện trở lại thành JSONx

```
SELECT

  XMLQUERY('

            declare namespace json="http://www.ibm.com/xmlns/prod/2009/jsonx";
```

```
        copy $n := $friendly
        modify(
          for $e in $n//*
          let $name := local-name($e)
          let $type :=
                  if ($name = ("addr", "phone", "customerinfo")) then
"json:object"
                  else if ($name eq "cid") then "json:number"
                  else "json:string"
          return (
                  do rename $e as $type,
                  do insert attribute name { $name } into $e
          )
        )
        return document {
          <json:object>
                  { $n }
```

```
            </json:object>

        }

' PASSING JSONXMLADMIN.JSONXTOFRIENDLY(INFO) as "friendly")

FROM

        JSONXMLADMIN.CUSTOMER

WHERE

        CID = 1000
```

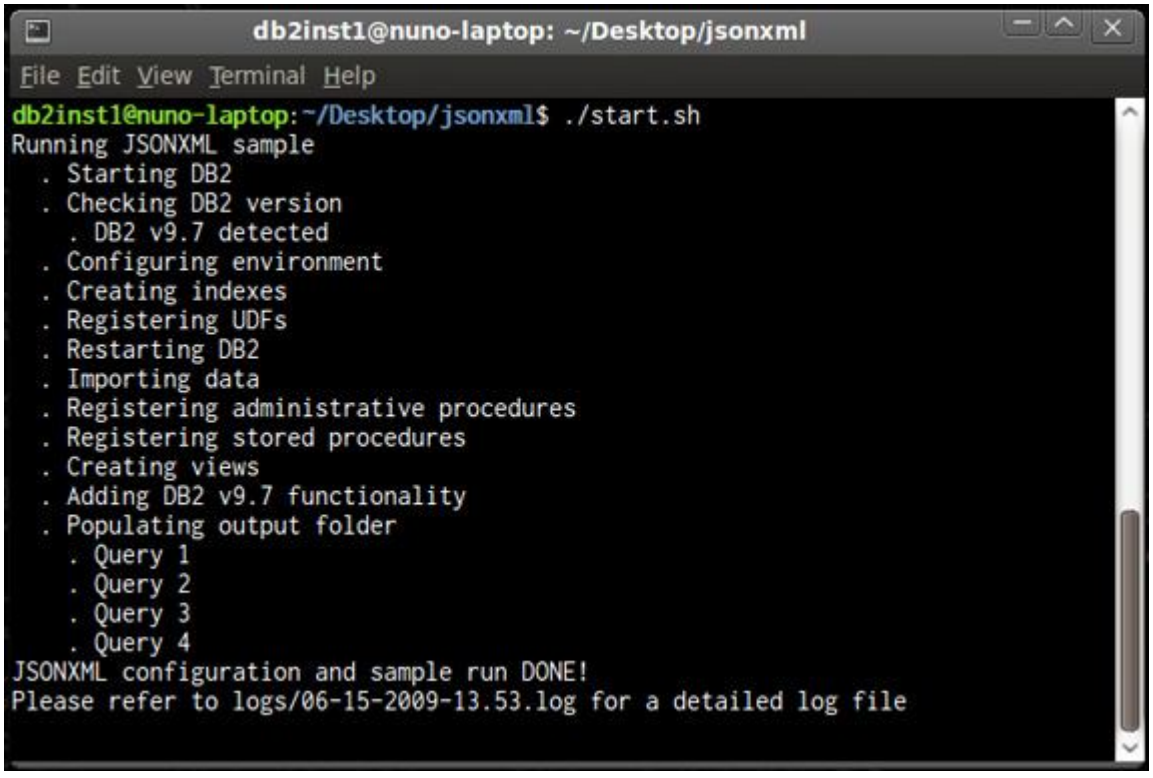
Chạy gói JSONx

Làm theo các bước liệt kê dưới đây để cài đặt gói JSONx:

- Để sử dụng nó trên nền Window, chỉ cần giải nén tệp tin jsonx.zip có sẵn ở phần tài nguyên trong hệ thống tệp tin của bạn. Trên các nền Linux®, giải nén tệp tin jsonx.zip bằng cách sử dụng lệnh **unzip -a -aa jsonx.zip.** Việc này sẽ đảm bảo rằng các ký tự kết thúc dòng đúng đắn được sử dụng.
- Trên các hệ thống Windows, hãy chắc chắn rằng môi trường xử lý dòng lệnh DB2 được khởi tạo. Trong các hệ thống dựa trên Linux, xin kiểm tra lại việc đăng nhập của bạn như một người sử dụng có quyền truy nhập DB2 (db2inst1 là người sử dụng DB2 mặc định).
- Bạn đã sẵn sàng chạy các kịch bản lệnh. (CHÚ Ý: Phải biết rằng các kịch bản lệnh khởi phát có thể lập cấu hình một số thông số hệ quản trị cơ sở dữ

liệu và nó có thể NGỪNG và KHỞI ĐỘNG LẠI hệ quản trị cơ sở dữ liệu để đảm bảo rằng hệ quản trị cơ sở dữ liệu có thể xử lý thủ tục lưu sẵn kiểu jar (jar-stored procedure). Nếu bạn còn phân vân về các kịch bản lệnh xin xem lại chúng trước khi chạy!). Chạy start.bat trên các nền Window. Trên các nền Linux, sử dụng start.sh. Sau khi kịch bản lệnh này hoàn tất, bạn sẽ thấy một kết quả tương tự như Hình 3 trong dấu nhắc lệnh hoặc shell bash của bạn. (Xem bản bằng chữ của kết quả ví dụ trong Hình 3.)

Hình 3. Kết quả ví dụ của gói JSONx



Lúc này mọi thứ đều được cài đặt và bạn có thể quan sát các kết quả truy vấn trong thư mục đầu ra. Để kiểm tra xem các kết quả được tạo ra có chính xác không, bạn có thể so sánh thư mục đầu ra với đầu ra tham khảo nằm trong thư mục tham khảo.

Nếu các mẫu không phù hợp với thư mục tham khảo, hãy tra cứu tệp nhật ký (log file) nằm trong thư mục nhật ký (logs directory) để tìm ra cái có thể đã gây ra thất bại khi thi hành.

Để có minh hoạ về các tweet (bài đăng) của các Twitter bằng XML thân thiện và bằng JSONx, xin xem Phụ lục A.

Kết luận

Trong bài này, một khuôn dạng mới (JSONx) đã được giới thiệu, cho phép lưu trữ JSON như là XML trong pureXML DB2. Sự khác nhau giữa JSONx và các khuôn dạng thân thiện khác đã được phác thảo ra. Một trường hợp sử dụng điển hình của JSON đã được bàn luận và các lợi ích của việc sử dụng pureXML để lưu trữ JSON được làm nổi bật. Cuối cùng, với một bản tải về JSONx (gói JSONx), bạn giờ đây có một nền để xây dựng các ứng dụng JSON có khả năng pureXML.

Các bài khác trong loạt bài này

- Phần 2: Tạo các dịch vụ phổ dụng cho pureXML để đưa ra JSON
- Phần 3: Tạo các tiểu trình OpenSocial cho pureXML

Hai bài viết tiếp theo trong loạt bài này sẽ tập trung vào việc trưng ra cơ sở dữ liệu mẫu JSONx (được tạo ra trong gói JSONx) thông qua Các Dịch vụ Phổ dụng JSON và sau đó tập trung vào việc tạo ra tầng trình bày với các tiểu trình OpenSocial, dựa vào Các Dịch vụ Phổ dụng JSON như là một nền phụ trợ (back-end).

Một khả năng trong tương lai là xem xét các cách để cho các nhà phát triển Web truy cập các tài liệu JSON (được lưu như là JSONx) bằng JSONPath.

Lời Cảm ơn

Chúng tôi xin cảm ơn Brien Muschett và William Palma về các đóng góp của họ.

Phụ lục A

Như đã tham khảo trong Một kịch bản JSON điển hình, nhiều dịch vụ Web cung cấp rất nhiều khuôn dạng để hiển thị thông tin. Các chuẩn công nghiệp chẳng hạn như Atom và RSS, JSON, hoặc kể cả các ký pháp XML thân thiện là các thí dụ về những khuôn dạng như vậy. Twitter đưa ra một API để truy cập tweet bằng ID. Trong phụ lục này, chúng tôi trình bày các kết quả của việc phát ra hai yêu cầu đến Twitter bằng cách sử dụng curl, một dưới dạng JSON và một dưới dạng XML thân thiện. Sau đó, chúng tôi biểu diễn cách chuyển đổi XML thân thiện này thành JSONx.

Trong Liệt kê 11, chúng tôi sử dụng curl để yêu cầu thông tin về một thông báo cá nhân (tweet có mã nhận dạng là 2311383114) mà đã được đăng lên twitter.com. Để lấy ra thông tin tweet (*trạng thái* của nó), bạn có thể gọi curl từ thiết bị đầu cuối của bạn như trong Liệt kê 11, và trạng thái của tweet được trả về dưới khuôn dạng JSON. (Chú ý: Lệnh triệu gọi curl trong các Liệt kê 11và 12 được tách thành hai dòng nhằm mục đích định dạng.)

