itspeople

# Database Links 2.0
# Access Remote Tables with REST and JSON

February 10, 2016

**Robert Marz**
Technical Architect

# Robert Marz

**Client**
- Senior Technical Architect with database centric view of the world

**its-people**
- Portfolio Manager Database Technologies
- Partner

**DOAG**
- Active Member Database Community in charge of High Availability topics


@RobbieDatabee

www.its-people.de/blog

Robert.Marz @its-people.de

# its-people auf einen Blick


## Unternehmensphilosophie

- Zusammenschluss selbständiger IT-Experten unter einer gemeinsamen Marke
- Bündelung von umfassendem IT-Wissen
- Partnerschaft auf Augenhöhe
- Gemeinsam stark!

## Unternehmensdaten

- Gründung: 2003
- Anzahl der Experten: 70


## Leistungsangebot

**DatenbankTechnologie**
Plattform & Anwendungsentwicklung

**Business Intelligence & DWH**
Architektur, Design & Implementierung

**Customer Data Integration**
Beratung, Konzepte & Projekte

**Best Practice ERP**
Mediation, Integration, Migration

IT Management Services
Projekt-, Qualitäts-, Test-Mgmt

Beratung, Planung, Konzeption, Implementierung, Support, Krisenbewältigung


## Stärken

- Kompetenz, Exzellenz und Qualität aus praktischer Erfahrung
- Partnerschaft auf Gegenseitigkeit
- Kundenorientierung auf höchster Ebene
- Wertekodex als Geschäftsgrundlage
- Teams für komplexe Projektaufgaben
- Nähe zum Kunden
- Dienstleistungen von Menschen für Menschen

**its-people**
Ihr Partner für Projekte im Umfeld Datenbanken, Data Warehouse, Business Intelligence, Customer Data Integration und ERP


**its-people Frankfurt**
Lyoner Str. 44-48, 60528 Frankfurt
Tel.: +49 (69) 2475 210-0
E-Mail: frankfurt@its-people.de

**its-people ERP Beratungsgesellschaft mbH**
Lyoner Str. 44-48, 60528 Frankfurt
Tel.: +49 (69) 2475 19-80
E-Mail: erp@its-people.de

