https://www.flickr.com/photos/designsbykari/4832871903/

Hello, I'm **Daniel Gustafsson**

Pivotal

I work at

with

and

GREENPLUM DATABASE

commit 8c1de5fb0010ae712568f1706b737270c3609bd8
Author: Peter Eisentraut <peter_e@gmx.net>
Date: Thu Dec 21 16:05:16 2006 +0000

    Initial SQL/XML support: xml data type and
    initial set of functions.

https://www.flickr.com/photos/marals/3821484254/

```
commit 5384a73f98d9829725186a7b65bf4f8adb3cfaf1
Author: Robert Haas <rhaas@postgresql.org>
Date:   Tue Jan 31 11:48:23 2012 -0500
```

Built-in **JSON** data type.


n/wp-content/uploads/2014/01/funny-hipster-guy-stereotype-

2017

1 + 1

Network Working Group
Internet-Draft
Expires: November 3, 2011

B. Muschett
R. Salz
M. Schenker
IBM

May 2, 2011

JSONx, an XML Encoding for JSON
draft-rsalz-jsonx-00.txt

Abstract

This document specifies a mapping between JSON (RFC 4627) and XML. The mapping maintains a high degree of fidelity. It is used by several IBM products.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

JSONx

JSONx is an IBM® standard format to represent JSON as XML. The appliance converts JSON messages that are sp
The appliance provides a style sheet that you can use to convert JSONx to JSON.

JSONx conversion rules specify how a DataPower® service converts a JSON structure to JSONx (XML).

The DataPower appliance provides the default `jsonx2json.xsl` style sheet for converting JSONx to JSON. This