Liệt kê 11. Lấy ra trạng thái với mã nhận dạng ID 2311383114 ở khuôn dạng JSON

purexml@watson.ibm.com:~$ curl

http://purexmltest:3ce3ac99@twitter.com/statuses/show.xml?id=2311383114

{

"text": "Hello World!",

"in_reply_to_user_id": null,

"user": {

"following": null,

"favourites_count": 0,

"profile_sidebar_fill_color": "e0ff92",

"description": null,

"verified": false,

"utc_offset": null,

"statuses_count": 1,

"profile_sidebar_border_color": "87bc44",

"followers_count": 0,

"created_at": "Wed Jun 24 14:18:32 +0000 2009",

"url": null,

"name": "pureXML TEST",

"friends_count": 0,

"profile_text_color": "000000",

"protected": false,

"profile_image_url": "http:\/\/s3.amazonaws.com\/twitter_production

\/profile_images\/280225879\/twitterxml-1_bigger_normal.png",

"profile_background_image_url": "http:\/\/static.twitter.com

\/images\/themes\/theme1\/bg.gif",

"notifications": null,

"time_zone": null,

"profile_link_color": "0000ff",

"screen_name": "purexmltest",

```
        "profile_background_tile": false,
        "profile_background_color": "9ae4e8",
        "location": null,
        "id": 50316451
    },
    "favorited": false,
    "created_at": "Wed Jun 24 15:04:13 +0000 2009",
    "in_reply_to_screen_name": null,
    "truncated": false,
    "id": 2311383114,
    "in_reply_to_status_id": null,
    "source": "web"
}
```


Một cách khác, bạn có thể lấy ra cùng một tài liệu ở dạng Atom, RSS, hoặc ở một dạng ký pháp XML thân thiện. Trong Liệt kê 12, bạn gọi ra cùng một hàm để lấy ra trạng thái đó, nhưng lần này ở dạng ký pháp XML thân thiện:

Liệt kê 12. Lấy ra trạng thái với mã nhận dạng 2311383114 ở khuôn dạng XML thân thiện

purexml@watson.ibm.com:~$ curl

http://purexmltest:3ce3ac99@twitter.com/statuses/show.xml?id=2311383114

<?xml version="1.0" encoding="UTF-8"?>

<status>

<created_at>Wed Jun 24 15:04:13 +0000 2009</created_at>

<id>2311383114</id>

<text>Hello World!</text>

<source>web</source>

<truncated>false</truncated>

<in_reply_to_status_id></in_reply_to_status_id>

<in_reply_to_user_id></in_reply_to_user_id>

<favorited>false</favorited>

```
  <in_reply_to_screen_name></in_reply_to_screen_name>
  <user>
    <id>50316451</id>
    <name>pureXML TEST</name>
    <screen_name>purexmltest</screen_name>
    <location></location>
    <description></description>
    <profile_image_url>http://s3.amazonaws.com/twitter_production/
        profile_images/280225879/twitterxml-
1_bigger_normal.png</profile_image_url>
    <url></url>
    <protected>false</protected>
    <followers_count>0</followers_count>
    <profile_background_color>9ae4e8</profile_background_color>
    <profile_text_color>000000</profile_text_color>
    <profile_link_color>0000ff</profile_link_color>
    <profile_sidebar_fill_color>e0ff92</profile_sidebar_fill_color>
```

```
    <profile_sidebar_border_color>87bc44</profile_sidebar_border_color>
    <friends_count>0</friends_count>
    <created_at>Wed Jun 24 14:18:32 +0000 2009</created_at>
    <favourites_count>0</favourites_count>
    <utc_offset></utc_offset>
    <time_zone></time_zone>
    <profile_background_image_url>
http://static.twitter.com/images/themes/theme1/bg.gif</profile_background_image_url>
    <profile_background_tile>false</profile_background_tile>
    <statuses_count>1</statuses_count>
    <notifications></notifications>
    <verified>false</verified>
    <following></following>
  </user>
</status>
```


Liệt kê 13 cho thấy một ví dụ lệnh gọi hàm DB2 do người dùng định nghĩa để chuyển JSON thành JSONx trên các kết quả thu được trong Liệt kê 12.

Liệt kê 13. Biểu diễn JSONx của tài liệu JSON thu được trong Liệt kê 12.

db2 => SELECT JSONXML.JSONTOXML('{"text":"Hello World!",

"in_reply_to_user_id":null,"user":{"following":null,

"favourites_count":0,"profile_sidebar_fill_color":"e0ff92",

"description":null,"verified":false,"utc_offset":null,

"statuses_count":1,"profile_sidebar_border_color":"87bc44",

"followers_count":0,"created_at":"Wed Jun 24 14:18:32 +0000 2009",

"url":null,"name":"pureXML TEST","friends_count":0,

"profile_text_color":"000000","protected":false,"profile_image_url":

"http:\/\/s3.amazonaws.com\/twitter_production

\/profile_images\/280225879\/twitterxml-1_bigger_normal.png",

"profile_background_image_url":"http:\/\/static.twitter.com

\/images\/themes\/theme1\/bg.gif",

"notifications":null,"time_zone":null,"profile_link_color":"0000ff",

"screen_name":"purexmltest","profile_background_tile":false,

"profile_background_color":"9ae4e8","location":null,"id":50316451},

"favorited":false,"created_at":"Wed Jun 24 15:04:13 +0000 2009",

"in_reply_to_screen_name":null,"truncated":false,"id":2311383114,

"in_reply_to_status_id":null,"source":"web"}') FROM SYSIBM.SYSDUMMY1

<json:object xmlns:json="http://www.ibm.com/xmlns/prod/2009/jsonx">

<json:string name="text">Hello World!</json:string>

<json:null name="in_reply_to_user_id"/>

<json:object name="user">

<json:null name="following"/>

<json:number name="favourites_count">0</json:number>

<json:string name="profile_sidebar_fill_color">e0ff92</json:string>

<json:null name="description"/>

<json:boolean name="verified">false</json:boolean>

<json:null name="utc_offset"/>

<json:number name="statuses_count">1</json:number>

<json:string name="profile_sidebar_border_color">87bc44</json:string>

<json:number name="followers_count">0</json:number>

<json:string name="created_at">Wed Jun 24 14:18:32 +0000 2009</json:string>

<json:null name="url"/>

<json:string name="name">pureXML TEST</json:string>

<json:number name="friends_count">0</json:number>

<json:string name="profile_text_color">000000</json:string>

<json:boolean name="protected">false</json:boolean>

<json:string name="profile_image_url"

>http://s3.amazonaws.com/twitter_production/profile_images

/280225879/twitterxml-1_bigger_normal.png</json:string>

<json:string name="profile_background_image_url"

>http://static.twitter.com/images/themes/theme1/bg.gif</json:string>

<json:null name="notifications"/>

<json:null name="time_zone"/>

<json:string name="profile_link_color">0000ff</json:string>

<json:string name="screen_name">purexmltest</json:string>

<json:boolean name="profile_background_tile">false</json:boolean>

<json:string name="profile_background_color">9ae4e8</json:string>

<json:null name="location"/>

<json:number name="id">50316451</json:number>

</json:object>

<json:boolean name="favorited">false</json:boolean>

<json:string name="created_at">Wed Jun 24 15:04:13 +0000 2009</json:string>

<json:null name="in_reply_to_screen_name"/>

<json:boolean name="truncated">false</json:boolean>

<json:number name="id">2311383114</json:number>

<json:null name="in_reply_to_status_id"/>

<json:string name="source">web</json:string>

</json:object>

Xây dựng một ứng dụng pureXML và JSON, Phần 2: Tạo ra các dịch vụ phổ dụng dùng cho pureXML để đưa ra JSON

Lập cấu hình, triển khai, và kiểm thử Các Dịch vụ Phổ dụng JSON trên một kho lưu trữ JSONx

Faton (Tony) Avdiu, Kỹ sư phần mềm, IBM

Susan Malaika, Chuyên viên kỹ thuật cao cấp, IBM Japan

Michael Schenker, Kỹ sư phần mềm, IBM

Tóm tắt:

Các Dịch vụ Phổ dụng pureXML® dùng cho JSON (viết tắt là Các Dịch vụ Phổ dụng JSON trong bài này) là một tập hợp các phép toán cơ sở dữ liệu, gồm chèn, cập nhật, xóa, và truy vấn, được đưa ra như là các dịch vụ Web. Các dịch vụ này cho phép một ứng dụng duy trì lâu bền JSON trong pureXML và truy vấn nó một cách dễ dàng thông qua HTTP với Máy chủ Ứng dụng WebSphere®. Hãy bắt đầu với việc lập cấu hình và kiểm thử Các Dịch vụ Phổ dụng JSON trong bài viết này.

03/11/2009 - Các tác giả đã bổ sung phần Thông tin thêm về Các phép toán Các dịch vụ Phổ dụng JSON với Bảng 3 để mô tả các thông số, các mã hóa, và các kiểu MIME gắn kết với các phép toán trong Các dịch vụ Phổ dụng JSON,

Giới thiệu

Đây là bài thứ hai trong loạt ba bài viết minh họa cách xây dựng một ứng dụng pureXML ba tầng có sử dụng các tiểu trình JavaScript hoặc OpenSocial. Bằng cách làm theo các bước trong bài này, bạn sẽ trưng ra dữ liệu JSON đã mô tả trong bài đầu tiên của loạt bài, thông qua Các Dịch vụ Phổ dụng JSON.

Các bài khác trong loạt bài này

- Phần 1: Lưu trữ và truy vấn JSON với pureXML của DB2
- Phần 3: Tạo các tiểu trình OpenSocial cho pureXML

JSONx là một biểu diễn XML không tổn hao tính chính tắc dùng cho JSON, nó cho phép các thiết bị và phần mềm hỗ trợ XML làm việc với JSON. Trong bài viết đầu của loạt bài này, chúng tôi đã giới thiệu về JSONx và đã minh họa cách lưu trữ JSONx trong một cơ sở dữ liệu pureXML. pureXML cho phép lưu trữ, lập chỉ mục, và truy vấn XML thông qua các ngôn ngữ như SQL/XML, XQuery và XPath.

Các Dịch vụ Phổ dụng dùng cho pureXML là một tập đơn giản nhưng cố định các phép toán cơ sở dữ liệu mà cho phép truy vấn và sửa đổi XML, được lưu trong một cột pureXML của cơ sở dữ liệu DB2®. Các phép toán cơ sở dữ liệu, gồm chèn, cập nhật, xóa, và truy vấn, được đưa ra như Các Dịch vụ Web, trả lại dữ liệu cho khách hàng ở dạng XML. Các Dịch vụ Phổ dụng sẽ sẵn dùng thông qua các bước lập cấu hình và triển khai dễ thực hiện. Để có thêm thông tin về Các Dịch vụ Phổ dụng xin xem bài viết trên developerWorks, "Các Dịch vụ Phổ dụng dùng cho pureXML bằng cách sử dụng Các Dịch vụ Web Dữ liệu " (xem Tài nguyên để nhận được liên kết).

Các thuật ngữ thường sử dụng

- HTML: Ngôn ngữ Đánh dấu Siêu Văn bản
- HTTP: Giao thức Truyền Siêu văn bản
- JSON: Ký pháp Đối tượng JavaScript
- SOA: Kiến trúc Hướng Dịch vụ

- URL: Mã Định vị Tài nguyên Đồng nhất
- WAR: Các tệp tin lưu trữ Web
- XML: Ngôn ngữ Đánh dấu Mở rộng

Các Dịch vụ Phổ dụng JSON tương tự như Các Dịch vụ Phổ dụng, chỉ khác là chúng thao tác dữ liệu JSONx được lưu trữ trong một cột pureXML. Như trước đây đã đề cập, các phép toán cơ sở dữ liệu bao gồm chèn, cập nhật, xóa, và truy vấn, tuy nhiên dữ liệu trả về cho trình khách là JSON, không phải là XML.