**its-people Köln**
Hohenzollernring 57, 50672 Köln
Tel.: +49 (221) 1602 5204
E-Mail: koeln@its-people.de

**its-people München**
Lichtenbergstr. 8, 85748 Garching
Tel.: +49 (89) 5484 2401
E-Mail: muenchen@its-people.de

**its-people Hamburg**
Am Strohhause 31, 20097 Hamburg
Tel.: +49 (40) 2360 8808
E-Mail: hamburg@its-people.de

# Database Links – the Issues

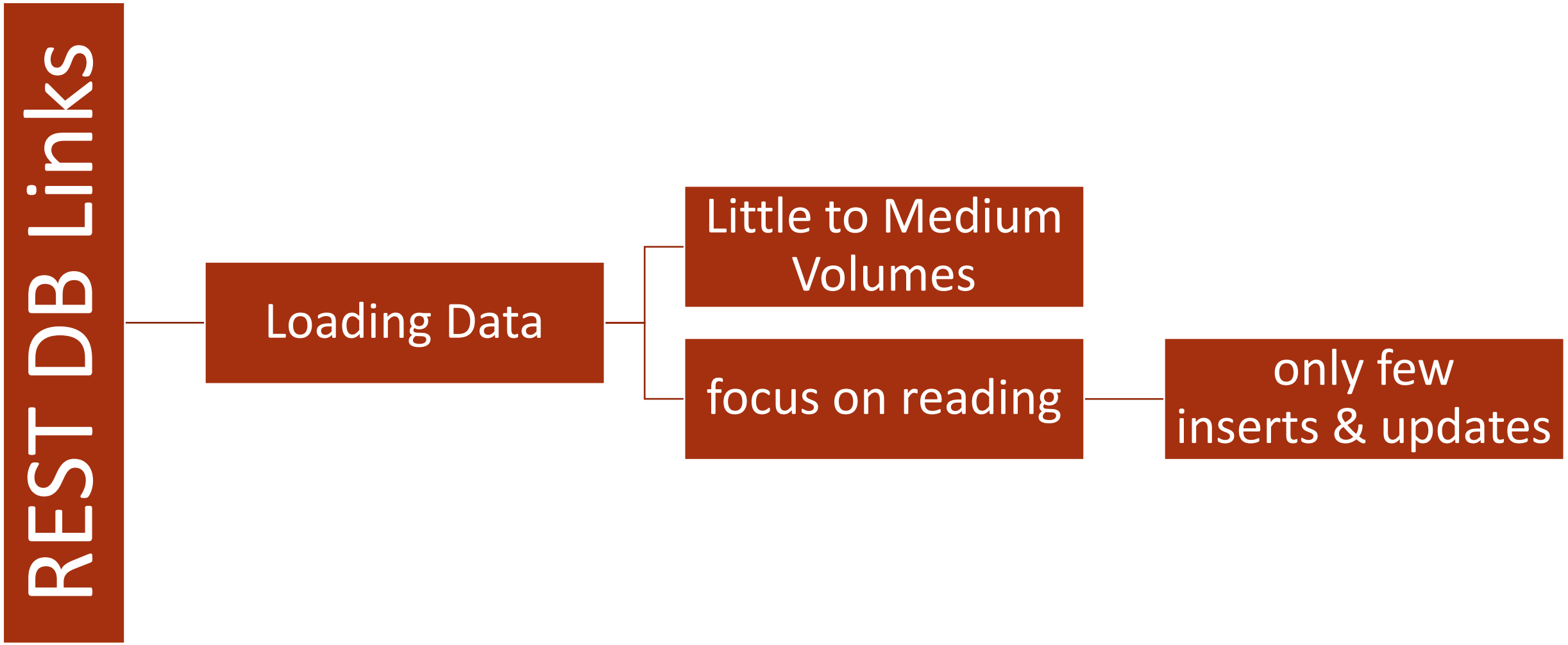
## Database Links

Are trapped inside the Oracle world

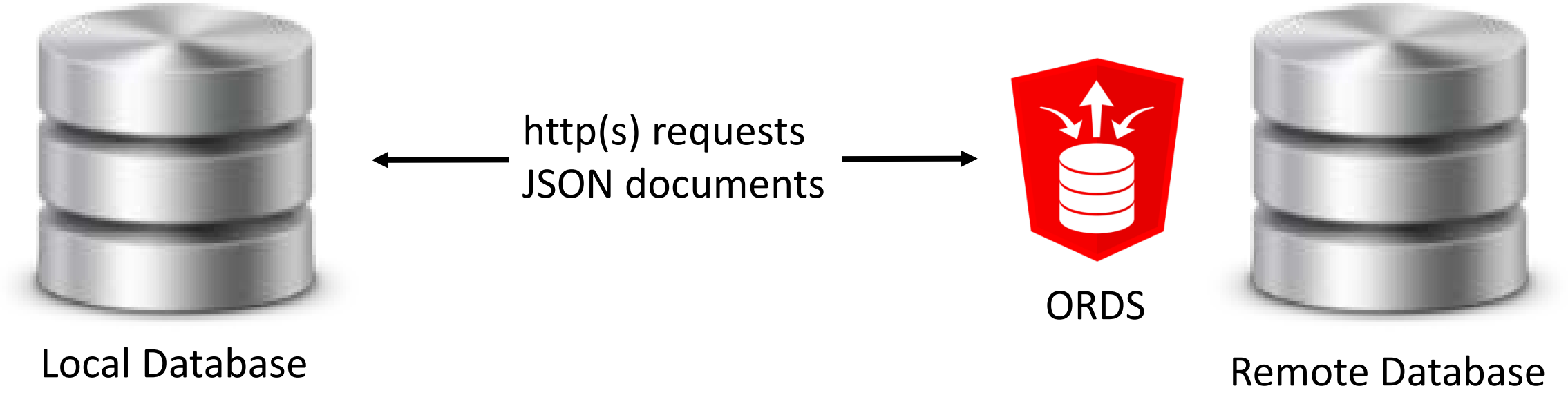
Only work within local or private networks: SQLNet