Google Trends | Compare

jsonx
Search term

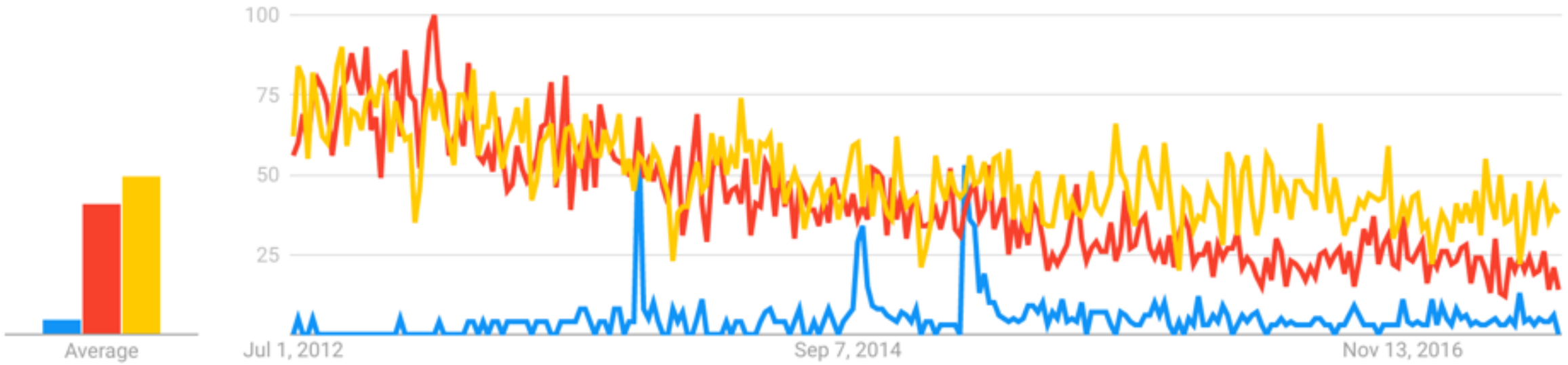
CONFIG.SYS
Search term

asn.1
Search term

+

Worldwide | Past 5 years | All categories | Web Search

Interest over time



Google Trends | Compare

jsonx
Search term

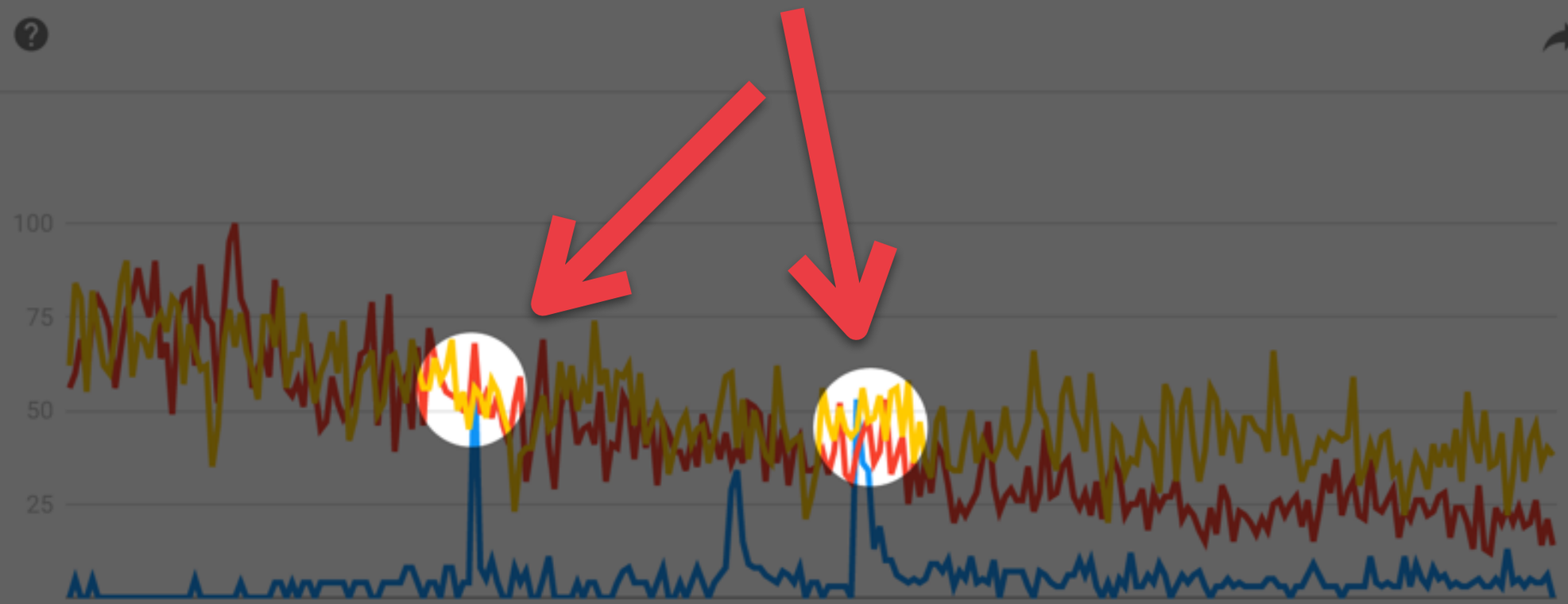
CONFIG.SYS
Search term

asn.1
Search term

+

Worldwide | Past 5 years | All categories | Web Search

Interest over time



```
{
  "name":"John Smith",
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021,
  },
  "phoneNumbers": [
    "212 555-1111",
    "212 555-2222"
  ],
  "additionalInfo": null,
  "remote": false,
  "height": 62.4,
  "ficoScore": " > 640"
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<json:object xsi:schemaLocation="http://www.datapower.com/schemas/
json jsonx.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:json="http://www.ibm.com/xmlns/prod/2009/jsonx">
  <json:string name="name">John Smith</json:string>
  <json:object name="address">
    <json:string name="streetAddress">21 2nd Street</json:string>
    <json:string name="city">New York</json:string>
    <json:string name="state">NY</json:string>
    <json:number name="postalCode">10021</json:number>
  </json:object>
  <json:array name="phoneNumbers">
    <json:string>212 555-1111</json:string>
    <json:string>212 555-2222</json:string>
  </json:array>
  <json:null name="additionalInfo" />
  <json:boolean name="remote">false</json:boolean>
  <json:number name="height">62.4</json:number>
  <json:string name="ficoScore"> > 640</json:string>
</json:object>
```