Các Dịch vụ Phổ dụng JSON đảm bảo rằng các ứng dụng khách JavaScript (và loại khác) đều không biết rằng một bộ lưu trữ XML được dùng để duy trì JSON. Các khả năng XML nguyên sinh (native) của DB2 có thể được tận dụng với các đối tượng JSON duy trì tính lâu bền, làm cho có thể xem, truy vấn, và thao tác dữ liệu với các ngôn ngữ khai báo chẳng hạn như XPath, tăng tốc đáng kể quá trình phát triển.

Trong bài này, chúng tôi trình bày việc lập cấu hình, triển khai, và kiểm thử Các Dịch vụ Phổ dụng JSON trên một kho lưu trữ JSONx dựa trên cơ sở dữ liệu mẫu DB2 mô tả trong bài đầu tiên của loạt bài này, "Xây dựng một ứng dụng pureXML và JSON, Phần 1: Lưu trữ và truy vấn JSON với pureXML của DB2 " (xem phần Tài nguyên để nhận được liên kết).

Nếu bạn cần nhiều chi tiết hơn về các bước này, xin tham khảo tài liệu README.pdf của gói tải về Các Dịch vụ Phổ dụng JSON để có hướng dẫn các bước được minh họa đầy đủ. (Sau khi giải nén gói này, bạn sẽ tìm thấy tệp tin README.pdf trong thư mục JSONUniversalServices).

Các điều kiện tiên quyết

Trước khi cài đặt Các Dịch vụ Phổ dụng JSON, bạn cần hoàn thành phần cài đặt mô tả trong bài đầu tiên của loạt bài này. Hãy chạy gói JSONx, nó sẽ tạo ra cơ sở dữ liệu, và đăng ký các hàm Java™ do người sử dụng định nghĩa (UDF) cần thiết trong DB2. Các hàm do người dùng định nghĩa này cho phép chuyển đổi JSON thành JSONx và ngược lại. Bạn cũng cần phải có bản DB2 9.5 hoặc 9.7, môi trường chạy thi hành Java (Java Runtime Environment) 1.5.0, và một máy chủ Web (chẳng hạn như WebSphere Application Server V6.x hoặc Apache Tomcat V6.x), cài trên hệ thống của bạn. Lưu ý rằng Java Runtime là một phần của bản cài đặt DB2. Nếu bạn còn chưa cài đặt các sản phẩm này, xin đọc bài đầu tiên. Để có liên kết đến bài viết đầu tiên, xin xem phần Tài nguyên.

Trong bài đầu tiên của loạt bài, bạn đã tạo một bảng, CUSTOMER (Khách hàng), bạn sẽ sử dụng nó để cài đặt và kiểm thử Các Dịch vụ Phổ dụng JSON của bạn. Bảng 1 cho thấy một cái nhìn khái quát của bảng CUSTOMER:

Bảng 1. Khái quát về Bảng JSONXMLADMIN.CUSTOMER

Tên cột	Kiểu dữ liệu	Mô tả
CID	INTEGER	Mã nhận dạng khách hàng
INFO	XML	Một bản ghi chứa thông tin cá nhân về khách hàng
COMMENT	VARCHAR(256)	Một mã nhận dạng văn bản nhỏ liên quan đến

khách hàng

Bây giờ bạn đã có các điều kiện tiên quyết được cài đặt và bảng của bạn đã được tạo ra, bạn có thể lập cấu hình và cài đặt Các Dịch vụ Phổ dụng JSON đối với một cột pureXML cụ thể.

Cài đặt Các Dịch vụ Phổ dụng JSON

Phần này giả định rằng bạn đã tải về và giải nén gói Các Dịch vụ Phổ dụng JSON vào một thư mục làm việc cục bộ, tạm thời. Thư mục dành cho Các Dịch vụ Phổ dụng JSON được sử dụng trong suốt bài này là C:\temp\JSONUniversalServices. Bạn sẽ thấy các thư mục và tệp tin như trong Hình 1.

Hình 1. Nội dung của gói tải về Các Dịch vụ Phổ dụng JSON

Các thư mục gồm có classes, services, và war. Các tệp tin gồm configure.bat, configure.sh, và README.pdf.

Khi các tệp tin và lệnh cần thiết đã sẵn sàng trên hệ thống của bạn, bạn có thể tiếp tục cài đặt Các Dịch vụ Phổ dụng JSON, mà cơ bản gồm có ba bước:

- Lập cấu hình Các Dịch vụ Phổ dụng JSON để làm cho hợp với cài đặt hệ thống cục bộ của bạn.
- Triển khai Các Dịch vụ Phổ dụng JSON trên máy chủ ứng dụng của bạn.
- Kiểm thử Các Dịch vụ Phổ dụng JSON.

Lập cấu hình Các Dịch vụ Phổ dụng JSON

Việc cấu hình của các dịch vụ Web được hoàn tất thông qua một kịch bản lệnh được cung cấp với phần tải về tên là configure.bat (đối với người sử dụng (Linux® nó có tên là configure.sh). Kịch bản lệnh này được sửa đổi để làm việc với gói JSONx. Do đó nếu bạn đã sửa đổi gói JSONx, hoặc nếu bạn quyết định sử dụng một cơ sở dữ liệu khác, bạn cần sửa đổi kịch bản lệnh configure.bat trước khi thực hiện nó. Bạn có thể phải điều chỉnh một số thông số để cho hợp với cài đặt hệ thống cục bộ của bạn. Bảng 1 cho thấy các thông số được sử dụng trong kịch bản lệnh lập cấu hình này.

Bảng 2. Tổng quan về các thông số được sử dụng trong kịch bản lệnh lập cấu hình

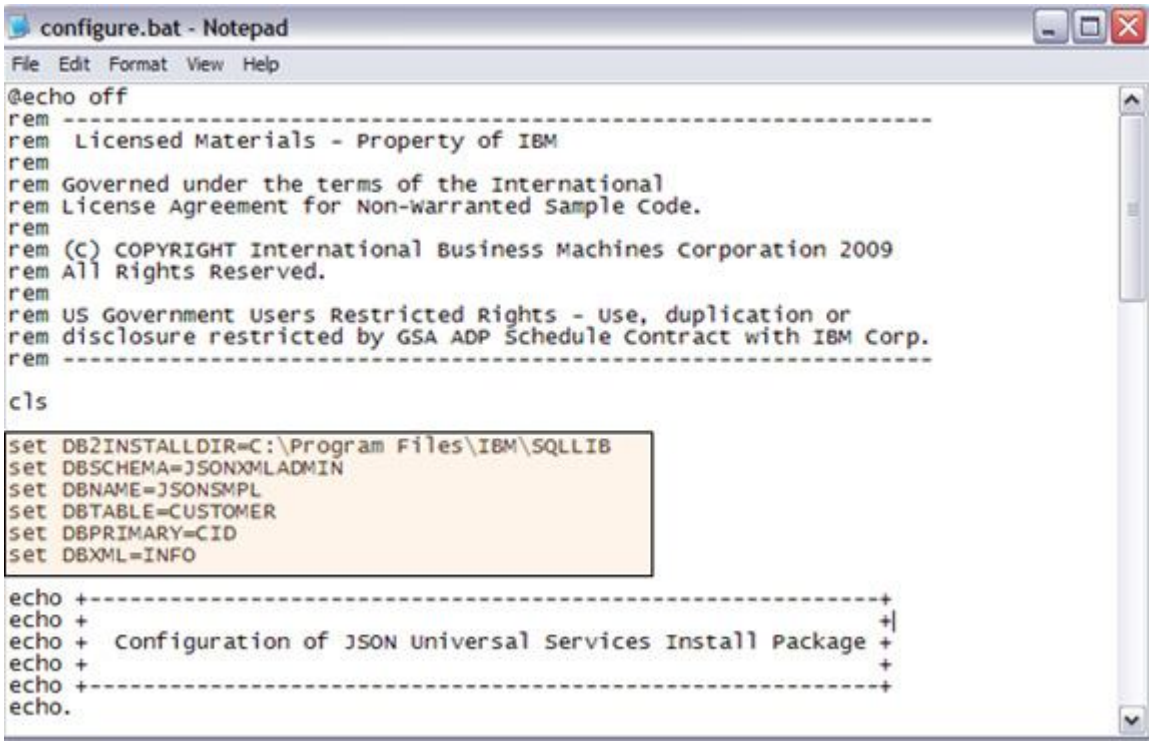
Tên thông số	Mô tả	Giá trị (theo cơ sở dữ liệu mẫu)
DB2INSTALLDIR	Đường dẫn đến thư mục cài đặt DB2.	C:\Program Files\IBM\SQLLIB

DBSCHEMA	Tên lược đồ cơ sở dữ liệu có chứa bảng có cột XML đang xét.	JSONXMLADMIN
DBNAME	Tên của cơ sở dữ liệu mà Các Dịch vụ Phổ dụng được sử dụng với cơ sở dữ liệu đó.	JSONSMPL
DBTABLE	Tên của bảng chứa cột XML.	CUSTOMER
DBPRIMARY	Tên của cột khóa chính của bảng xác định bởi DBTABLE. Cột này có thể là kiểu INTEGER hoặc VARCHAR.	CID
DBCOMMENT	Tên của cột chú thích của bảng xác định bởi DBTABLE.	COMMENT
DBXML	Lưu ý rằng giá trị của ba thông số trong Bảng 2 DBPRIMARY, DBCOMMENT, và DBXML, cũng giống như tên các cột trong Bảng 1.	INFO

Lưu ý rằng giá trị của ba thông số trong Bảng 2 DBPRIMARY, DBCOMMENT, và DBXML, cũng giống như tên các cột trong Bảng 1.

Hình 2 cho thấy tệp tin cấu hình, sau khi đã được sửa theo các giá trị đưa ra trong Bảng 2. (Xem phần văn bản của Hình 2.)

Hình 2. Biên tập tệp tin cấu hình configure.bat



Sau khi configure.bat (đối với người sử Linux dụng là: configure.sh) được sửa đổi với các thông số phù hợp, bây giờ đã có thể thi hành kịch bản lệnh thực hiện cấu hình Các Dịch vụ Phổ dụng JSON. Kịch bản lệnh này tạo ra một tệp tin lưu trữ ứng dụng Web (WAR), chứa tất cả các tệp tin cần thiết cho ứng dụng Web Các Dịch vụ Phổ dụng JSON, và nó có thể được triển khai lên máy chủ ứng dụng của bạn. Xin nhớ rằng kịch bản lệnh này phải được thi hành trên các bộ xử lý dòng lệnh DB2, nếu không thì việc thi hành sẽ thất bại.