Objects are treated as local

# Database Links 2.0 – Use Case



# Database Links 2.0 – Architecture



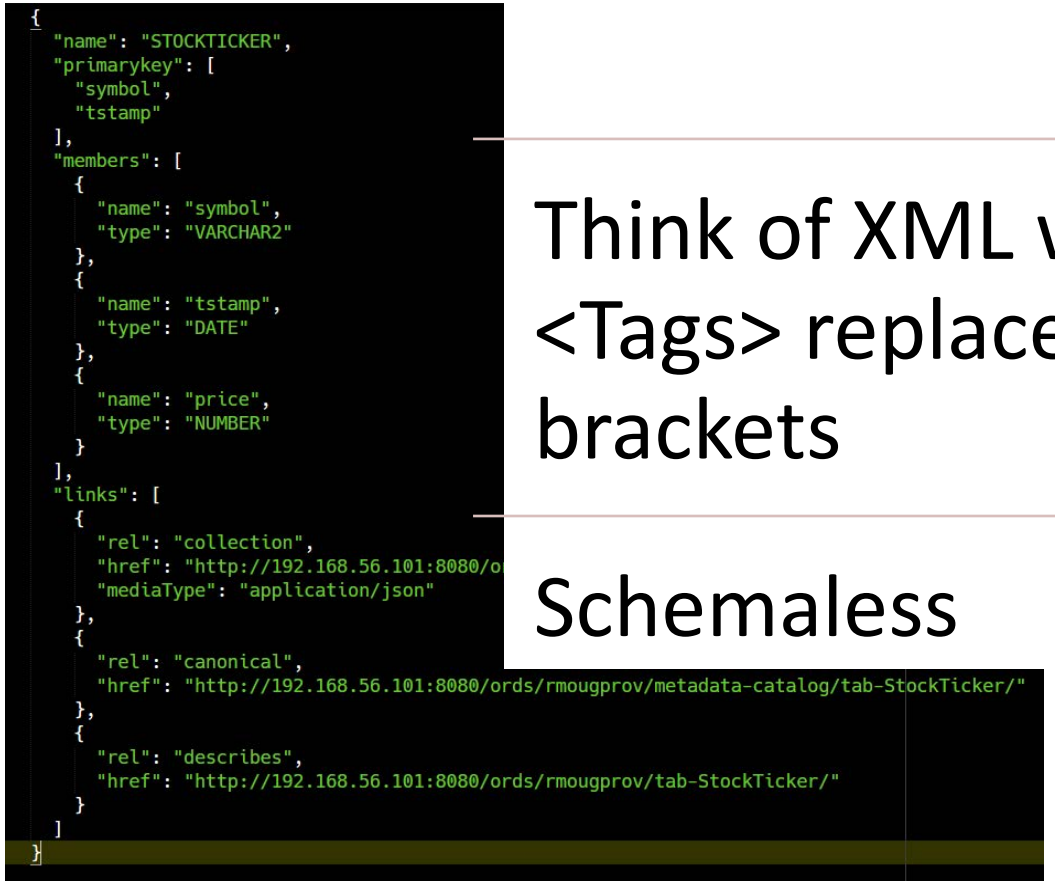
# What is REST?

| | | |
|---|---|---|
| REST | Representational state transfer | |
| | programming paradigm | distributed systems |
| | | Web services. |
| RESTful Applications | implements 6 constraints most important: | Uniform Interface (API via URIs) |
| | | Stateless |
| | | Cacheable |
| Implementation | Transport protocol | http(s) |
| | content | JSON Documents |

# What is JSON?

# JSON

**J**ava **S**cript
**O**bject **N**otation




- Think of XML with <Tags> replaced by brackets
  - “Key“:“Value“ Pairs
  - {} – Groupings
  - [] - Arrays
- Schemaless
  - no constraints for your implementation
  - Developers hell when dealing with documents not produced by your code

# Oracle REST Data Services ORDS (1/2)

## Java

Evloved from APEX Listener

### Deploy in Application Server

- Tomcat
- Glassfish
- WebLogic

### Standalone mode

- Brings own http-server

## Links

### Installation

- Install ORDS in less than 5 Minutes by Colm Divilly (@cdivilly): http://blog.cdivilly.com/2015/03/11/install-ords-3.0.0/

### Official Homepage

- http://www.oracle.com/technetwork/developer-tools/rest-data-services/overview/index.html

### Documentation

- https://docs.oracle.com/cd/E56351_01/doc.30/e56293/develop.htm

### Video

- Oracle REST Data Services by Oracle Database Development Tools: https://www.youtube.com/watch?v=8XIbFRm-c6w