=

👊💙🤘

```
postgres=# select jsonx_build_object('{"foo":"bar"}');
                 jsonx_build_object
-----------------------------------------------------
 <json:string name="foo">bar</json:string>+

(1 row)
```

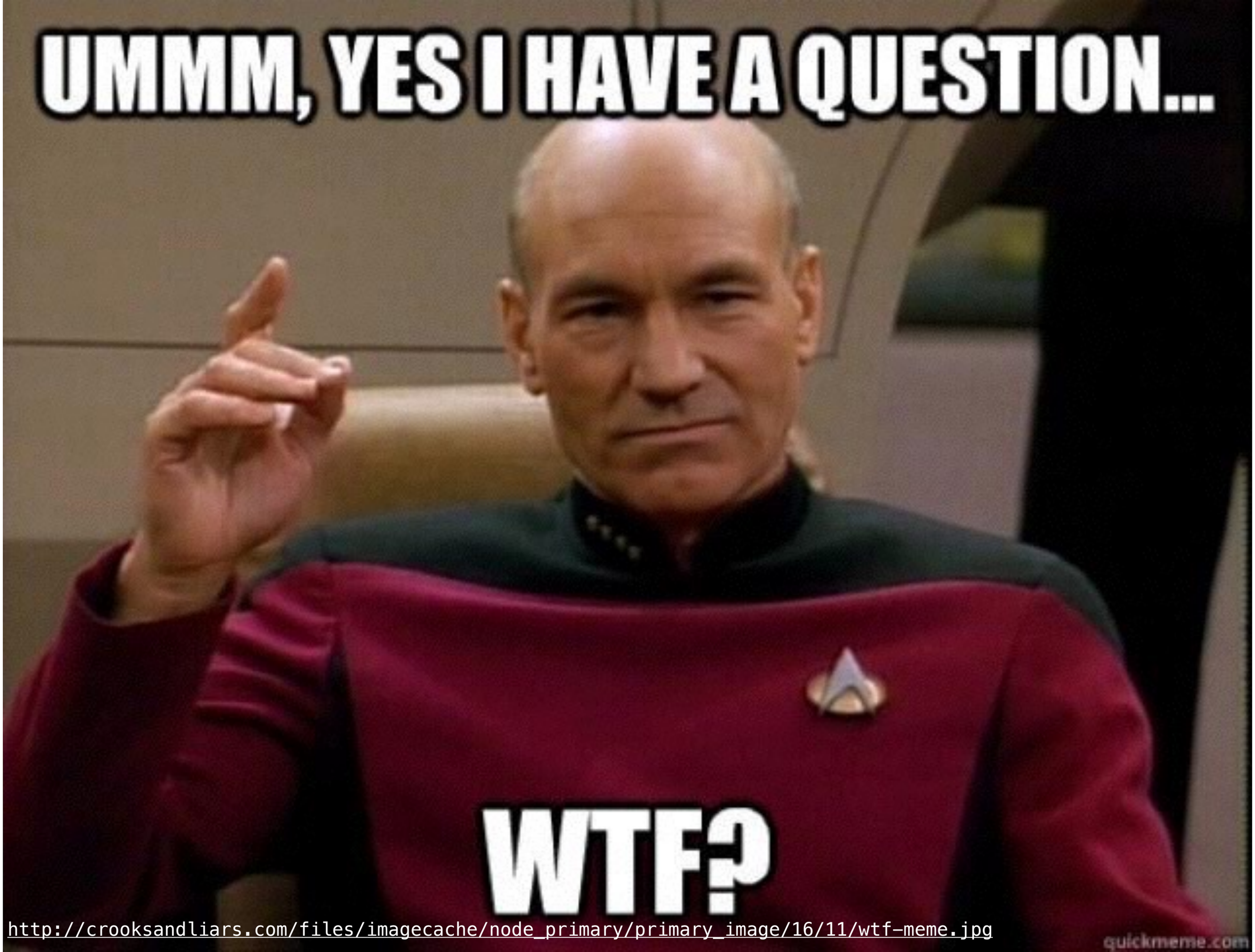
http://howfuckedismydatabase.com/nosql/

```
postgres=# select jsonx_build_object('{"answer":42}');
                  jsonx_build_object
-------------------------------------------------------
 <json:number name="answer">42</json:number>+

(1 row)
```

```
postgres=# select jsonx_build_object('{"zip": 11249,
"phone":["1-800-AWESOME","1-800-TOFURKY"]}');

                jsonx_build_object
-----------------------------------------------
 <json:number name="zip">11249</json:number>+
 <json:array name="phone">                  +
   <json:string>1-800-AWESOME</json:string> +
   <json:string>1-800-TOFURKY</json:string>  +
 </json:array>                              +

(1 row)
```

```
postgres=# select jsonx_build_document('{"languages":
["algol","fortran"]}');

                        jsonx_build_document
-----------------------------------------------------------------------
<json:object xsi:schemaLocation="http://www.datapower.com/schemas/
json jsonx.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:json="http://www.ibm.com/xmlns/prod/2009/jsonx">
   <json:array name="languages">
     <json:string>algol</json:string>
     <json:string>fortran</json:string>
   </json:array>
 </json:object>

(1 row)
```

```
postgres=# select xmlexists('//*[@name]' PASSING
jsonx_build_document('{"zip": 11249, "phone":["1-800-
AWESOME","1-800-TOFURKY"]}'));

 xmlexists
___________
 t

(1 row)
```