Triển khai Các Dịch vụ Phổ dụng JSON

Như đã mô tả trong phần trước, việc biên tập và thi hành kịch bản lệnh configure.bat đã tạo ra một tệp tin WAR mà bây giờ bạn đã có thể triển khai đến

máy chủ ứng dụng của bạn. Trước khi triển khai tệp tin WAR trên máy chủ ứng dụng, cần phải lập cấu hình một nguồn dữ liệu trên máy chủ ứng dụng, cho phép Các Dịch vụ Phổ dụng JSON truy cập cơ sở dữ liệu DB2 của bạn.

Để thiết lập nguồn dữ liệu, hãy làm theo quy trình lập cấu hình chuẩn của máy chủ ứng dụng của bạn. Yêu cầu duy nhất cần cân nhắc trong khi thiết lập là tên của nguồn dữ liệu, nó phải đáp ứng mẫu sau đây: jdbc/tên_cơ_sở_dữ_liệu. Tên nguồn dữ liệu dùng cho thí dụ trong bài viết này là jdbc/jsonsmpl. Nếu bạn muốn có thông tin bổ sung về cách thiết lập một nguồn dữ liệu, xin tham khảo tài liệu về máy chủ ứng dụng của bạn hoặc đọc tài liệu README.pdf có trong gói tải về Các Dịch vụ Phổ dụng JSON (trong thư mục JSONUniversalServices).

Bước thứ hai là triển khai tệp tin WAR thực tế lên máy chủ ứng dụng. Tuy nhiên, do có các khác biệt trong các tạo phẩm dịch vụ Web mà bản WebSphere Application Server V6.x và Apache Tomcat V6.x đòi hỏi, kịch bản lệnh lập cấu hình tạo ra hai tệp tin WAR riêng biệt, mỗi tệp tin cho một máy chủ ứng dụng.

Các tệp tin WAR có thể tìm thấy trong thư mục con war, nằm trong thư mục làm việc của Các Dịch vụ Phổ dụng JSON C:\temp\JSONUniversalServices. Đặc biệt, thư mục này chứa hai thư mục con: tomcat và was. Thư mục tomcat chứa tệp tin WAR sẽ cài đặt trên Tomcat Apache V6.x, còn thư mục chứa tệp tin WAR sẽ cài trên máy chủ ứng dụng WebSphere V6.x. Để cài đặt tệp tin WAR trên máy chủ ứng dụng của bạn, hãy làm theo thủ tục tiêu chuẩn để thực hiện công việc này. Nếu cần nhiều thông tin hơn, xin tham khảo tài liệu của máy chủ ứng dụng của bạn hoặc đọc tài liệu README.pdf kèm trong gói tải về Các Dịch vụ Phổ dụng JSON.

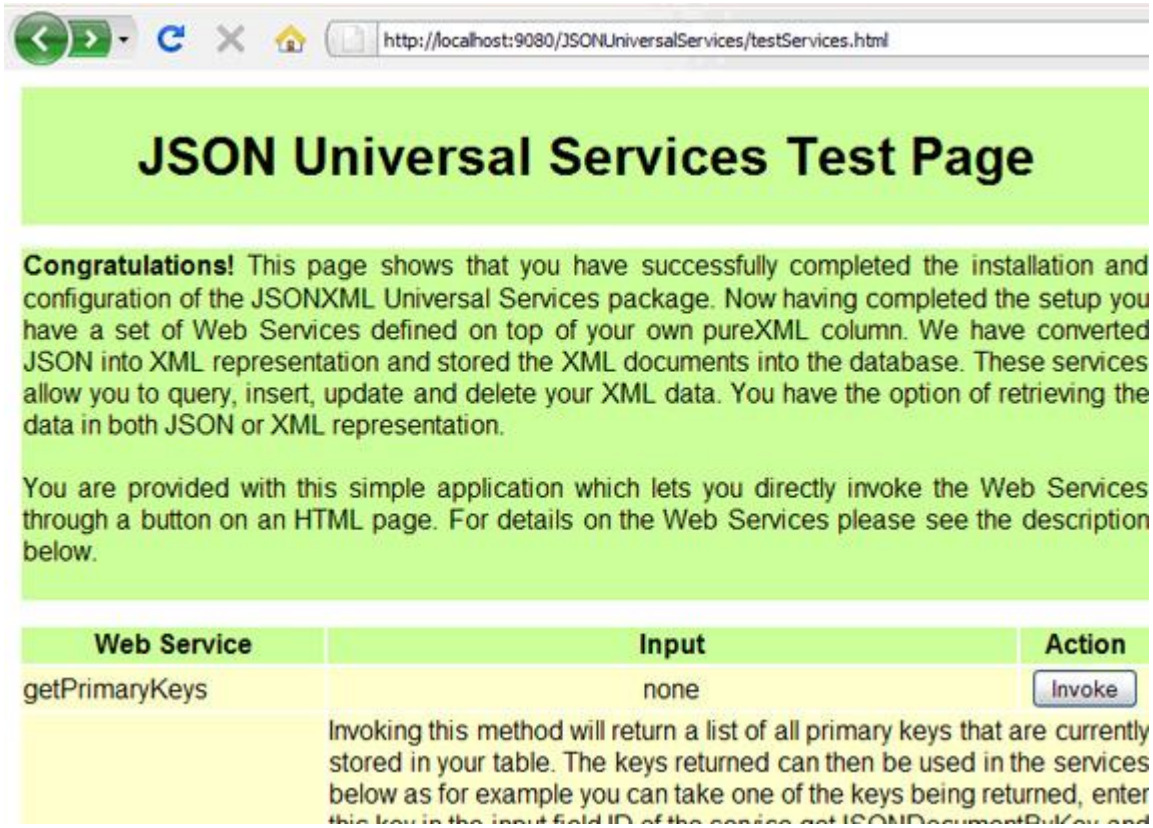
Bây giờ đã có thể kiểm thử bộ dịch vụ Web của bạn bằng trang kiểm thử Các Dịch vụ Phổ dụng JSON, đã có sẵn từ trước qua việc triển khai tệp tin WAR.

Kiểm thử Các Dịch vụ Phổ dụng JSON

Để đảm bảo rằng việc cài đặt và cấu hình của Các Dịch vụ Phổ dụng JSON đã thành công trên hệ thống cục bộ của bạn, bạn cần phải kiểm thử các dịch vụ này. Cách dễ nhất để kiểm tra là sử dụng một trang HTML đơn giản, có sẵn trên máy chủ ứng dụng của bạn như là một phần triển khai các Các Dịch vụ Phổ dụng JSON.

Để truy cập trang kiểm thử Các Dịch vụ Phổ dụng JSON, hãy hướng trình duyệt của bạn đến URL http://localhost:8080/JSONUniversalServices/testServices.html. Lưu ý là cổng 8080 được sử dụng nếu bạn đã cài đặt nó trên Apache Tomcat. Nếu bạn sử dụng WebSphere Application Server bạn phải sử dụng cổng 9080 để truy cập trang này. Chú ý nữa là tuỳ vào hệ thống cục bộ của bạn, bạn có thể phải sửa lại cho phù hợp tên máy hoặc cổng. Bạn sẽ nhìn thấy trang này ở hình 3.

Hình 3. Trang kiểm thử Các Dịch vụ Phổ dụng JSON



Trang kiểm thử đơn giản này cho phép bạn chèn, cập nhật, xóa, và truy vấn dữ liệu JSON, được lưu trữ như là JSONx trong cơ sở dữ liệu của bạn. Các dịch vụ Web được gọi ra thông qua các nút được cung cấp trên trang kiểm thử. Hình 4 cho thấy đáp ứng của dịch vụ Web, sau khi chúng tôi đã gọi tác vụ getPrimaryKeys thông qua nút trên trang kiểm thử.

Hình 4. Đáp ứng sau khi gọi hoạt động getPrimaryKeys của dịch vụ Web

Các tài liệu JSON được chuyển đổi và chèn vào cột như là các tài liệu XML. Bạn có thể lấy ra các tài liệu đó ở cả hai khuôn dạng JSON và XML.

Thí dụ, bằng cách gọi ra phương thức getJSONDocumentByKey bạn sẽ nhận được biểu diễn JSON của tài liệu:

Liệt kê 1. Thông tin khách hàng Kathy Smith dưới dạng JSON

```
{
  "customerinfo" : {
    "cid" : 1000 ,
    "name" : "Kathy Smith" ,
    "addr" : {
```

```
      "country"    : "Canada" ,
      "street"     : "5 Rosewood" ,
      "city"       : "Toronto" ,
      "prov-state" : "Ontario" ,
      "pcode-zip"  : "M6W 1E6"
    } ,
    "phone" : {
      "work" : "416-555-1358"
    }
  }
}
```


Ngoài ra, nếu bạn gọi ra phương thức getXMLDocumentByKey bạn sẽ nhận được tài liệu XML (ở khuôn dạng JSONx) giống như nó được lưu trữ trên cơ sở dữ liệu DB2 pureXML của bạn:

Liệt kê 2. Thông tin khách hàng Kathy Smith dưới dạng JSONx

```
<json:object xmlns:json="http://www.ibm.com/xmlns/prod/2009/jsonx">
  <json:object name="customerinfo">
        <json:number name="cid">1000</json:number>
        <json:string name="name">Kathy Smith</json:string>
        <json:object name="addr">
          <json:string name="country">Canada</json:string>
          <json:string name="street">5 Rosewood</json:string>
          <json:string name="city">Toronto</json:string>
          <json:string name="prov-state">Ontario</json:string>
          <json:string name="pcode-zip">M6W 1E6</json:string>
        </json:object>
        <json:object name="phone">
          <json:string name="work">416-555-1358</json:string>
        </json:object>
  </json:object>
</json:object>
```

Hơn nữa, trang này còn cung cấp các dịch vụ khác, chẳng hạn như chèn, cập nhật, xóa, và truy vấn XML. Xin đọc phần sau đây và các hướng dẫn và trên trang kiểm thử để có thêm thông tin.

Thông tin bổ sung về các hoạt động của Các dịch vụ phổ dụng JSON

Trong phần này, chúng tôi gộp thêm Bảng 3 mô tả các thông số, các mã hóa của chúng, và các kiểu MIME (Multipurpose Internet Mail Extensions) gắn với các hoạt động trong Các Dịch vụ Phổ dụng JSON. Các hoạt động này sẵn có thông qua các yêu cầu HTTP GET, HTTP POST mã hoá qua URL (URL-encoded) và kiểu mime thuần văn bản.