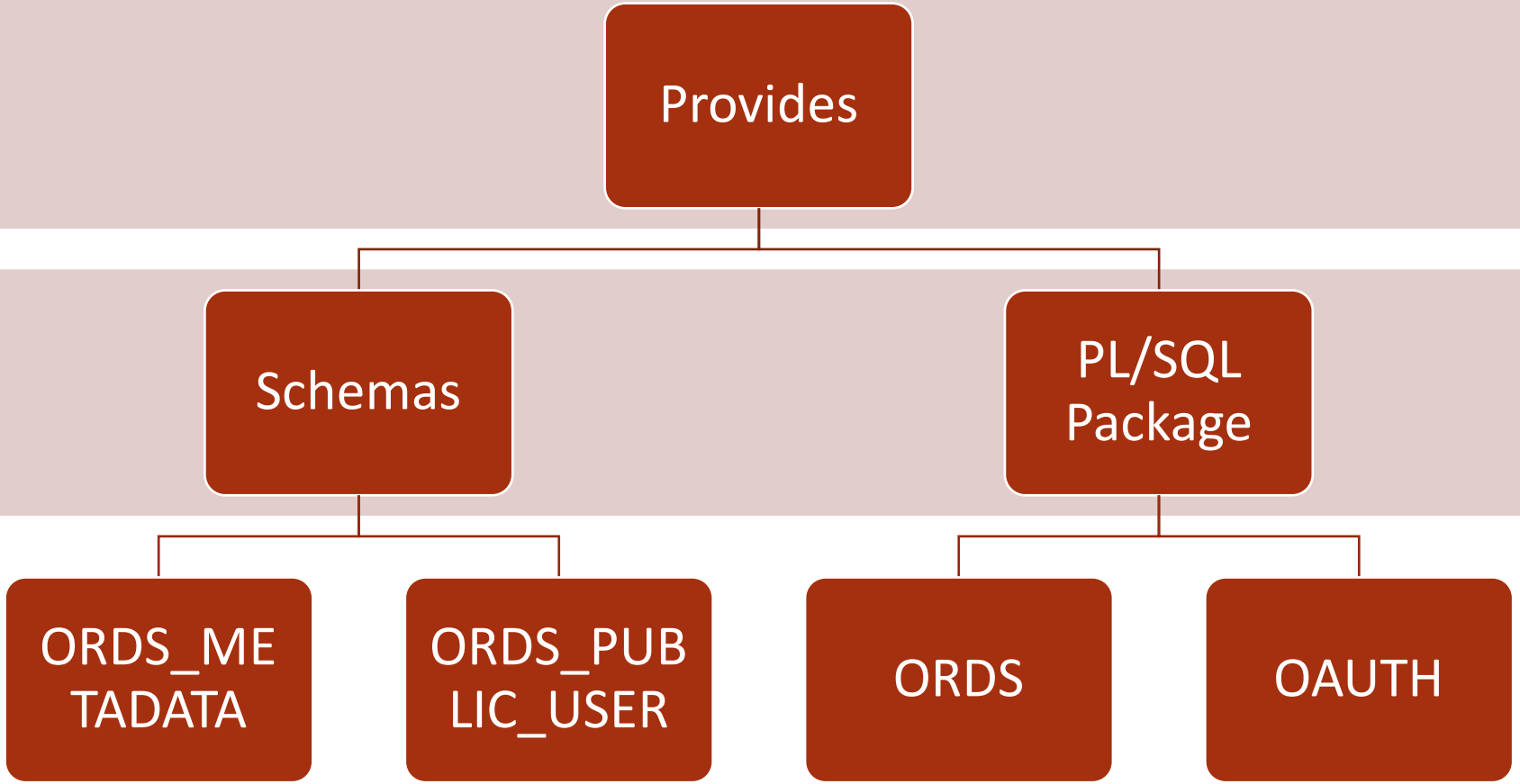
# Oracle REST Data Services ORDS (2/2)

## REST

- Restful Services: Implement full
- SODA: Simple Oracle Document Access

## AutoRest

- Rest enable tables
  - very easy
  - some limitations



# Anatomy of a ORDS AutoREST URL

ORDS Base Table Parameter

http://192.168.56.101:8080/ords/rmougprov/tab-StockTicker/?offset=55&limit=100

/ORCL,2016-01-08

PKCol1,PKCol2

Protocol Host:Port Schema or Primary Key

| HTTP Method | ORDS AutoREST Action |
|---|---|
| GET | Retrieve Data – Single Row or Rowset |
| PUT | Insert or Modify Row |
| POST | Bulk Insert csv-data |
| DELETE | Delete Row |

# DEMO Environment


**OTN Developer Day VM**

- Virtual Box Appliance
  - Oracle Linux 7
  - Oracle Database 12c EE (12.1.0.2 with In-Memory Option)
  - Oracle Application Express 5.0
  - Oracle REST Data Services 3.0.3
  - http://www.oracle.com/technetwork/database/enterprise-edition/databaseappdev-vm-161299.html

**Schemas**

- rmougCons: Consumer: Local Database
- rmougProv: Provider: Remote Database

**Client**

- Browser: Google Chrome
- REST Client: Insomnia
- SQL Developer
- sqlcl

# DEMO 00: cleanup & setup


```
connect rmougProv/rmoug;

begin
  ords.drop_rest_for_schema('RMOUGPROV');
end;
/

commit;

connect sys/oracle as sysdba


drop user rmougProv cascade;
drop user rmougCons cascade;

-- Provider ( Data Source )
grant connect, resource, unlimited tablespace to rmougProv
identified by rmoug;

-- Consumer ( Origin of the link )
grant connect
    , resource
    , unlimited tablespace
    , create view
    to rmougCons identified by rmoug;

select id, parsing_schema from all_ords_schemas;
```