http://crooksandliars.com/files/imagecache/node_primary/primary_image/16/11/wtf-meme.jpg

In all seriousness..

commit a570c98d7fa0841f17bbf51d62d02d9e493c7fcc
Author: Andrew Dunstan <andrew@dunslane.net>
Date: Fri Mar 29 14:12:13 **2013** -0400

Add new JSON processing functions and parser API.

```
pg_parse_json();
```

```
JsonLexContext  *lex;
JsonxState      *state;
JsonSemAction   *sem;

text *json = cstring_to_text(json_text);

sem = palloc0(sizeof(JsonSemAction));
state = palloc(sizeof(JsonxState));

sem->semstate = (void *) state;


lex = makeJsonLexContext(json, true);

pg_parse_json(lex, sem);
```

```
JsonLexContext  *lex;
JsonxState      *state;
JsonSemAction   *sem;

text *json = cstring_to_text(json_text);

sem = palloc0(sizeof(JsonSemAction));
state = palloc(sizeof(JsonxState));

sem->semstate = (void *) state;
sem->object_field_start = object_field_start;
sem->scalar = get_scalar;
lex = makeJsonLexContext(json, true);

pg_parse_json(lex, sem);
```

```c
typedef struct JsonxState
{
  char *key;
  char *value;
} JsonxState;
```

```c
static void
get_scalar(void *state, char *token,
           JsonTokenType type)
{
  JsonxState *s = (JsonxState *) state;

  switch(type)
  {
    case JSON_TOKEN_STRING:
      s->value = pstrdup(token);
      break;
    case JSON_TOKEN_NUMBER:
      ...
      break;
    default:
      break;
  }
}
```

```c
static void
object_field_start(void *state, char *name,
                   bool isnull)
{
  JsonxState *s = (JsonxState *) state;

  s->key = pstrdup(name);
}
```

```
$ git diff origin/master --stat
 src/backend/utils/adt/xml.c     | 227 ++++++++++++++++++++++++++++++++++++++++
 src/include/catalog/pg_proc.h   |   3 ++
 src/include/utils/xml.h         |   4 +++
 3 files changed, 234 insertions(+)
```

JSON parsing is easy, build cool(er) stuff!

Thanks / I'm sorry

daniel@yesql.se @d_gustafsson