- Đối với các hoạt động **getDocumentByKey** và **getXMLDocumentByKey** bạn có thể chỉ rõ một mã nhận dạng và lấy ra dữ liệu được lưu trữ tương ứng dạng JSON hay XML tương ứng.

- Đối với các hoạt động **insert** (chèn) và **update** (cập nhật) bạn có thể chỉ rõ một mã nhận dạng của một tài liệu JSON sẽ chèn hay cập nhật, cùng với tài liệu mới. Nếu yêu cầu thành công bạn sẽ nhận được một số đếm cập nhật là 1. Nếu không, bạn sẽ nhận được một số đếm cập nhật là 0.

- Đối với hoạt động **delete** (xóa) bạn có thể chỉ rõ một mã nhận dạng của một tài liệu sẽ bị xóa từ kho lưu trữ pureXML. Nếu yêu cầu xóa thành công bạn sẽ nhận được một số đếm cập nhật là 1. Nếu không, bạn sẽ nhận được một số đếm cập nhật là 0.

- Đối với hoạt động **runXMLQuery** (chạy truy vấn XML), bạn có thể cung cấp một truy vấn XQuery hoặc một đầu vào cho một truy vấn XMLQUERY (một phần của SQL/XML) để lấy ra nội dung XML mà có

thể trùm qua nhiều tài liệu được lưu trữ. Xem trang kiểm thử để có một số thí dụ.

Bảng 3. Tổng quan về các hoạt động của Các Dịch vụ Phổ dụng JSON

Hoạt động	Thông số đầu vào	HTTP GET	HTTP POST MIME_TYPE application/x-www-form-urlencoded	HTTP POST MIME-TYPE text/plain, application/json	Đáp ứng
getDocumentByKey	**id**	**id:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi truy vấn	**id:** Cặp tham số/giá trị được mã hoá qua URL trong thông báo	**id:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi truy vấn	tài liệu JSON (text/plain)

getXMLDocumentByKey	**id**	**id:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi truy vấn	**id:** Cặp tham số/giá trị được mã hoá qua URL trong thông báo	**id:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi truy vấn	Biểu diễn JSONx của tài liệu (text/plain)
insert	**id** (duy nhất) **doc** (tài liệu JSON)	**id:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi	**id:** Cặp tham số/giá trị được mã hoá qua URL trong thông báo **doc:** Cặp tham số/giá trị được mã hoá qua URL trong thông báo	**id:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi truy vấn **doc:** Nội dung thông báo POST	Số đếm cập nhật (text/plain)

		truy vấn **doc:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi truy vấn			
update	**id** **doc** (tài liệu JSON)	**id:** Cặp tham số/giá trị được mã hoá qua URL trong	**id:** Cặp tham số/giá trị được mã hoá qua URL trong thông báo **doc:** Cặp tham số/giá trị được mã hoá qua URL trong thông báo	**id:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi truy vấn **doc:** Nội dung thông báo POST	Số đếm cập nhật (text/plain)

		chuỗi truy vấn **doc:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi truy vấn			
delete	**id**	**id:** Cặp tham số/giá trị được mã hoá qua URL	**id:** Cặp tham số/giá trị được mã hoá qua URL trong thông báo	**id:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi truy vấn	Số đếm cập nhật (text/plain)

		trong chuỗi truy vấn			
runXMLQuery	**query** (truy vấn XQuery hoặc đầu vào cho một XMLQUERY)	**query:** Cặp tham số/giá trị được mã hoá qua URL trong chuỗi truy vấn	**query:** Cặp tham số/giá trị được mã hoá qua URL trong thông báo	**query:** Nội dung thông báo POST	Tài liệu XML với các kết quả (text/xml)

Kết luận

Các bài khác trong loạt bài này

- Phần 1: Lưu trữ và truy vấn JSON với pureXML của DB2

- Phần 3: Tạo các tiểu trình OpenSocial cho pureXML

Bài viết này đã cung cấp cho bạn một giới thiệu nhập môn về Các Dịch vụ Phổ dụng JSON, gồm việc lập cấu hình, triển khai và kiểm thử Các Dịch vụ Phổ dụng JSON trên kho lưu trữ JSONx dựa trên cơ sở dữ liệu mẫu DB2.

Bài đầu tiên trong loạt bài này tập trung vào việc giới thiệu định dạng mới JSONx, cho phép lưu trữ JSON như là XML trong cơ sở dữ liệu pureXML DB2. Bài viết cũng bàn luận về một trường hợp sử dụng đối với JSON và phác thảo ra các lợi ích của việc sử dụng pureXML để lưu trữ JSON.

Bài tiếp theo của loạt bài này sẽ tập trung vào việc tạo ra tầng trình bày với các tiểu trình Open-Social dựa vào Các Dịch vụ Phổ dụng JSONx như là một nền phụ trợ.

Tải về

Mô tả	Tên	Kích thước	Phương thức tải
JSON Universal Services package	JSONUniversal Services.zip	3692KB	HTTP

Thông tin về phương thức tải

Tài nguyên

Học tập

- Các Dịch vụ Phổ dụng dùng cho pureXML bằng cách sử dụng Các Dịch vụ Web Dữ liệu (Susan Malaika và Christian Pichler, developerWorks, 8/2008): Nhận thông tin bổ sung về Các Dịch vụ Phổ dụng dùng cho pureXML.

- Xây dựng một ứng dụng pureXML và JSON, Phần 1: Lưu trữ và truy vấn JSON với pureXML của DB2 (Nuno Job, Susan Malaika, và Michael Schenker; developerWorks, 10/2009): Xem bài đầu tiên trong loạt bài này và bắt đầu thu lợi ích từ việc lưu giữ lâu bền các đối tượng JSON để duy trì trạng thái qua các phiên. Lưu trữ, quản lý, và truy vấn JSON bằng DB2 pureXML và một ánh xạ JSON-thành-XML đơn giản. (Phần 1 của loạt bài ba phần.)

- Xây dựng một ứng dụng pureXML và JSON, Phần 3: Tạo các tiểu trình OpenSocial dùng cho pureXML (Han Nguyen, Andy B Smith, và Mark D Weitzel; developerWorks, 11/2009): Định nghĩa, triển khai, và thử nghiệm các tiểu trình OpenSocial để tương tác với Các Dịch vụ Phổ dụng JSON. (Phần 3 của loạt bài 3 phần.)

- IBM Data Studio: Bắt đầu với Các dịch vụ Web Dữ liệu (Michael L. Pauser, developerWorks, 11/2007): Đọc một bài giới thiệu nhập môn rất chi tiết và đơn giản về cách phát triển Dịch vụ Web Dữ liệu đầu tiên của bạn.

- "Các Khuôn dạng và Dịch vụ Công nghiệp với pureXML": Hãy tải về nhiều thí dụ, miễn phí! Mỗi thí dụ minh họa cách làm việc với Các Khuôn

dạng Công nghiệp dựa trên XML và pureXML. Các thí dụ cho thấy cách đăng ký một Lược đồ XML, cách thực hiện việc xác thực các cá thể tài liệu XML, cách truy vấn dữ liệu XML bằng cách sử dụng XQuery hoặc SQL/XML và nhiều hơn nữa.

- Các dịch vụ Web Dữ liệu: Xây dựng Các dịch vụ Web, một cách mới để truy cập máy chủ cơ sở dữ liệu IBM (Vijay Bommireddipalli, developerWorks, 12/2007): Tạo và tùy biến một Dịch vụ Web Dữ liệu với nền tảng lý thuyết hữu ích về Các dịch vụ Web Dữ liệu, bao gồm một tổng quan kiến trúc về Các Dịch vụ Web Dữ liệu. Bài viết nhằm vào các khía cạnh khác nhau của Các dịch vụ Web Dữ liệu, chẳng hạn như an ninh.

- Sử dụng các khuôn dạng dữ liệu chuẩn công nghiệp với WebSphere ESB và pureXML của DB2 Version 9 (Mohab El-Hilaly, Andre Tost, và Alaa Youssef; developerWorks, 6/2007): Tìm một kịch bản mà chỉ ra cách làm thế nào sử dụng WebSphere Enterprise Service Bus và pureXML DB2 9 để có thể giúp một công ty dược phẩm đáp ứng các nghĩa vụ pháp lý về trưng ra dữ liệu tiêu chuẩn hóa về các sản phẩm của mình.

- Chứng nhận XML của IBM: Tìm hiểu cách làm thế nào để bạn có thể trở thành một Nhà phát triển được IBM chứng nhận về XML và công nghệ liên quan.

- Thư viện kỹ thuật XML: Xem vùng XML trên developerWorks để có nhiều bài viết và mách nước về kỹ thuật, các hướng dẫn, tiêu chuẩn, và Sách đỏ của IBM.

- Các sự kiện và phát tin kỹ thuật trong developerWorks: Theo sát công nghệ trong các phiên này.

- developerWorks podcasts: Nghe các bài phỏng vấn và thảo luận thú vị đối

Xây dựng một ứng dụng pureXML và JSON, Phần 3: Tạo ra các tiểu trình OpenSocial dùng cho pureXML

Xác định, triển khai, và thử nghiệm các tiểu trình OpenSocial để tương tác với Các Dịch vụ Phổ dụng JSON

Han Nguyen, Kỹ sư phần mềm, IBM

Andy Smith, Kỹ sư phần mềm, IBM

Mark Weitzel, Kiến trúc sư phân mềm, IBM

Tóm tắt: Với công nghệ Web 2.0 của các tiểu trình (gadget) OpenSocial, các nhà phát triển có thể dễ dàng gồm thêm các ứng dụng của họ vào các địa chỉ Web phổ biến, chẳng hạn như iGoogle, MySpace, Hi5, LinkedIn, và các địa chỉ khác. Trong bài này, hãy khảo sát các tiểu trình OpenSocial thông qua việc xây dựng thực hành một ứng dụng tận dụng các khả năng pureXML® của DB2®. Bài viết này là phần cuối cùng trong loạt bài ba phần, nó minh họa cách xây dựng một ứng dụng pureXML có giao diện người sử dụng là một tiểu trình mà bạn có thể triển khai trong bất kỳ địa chỉ Web tương thích với OpenSocial nào. Hãy làm theo các bước trong bài này để xây dựng một giao diện người sử dụng mà lưu trữ và lấy ra dữ liệu JSON đã mô tả trong bài viết đầu tiên thông qua Các Dịch vụ Phổ dụng JSON đã tạo ra trong bài thứ hai.

Nền tảng về OpenSocial

Các bài khác trong loạt bài này

- Phần 1: Lưu trữ và truy vấn JSON với pureXML của DB2
- Phần 2: Tạo Các Dịch vụ Phổ dụng cho pureXML để đưa ra JSON

OpenSocial là một đặc tả định hướng cộng đồng, nó định nghĩa một mô hình thành phần dựa trên trình duyệt, được biết đến như là *các tiểu trình* (tạm dịch thuật ngữ tiếng Anh "gadget" – một vật dụng nhỏ, hấp dẫn bởi tính mới mẻ. Nhiều khi cũng tương tự như "widget" – N.D.), và một giao diện lập trình ứng dụng (API) để truy cập thông tin lược tả của người sử dụng, cũng như *đồ thị xã hội* của họ (gồm cả những điều như bạn bè và các hoạt động của họ). Các ứng dụng thực hiện các API này sẽ có khả năng liên tác làm việc với một tập hợp nhiều trang mạng xã hội như: iGoogle, MySpace, Yahoo, Orkut, Hi5, LinkedIn, và các trang khác. Trong bài này chúng tôi sẽ tập trung vào các tiểu trình OpenSocial và trình bày cách làm thế nào để chúng có thể là một biện pháp mạnh mẽ để mở rộng tầm với của ứng dụng của bạn trên Web.

Các thuật ngữ thường sử dụng

- API: Giao diện lập trình ứng dụng
- CSS: Các phiếu định kiểu đè lên
- HTML: Ngôn ngữ Đánh dấu Siêu Văn bản
- HTTP: Giao thức Truyền Siêu văn bản
- JSON: Ký pháp Đối tượng JavaScript
- SOA: Kiến trúc Hướng Dịch vụ
- UI: Giao diện người sử dụng
- URL: Mã Định vị Tài nguyên Đồng nhất
- XML: Ngôn ngữ Đánh dấu Mở rộng

Tiểu trình OpenSocial là gì?

- Tiểu trình OpenSocial là một trình diễn nhỏ của một ứng dụng Web, nó thực hiện một tập hợp riêng các API. Một tiểu trình được mô tả bởi một tài liệu XML tuân thủ đặc tả OpenSocial. Định nghĩa này bao gồm giao diện người sử dụng như HTML, các phiếu định kiểu CSS, và JavaScript dành cho logic xử lý nghiệp vụ, và siêu dữ liệu bổ sung thêm dành cho tác giả, tiêu đề, v.v.

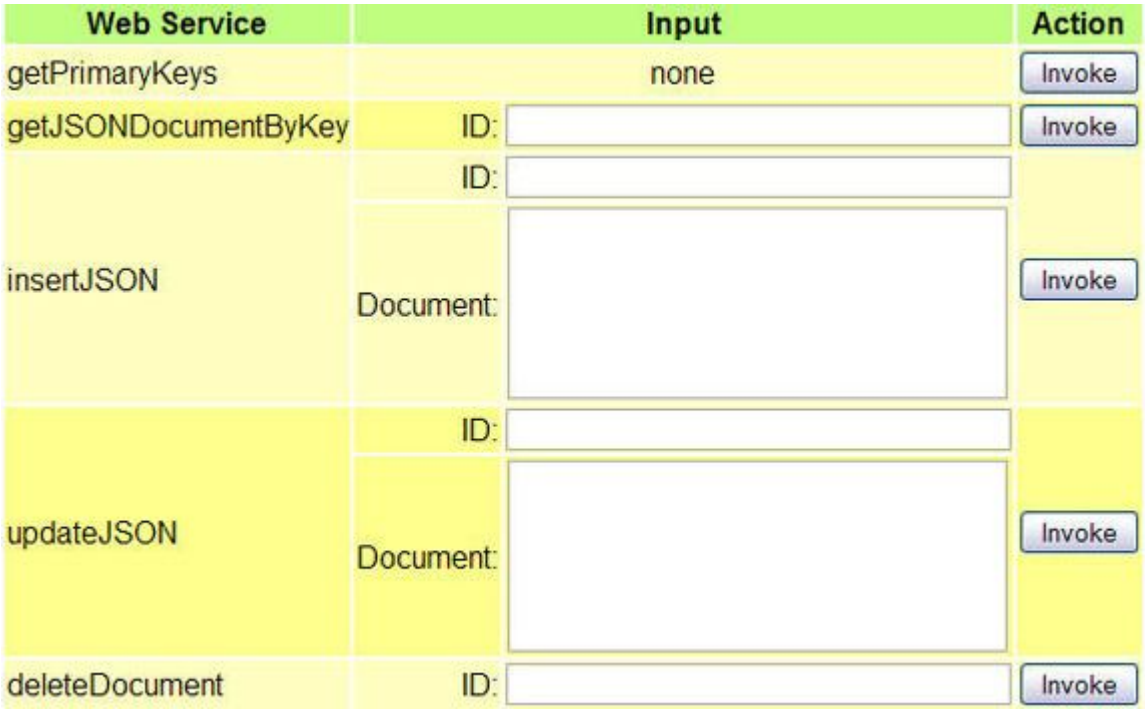
- Một trang web thực hiện đặc tả OpenSocial và có thể là chủ chứa các ứng dụng gọi là một *thùng chứa OpenSocial* (OpenSocial container). Điều này có nghĩa là nó có thể xử lý các định nghĩa tiểu trình bằng XML, và cung cấp HTML thích hợp cho trình duyệt. Điều quan trọng là cần nhận thấy rằng các định nghĩa tiểu trình không nhất thiết phải được chứa trong trang web là thùng chứa OpenSocial đó. Hơn nữa, các tiểu trình thường đưa ra một dịch vụ từ một trang hoàn toàn khác. Thùng chứa cung cấp các cơ chế, chẳng hạn như các yêu cầu HTML đã ký (signed HTML requests), để mang lại một mức độ bảo đảm rằng các cuộc gọi có nguồn gốc từ tiểu trình trong trang web của mình. Là một thùng chứa OpenSocial, một trang web có thể cung cấp một cách dễ dàng để gộp nhóm một tập hợp các dịch vụ từ khắp các nơi trên Web.

Tạo ra tiểu trình

Trang này cung cấp cho người sử dụng một phương tiện để kiểm tra tập hợp cơ sở các hàm do Các Dịch vụ Phổ dụng cung cấp và hiển thị kết quả cho người sử dụng

Hình 1 là một ảnh chụp màn hình của ứng dụng kiểm thử mẫu. Nó sẽ được dùng làm khuôn mẫu mà bạn sử dụng để xây dựng tiểu trình OpenSocial mẫu. Điều này cung cấp cho người sử dụng các khả năng về giao diện người dùng (UI) cơ bản cần có để tương tác với Các Dịch vụ Phổ dụng.

Hình 1. Ứng dụng kiểm thử mẫu



Trước tiên, bạn sẽ tạo ra một vài hàm JavaScript để cung cấp các truy vấn cơ sở điều khiển các kết nối với dịch vụ pureXML. Sau đó, bạn sẽ bổ sung hỗ trợ HTML cùng với đặc tả tiểu trình này. Cuối cùng, bạn xem cách làm thế nào để tạo và triển khai ứng dụng này.

Các hàm JavaScript để kết nối với các dịch vụ pureXML

Liệt kê 1 cung cấp JavaScript mẫu đã được tạo ra để xử lý các yêu cầu.

Liệt kê 1. Các hàm JavaScript PureJSON

```
var prefs = new gadgets.Prefs();
function getPrimaryKeys()            {
            var args = {};
            doPOST("getPrimaryKeys",args,displayJSONobj);
};
function getJSONDocumentByKey(key)     {
            var args = {
              id: key
            };
            doPOST("getDocumentByKey",args,displayJSONobj);
};
function insertJSON(key, data)       {
            var args = {
              id: key,
              doc: data
            };
```

```
        doPOST("insert",args,response);
};
function updateJSON(key, data)    {
        var args = {
          id : key,
          doc : data
        };
   doPOST("update",args,response);
};
function deleteDocument(key)      {
        var args = {
          id : key
        };
   doPOST("delete",args,response);
};
function doPOST(command,args,processResponseCallback)  {
        var url = "http://" + prefs.getString("pureXMLHostAddress") +
```

```
                                "/" + prefs.getString("contextRoot") +
"/query/"+command;

        var params = {};

        postdata = gadgets.io.encodeValues(args);

        params[gadgets.io.RequestParameters.METHOD] =
gadgets.io.MethodType.POST;

        params[gadgets.io.RequestParameters.POST_DATA]= postdata;

        gadgets.io.makeRequest(url, processResponseCallback, params);

};

function response(obj)      {

        alert("Gadget implementation responsibility.");

};

function displayJSONobj(obj)        {

        alert("Gadget implementation responsibility.");

};
```

Hãy xem mục đích của từng hàm:

- getPrimaryKeys gửi một yêu cầu getPrimaryKeys đến dịch vụ để lấy ra toàn bộ các khóa chính từ bảng DB2 và hiển thị kết quả bằng cách sử dụng hàm gọi ngược displayJSONobj khi giao dịch hoàn tất.

- getJSONDocumentByKey sử dụng một giá trị khoá để gửi một yêu cầu getDocumentByKey đến dịch vụ để lấy ra một bản ghi JSON đơn lẻ với khóa chính khớp với giá trị đầu vào. Nó hiển thị kết quả bằng cách sử dụng hàm gọi ngược displayJSONobj.

- insertJSON chờ hai thông số đầu vào: giá trị khoá duy nhất dành cho khóa chính và dữ liệu hàng dưới khuôn dạng JSON. Nó gửi một yêu cầu đến dịch vụ để tạo ra một hàng mới trong bảng DB2, và xác nhận tình trạng chèn khi giao dịch trả về bằng cách sử dụng hàm gọi ngược response.

- updateJSON cung cấp khả năng gửi một yêu cầu đến dịch vụ để cập nhật một bản ghi JSON với khóa chính khớp với giá trị khoá đầu vào, và dữ liệu cập nhật được xác định như là dữ liệu đầu vào ở khuôn dạng JSON. Nó cũng sẽ gọi ra hàm gọi ngược response để xác nhận cập nhật đó khi giao dịch hoàn tất.

- deleteDocument cung cấp khả năng gửi một yêu cầu đến dịch vụ để cập nhật một bản ghi JSON với khóa chính khớp với giá trị khoá đầu vào, và dữ liệu cập nhật được xác định như là dữ liệu đầu vào ở khuôn dạng JSON. Nó cũng sẽ gọi ra hàm gọi ngược response để xác nhận cập nhật đó khi giao dịch hoàn tất.

- displayJSONobj là một hàm trừu tượng JavaScript mang lại một móc treo cho các nhà phát triển để cung cấp các cách tiếp cận khác nhau để biểu hiện các kết quả cuộc gọi trong trình duyệt.

- response là một hàm có thể xử lý các giá trị trả về mà không phải là JSON. Nó đi theo một chiến lược tương tự như displayJSONobj bằng cách chờ các nhà phát triển viết các triển khai thực hiện riêng của chính họ đè lên hàm đó.

Các hàm nói trên gọi hàm doPOST để đệ trình các yêu cầu của chúng. Hàm doPOST chờ đợi lệnh truy vấn của dịch vụ, các giá trị đầu vào của truy vấn, và một hàm gọi ngược để xử lý kết quả trả về từ các dịch vụ đó. Nó sử dụng các thông số đầu vào này để xây dựng các cuộc gọi gadgets.io.makeRequest đến các dịch vụ. gadgets.io.makeRequest là một hàm các tiểu trình OpenSocial cung cấp việc hỗ trợ cho các tiểu trình để nhận dữ liệu từ và gửi dữ liệu đến các trang web bên thứ ba. Chữ ký của nó trông giống như thế này: gadgets.io.makeRequest(url, callback, opt_params)

Các thông số là:

- url - Một chuỗi ký tự chứa URL của trang web mà bạn muốn gửi yêu cầu đến. Bạn có thể thấy rằng để xây dựng URL thì hàm doPOST lấy ra hai chuỗi ký tự từ đối tượng prefs bằng cách sử dụng phương thức getString. (Các thông số, pureXMLHostAddress và contextRoot, tham chiếu đến các tên phần tử trong định nghĩa tiểu trình, chúng tôi sẽ trình bày trong phần tiếp theo.) Hai chuỗi ký tự này được nối với /query/ và tên lệnh (insert, getDocumentByKey, update, delete, v.v...). Thí dụ, khi hàm insertJSON() gọi doPOST, URL là: http://xmlim.watson.ibm.com:9080/JSONUniversalServices/query/insert

- callback - Một tham chiếu đến một hàm mà sẽ được gọi khi yêu cầu trả về kết quả. Thí dụ, response là một hàm gọi ngược hiển thị trạng thái của một giao dịch, và displayJSONobj là một hàm gọi ngược hiển thị đối tượng JSON được trả về hoặc một thông báo nếu đối tượng đó rỗng.

- opt_params - Một đối tượng JavaScript chứa các thông số bổ sung cho cuộc gọi (phương thức HTTP và dữ liệu POST). Chúng ta đặt phương thức HTTP yêu cầu là gadgets.io.MethodType.POST để cho biết nó là một phương thức POST. Chúng tôi cũng chuyển dữ liệu gửi lên trong đối tượng postdata (được chỉ rõ như là một đối tượng chứa các cặp khóa/giá trị bằng cách sử dụng gadgets.io.encodeValues() để định dạng đầu vào args).

Hãy ghi lưu Liệt kê 1 dưới tên là PureJSON.js. Bạn sẽ xây dựng định nghĩa tiểu trình và nạp tệp tin này trong bước tiếp theo.

Định nghĩa tiểu trình OpenSocial

Lúc này bạn đã có JavaScript cần thiết, bạn sẽ tạo ra định nghĩa tiểu trình.

Tạo ra một tài liệu XML định nghĩa tiểu trình

Bạn cần một định nghĩa tiểu trình đơn giản. Liệt kê 2 cung cấp định nghĩa tiểu trình mẫu. Mỗi định nghĩa tiểu trình được bao bọc trong thẻ <Module>. <ModulePrefs> xác định thông tin và tính năng cơ bản của tiểu trình này.

Liệt kê 2. Định nghĩa XML cho tiểu trình OpenSocial

```
<?xml version="1.0" encoding="UTF-8"?>

<Module>

 <ModulePrefs title="Universal Services">
```

```
  <Require feature="opensocial-0.8"/>
  <Require feature="dynamic-height"/>
  <Require feature="minimessage"/>
 </ModulePrefs>
 <UserPref name="pureXMLHostAddress"
           display_name="PureXML Host Address"
           default_value="xmlim.watson.ibm.com:9080"/>
 <UserPref name="contextRoot"
           display_name="Context Root"
           default_value="JSONUniversalServices"/>

 <Content type="html">
 <![CDATA[
        <!-- Note: We will add more code and expand this section later  -->
  ]]>
 </Content>
</Module>
```

Điều đầu tiên bạn sẽ để ý thấy trong định nghĩa tiểu trình là thuộc tính <title> của phần tử <ModulePrefs>. Bạn có thể quy định các thuộc tính xác định thông tin cho tiểu trình như tiêu đề, tác giả, thư điện tử, v.v.... Nhằm phục vụ cho bản trình diễn này, chúng tôi sẽ chỉ sử dụng thuộc tính title. Tuy nhiên điều quan trọng là cần lưu ý rằng một số thùng chứa đòi hỏi các thuộc tính nhất định phải được cung cấp để triển khai sống (live). Bạn sẽ cần tham khảo tài liệu của nhà phát triển nền để tìm thông tin này.

Bạn cũng sẽ cung cấp các đặc tính mà tiểu trình đòi hỏi. Tiểu trình mẫu sẽ nạp các đặc tính sau đây:

- opensocial-0.8 - Tiểu trình này sử dụng các API cài đặt OpenSocial version 0.8 và do đó nó có thể được triển khai trên các thùng chứa có hỗ trợ đặc tả OpenSocial v0.8.

- dynamic-height - Với đặc tính này, nhà phát triển tiểu trình có khả năng thay đổi lại kích thước chiều cao tiểu trình khi nội dung được thêm vào hoặc loại bớt. Đặc tính này được gọi ra khi bạn hiển thị hoặc loại bỏ các thông báo giao dịch DB2 trong tiểu trình.

- minimessage - Đặc tính này cung cấp một tập hợp các API để tạo và hiển thị các thông báo cho những người sử dụng trong tiểu trình. Chúng ta sử dụng minimessage để tạo ra các thông báo tình trạng giao dịch DB2 trong thí dụ này.

Bạn cũng đã định nghĩa một tập hợp các sở thích của người sử dụng (<UserPref>) để thiết lập động điểm cuối Các Dịch vụ Phổ dụng pureXML và đường dẫn ngữ cảnh của ứng dụng mà tiểu trình sẽ được triển khai lên. Các phần tử <UserPref> đã định nghĩa được trưng ra như là đầu vào của người sử dụng trong giao diện tiểu

trình khi nó được biểu hiện. Người sử dụng sau đó có thể biên tập và sửa đổi các thiết lập này một cách phù hợp. Lưu ý rằng thuộc tính name của các phần tử <UserPref> ấy cũng chính là các tên được sử dụng trong hàm JavaScript doPOST trong Liệt kê 1 để nhận các chuỗi ký tự đã dùng để xây dựng URL của điểm cuối.

Thứ tư, phần nội dung <Content> định nghĩa kiểu nội dung của tiểu trình là HTML. Trong phần CDATA (dữ liệu ký tự), bạn định nghĩa chính nội dung của tiểu trình, ở đây một bảng HTML được tạo ra để nắm giữ đầu vào của người sử dụng, kết hợp với minimessage, các sở thích của người sử dụng, và các hàm JavaScript để trở thành một tiểu trình đang chạy. Bạn sẽ mở rộng phần này trong bước tiếp theo để xây dựng một trình khách kết nối với các dịch vụ pureXML.

Thêm nội dung vào tiểu trình

Trong phần CDATA của định nghĩa tiểu trình mô tả trong bước trước đây, bây giờ hãy bổ sung thêm bảng HMTL mẫu và các hàm JavaScript để nắm giữ đầu vào và các hành động của người sử dụng.

Đầu tiên, thêm phiếu định kiểu sẽ dùng trong bảng HTML như trong Liệt kê 3.

Liệt kê 3. Phiếu định kiểu được sử dụng cho bảng

```
<style type="text/css">
 table.layout {border:0; width:50%;}
 td.green {background-color:#BFFF80;
```

```
   font-family:sans-serif, verdana;}

td.white {background-color:#FFFFFF;

   font-family:sans-serif, verdana;}

th.green {background-color:#BFFF80;

   font-family:sans-serif, verdana;}

th.white {background-color:#FFFFFF;

   font-family:sans-serif, verdana;}

td.row-bright{background-color:#FFFFBF;

   font-family:sans-serif, verdana;

   text-align:center;}

td.row-dark {background-color:#FFFF8C;

   font-family:sans-serif, verdana;

   text-align:center;}

tr.row-bright {background-color:#FFFFBF;

   font-family:sans-serif, verdana;}

tr.row-dark {background-color:#FFFF8C;

   font-family:sans-serif, verdana;}
```

```
  tr.empty {background-color:#FFFFFF;
    height: 10px;}
  tr.empty-small {background-color:#FFFFFF;
    height: 5px;}
</style>
```

Thứ hai, bao gồm thêm tệp tin JavaScript, PureJSON.js từ Liệt kê 1. Vào lúc tiểu trình được nạp, PureJSON.js sẽ được gộp vào trang này, làm cho tất cả các hàm yêu cầu POST sẵn dùng đối với các hành động trong bảng HTML.

Liệt kê 4. Nạp tệp tin JavaScript

```
<script type="text/javascript" src="./PureJSON.js"></script>
```

Thứ ba, bổ sung một thẻ <script> khác như trong Liệt kê 5. Mỗi hàm JavaScript khớp với một hành động của người sử dụng khai báo trong bảng HTML (mà bạn sẽ định nghĩa trong phần tiếp theo), và gọi một trong các hàm được định nghĩa trong PureJSON.js từ Liệt kê 1. Khi khối mã này được nạp, một đối tượng

minimessage được tạo ra bằng cách sử dụng new gadgets.MiniMessage(_MODULE_ID_). Hàm gadgets.util.registerOnLoadHandler(gadgets.window.adjustHeight), nằm ở dưới cùng của thẻ, được gọi để đăng ký tiểu trình với thùng chứa OpenSocial, báo cho nó sửa đổi khung cho vừa với nội dung tiểu trình vào lúc biểu hiện. Trong phạm vi thẻ <script> này, chúng ta cũng viết đè lên các hàm response và displayJSONobj để hiển thị kết quả truy vấn cho người sử dụng.

Liệt kê 5. Bổ sung các hàm tiểu trình

```
<script type="text/javascript" src="./PureJSON.js"></script>

<script type="text/javascript">

var msg = new gadgets.MiniMessage(__MODULE_ID__);

function displayJSONobj(obj)    {
  var str = "The returned record is empty, it might not exist";
  if(obj.text!=""){
```

```
        str = obj.text;
  }
  msg.createDismissibleMessage(str);
  gadgets.window.adjustHeight();
};

function callGetJSONDoc()       {
  getJSONDocumentByKey(document.getElementById("key1").value);
};

function getJSONDocumentByKeyReturn(obj)     {
  var str = obj.text;
  msg.createDismissibleMessage(str);
  gadgets.window.adjustHeight();
};

function callInsertJSON()         {
```

```
    insertJSON(document.getElementById("key2").value,
        document.getElementById("document1").value);
  };

  function callUpdateJSON()        {
    updateJSON(document.getElementById("key3").value,
        document.getElementById("document2").value);
  };

  function callDeleteDoc()          {
    deleteDocument(document.getElementById("key4").value);
  };

 function response(obj)   {
    var str = gadgets.json.parse(gadgets.util.unescapeString(obj.text));

if(str.updateCount == 1){
```

```
  var successMsg = msg.createDismissibleMessage(
      "Received returned code = 1. Transaction successful!");
  successMsg.style.color = "green";
 }
 else     {
  var failMsg = msg.createDismissibleMessage(
   "Did not receive returned code = 1. Transaction may have failed!");
    failMsg.style.color = "red";
 }
 gadgets.window.adjustHeight();
};

gadgets.util.registerOnLoadHandler(gadgets.window.adjustHeight);

</script>
```

Cuối cùng, bổ sung bảng HTML này để nắm giữ đầu vào và hành động của người sử dụng. Liệt kê 6 hiển thị mã HTML. Hãy ghi lưu định nghĩa tiểu trình với tên JSONclient.xml. Ảnh chụp màn hình trong Hình 1 là biểu hiện của bảng này.

Liệt kê 6. Bảng HTML

```
<table class="layout" cellspacing="2">
 <tr>
  <th class="green">Web Service</th>
  <th class="green" colspan="2">Input</th>
  <th class="green">Action</th>
 </tr>
 <tr class="row-bright">
  <td>getPrimaryKeys</td>
  <td colspan="2" align="center">none</td>
  <td align="center">
   <input type="submit" value="Invoke" onClick="getPrimaryKeys()"/>
  </td>
```

```
 </tr>
<tr class="row-dark" >
 <td>getJSONDocumentByKey</td>
 <td align="right">ID:</td>
 <td align="center"><input type="text" id="key1" size="40" /></td>
 <td align="center">
  <input type="submit" value="Invoke" onClick="callGetJSONDoc()"/>
 </td>
</tr>
<tr class="row-bright">
  <td rowspan="2">insertJSON</td>
  <td align="right">ID:</td>
  <td align="center"><input type="text" id="key2" size="40" /></td>
  <td align="center" rowspan="2">
    <input type="submit" value="Invoke" onClick="callInsertJSON()"/>
  </td>
</tr>
```

```
<tr class="row-bright">
  <td align="right">Document:</td>
  <td align="center">
    <textarea id="document1" cols="30" rows="5" ></textarea>
  </td>
</tr>
<tr class="row-dark" >
  <td rowspan="2">updateJSON</td>
  <td align="right">ID:</td>
  <td align="center"><input type="text" id="key3" size="40" /></td>
  <td rowspan="2" align="center">
        <input type="submit" value="Invoke" onClick="callUpdateJSON()"/>
  </td>
</tr>
<tr class="row-dark">
  <td align="right">Document:</td>
  <td align="center">
```

```
      <textarea id="document2" cols="30" rows="5" ></textarea>
    </td>
  </tr>
  <tr class="row-bright">
    <td>deleteDocument</td>
    <td align="right">ID:</td>
    <td align="center"><input type="text" id="key4" size="40" /></td>
    <td align="center">
      <input type="submit" value="Invoke" onClick="callDeleteDoc()"/>
    </td>
  </tr>
</table>
```

Triển khai và kiểm thử

Bây giờ bạn sẽ triển khai tiểu trình và sau đó là kiểm thử tiểu trình đó.

Triển khai tiểu trình

Để kiểm thử Các Dịch vụ Phổ dụng mới này, bạn chỉ cần triển khai cả PureJSON.js và JSONclient.xml lên một máy chủ HTTP. Khi mà các tập tin đã sẵn sàng thông qua một URL, thùng chứa OpenSocial sẽ có thể lấy ra đặc tả tiểu trình và biểu hiện nội dung tương ứng. Một máy chủ OpenSocial có một số tùy chọn đối với cả triển khai cục bộ đang sẵn có thông qua các dự án mã nguồn mở cũng như các nền hệ thống chủ chứa bên ngoài. Để đơn giản hoá, bạn sẽ triển khai tiểu trình mẫu này lên hộp cát (sandbox) iGoogle, là thùng chứa OpenSocial.

1. Đăng nhập http://www.google.com/ig/sandbox

2. Khi đã đăng nhập, chuyển đến liên kết **Add stuff** trong phần trên bên phải của trang. Thao tác này sẽ liên kết đến một trang cung cấp các khả năng tìm kiếm các tiểu trình hiện đã đăng ký. Xem Hình 2. Chú ý rằng URL phải truy cập được từ Web. Thí dụ, nếu bạn đang chạy phía sau một tường lửa, thì bản triển khai của bạn có thể không làm việc vì thùng chứa OpenSocial, trong trường hợp này là iGoogle, không thể tải và xử lý được định nghĩa tiểu trình, (Xem bản lớn hơn của Hình 2).

Hình 2. Bổ sung các ứng dụng cho hộp cát iGoogle của bạn

3. Tiếp theo bạn cần bổ sung tiểu trình mới này bằng URL. Trong phần dẫn hướng bên trái, bạn sẽ thấy một liên kết đến **Add feed or gadget** (Thêm nguồn cấp hoặc tiểu trình). Liên kết này sẽ mở ra một hộp thoại mà bạn sẽ cần để nhập vào URL đầy đủ cho JSONclient.xml. Xem Hình 3.

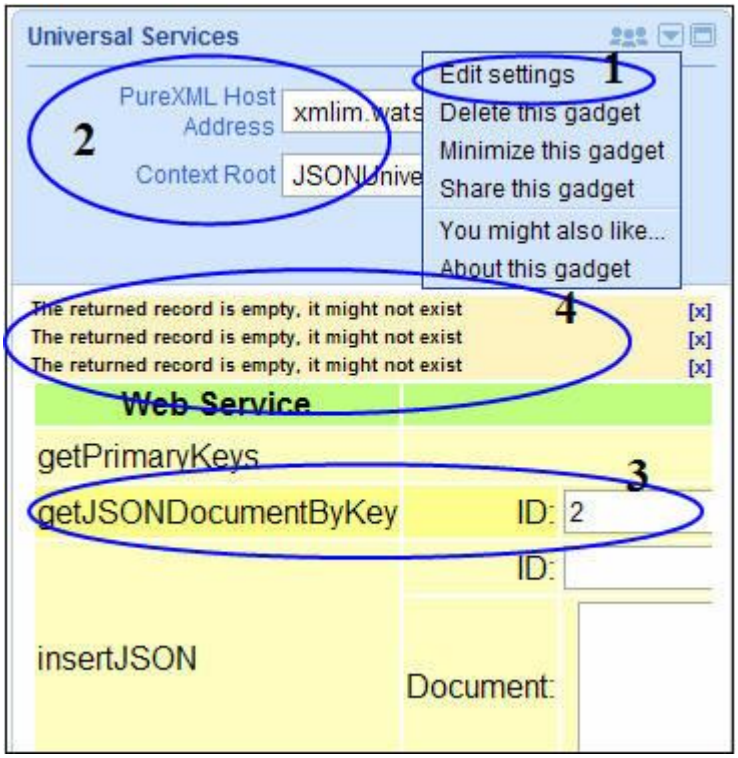
Hình 3. Bổ sung một tiểu trình OpenSocial cho hộp cát iGoogle

4. Bây giờ bạn sẽ có thể quay lại trang chủ iGoogle của bạn và xem tiểu trình vừa được bổ sung trên trang của bạn.

Kiểm thử tiểu trình

Hình 4 cho thấy tiểu trình đã được biểu hiện trong iGoogle.

Hình 4. Tiểu trình được biểu hiện trong iGoogle



1. Để biên tập các thiết lập cài đặt tiểu trình, chọn biểu tượng hình tam giác để mở danh sách thả xuống các tùy chọn. Chọn **Edit settings**.

2. Cập nhật địa chỉ máy chủ chứa pureXML và đường dẫn gốc ngữ cảnh để chỉ đến các dịch vụ phổ dụng của bạn.

3. Trong trường getJSONDocumentByKey, nhập vào một khoá tài liệu để lấy ra một bản ghi JSON, sau đó nhấn **Invoke.** Để kiểm thử các giao dịch khác, lặp lại bước này.

4. Minimessage hiển thị kết quả. Để loại bỏ các thông báo, chọn **x** ở cuối hàng này.

Kết luận

Các bài khác trong loạt bài này

- Phần 1: Lưu trữ và truy vấn JSON với pureXML của DB2
- Phần 2: Tạo Các Dịch vụ Phổ dụng cho pureXML để đưa ra JSON

Trong bài này, chúng tôi đã cung cấp một tập hợp mã mẫu, nó di trú ứng dụng Web kiểm thử truyền thống thành một ứng dụng khả chuyển và có thể cấu hình được, có thể triển khai nhanh chóng đến một số nền trên Web. Mặc dù ứng dụng tự nó tương đối đơn giản, mẫu tham khảo này có thể được sử dụng để lắp ráp và làm bản mẫu các ứng dụng mới một cách nhanh chóng. Trình khách tiểu trình OpenSocial cung cấp cho người sử dụng một cách tiện lợi để kết nối động với bất kỳ Các dịch vụ Phổ dụng pureXML nào thông qua sự hỗ trợ của UserPrefs. Với POST makeRequest, các nhà phát triển tiểu trình chỉ cần tạo ra các truy vấn kết nối với các dịch vụ để thực hiện các giao dịch cơ sở dữ liệu của mình. Người tiêu dùng tiểu trình có thể bổ sung tiểu trình vào bất kỳ thùng chứa OpenSocial nào không phân biệt vị trí nơi nó được triển khai vật lý và chứa ở đó, tránh cho người tiêu dùng không phải quản lý nội dung và cơ sở hạ tầng.