Script: „00 - dba - cleanup.sql“

```
drop table StockTicker purge;
create table StockTicker
  ( symbol    varchar2(20)  not null
  , tstamp    date          not null
  , price     number
  , constraint pk_StockTicker primary key (symbol, tstamp)
  );

ALTER SESSION SET TIME_ZONE = 'MST';
insert into StockTicker ( symbol, tstamp, price)
select symbol, tstamp, startPrice + rn * Dev price
from ( select 'ORCL' symbol
            , 35 startPrice
            , 1/4 Dev
         from dual
       union all
       select 'TDC' symbol
            , 24 startPrice
            , -1/8 Dev
         from dual
     ) smbls
  , ( select rownum -1 rn
           , (trunc(sysdate, 'hh') + rownum/(24*60))-
1/(24*60) tstamp
        from dual
      connect by rownum < 62
    ) timestream;
Commit;
```

Script: „00 - rmougProv - setup.sql“

# DEMO 01: ords_enable


```
begin
  ords.enable_schema;
  commit;  -- This commit is important!
end;
/

begin
  ords.enable_object (
    P_ENABLED        => true,
    P_SCHEMA         => 'RMOUGPROV',
    P_OBJECT         => 'STOCKTICKER',
    P_OBJECT_TYPE    => 'TABLE',
    P_OBJECT_ALIAS   => 'tab-StockTicker',
    P_AUTO_REST_AUTH => false
  );
  commit;  -- This commit is important,
too!
end;
/
```

Script: „01 - rmougProv - ords_enable.sql“

# DEMO 02: queries


```
select *
  from StockTicker
order by tstamp, symbol
/


select symbol, count(*)
  from StockTicker
 group by symbol
/
```

Script: „02 - rmougProv – queries .sql“

# DEMO 02a: REST requests 1 Get StockTicker


RMOUG Demo Get StockTicker

GET http://192.168.56.101:8080/ords/rmougprov/tab-StockTicker/?offset=55&limit=100

BODY PARAMS AUTH HEADERS

HTTP method cannot send a request body.

RESPONSE HEADERS 7 KB 1975 ms SUCCESS 200


```
{
    "items": [
        {
            "symbol": "ORCL",
            "tstamp": "2016-01-19T02:43:00Z",
            "price": 45.75,
            "links": [
                {
                    "rel": "self",
                    "href": "http://192.168.56.101
                        :8080/ords/rmougprov/tab
                        -StockTicker/ORCL,2016-01
                        -19T02%3A43%3A00Z"
                }
            ]
        },
        {
            "symbol": "TDC",
            "tstamp": "2016-01-19T02:43:00Z",
            "price": 18.625,
            "links": [
                {
                    "rel": "self",
                    "href": "http://192.168.56.101
```

# DEMO 02a: REST requests 2 Get RMOUG Schema Metadata


RMOUG Demo Get RMOUG Schema Metadata

GET http://192.168.56.101:8080/ords/rmougprov/metadata-catalog/

BODY PARAMS AUTH HEADERS

HTTP method cannot send a request body.

RESPONSE HEADERS 495 B 165 ms SUCCESS 200


```
{
    "items": [
        {
            "name": "STOCKTICKER",
            "links": [
                {
                    "rel": "describes",
                    "href": "http://192.168.56.101
                        :8080/ords/rmougprov/tab
                        -StockTicker/"
                },
                {
                    "rel": "canonical",
                    "href": "http://192.168.56.101
                        :8080/ords/rmougprov/metadata
                        -catalog/tab-StockTicker/",
                    "mediaType": "application/json"
                }
            ]
        }
    ],
    "hasMore": false,
    "limit": 25,
    "offset": 0,
    "count": 1
```

# DEMO 02a: REST requests 3 Get StockTicker Metadata


RMOUG Demo Get StockTicker Metadata

GET http://192.168.56.101:8080/ords/rmougprov/metadata-catalog/tab-StockTicker/

BODY PARAMS AUTH HEADERS

HTTP method cannot send a request body.

RESPONSE HEADERS 492 B 898 ms SUCCESS 200


```
{
    "name": "STOCKTICKER",
    "primarykey": [
        "symbol",
        "tstamp"
    ],
    "members": [
        {
            "name": "symbol",
            "type": "VARCHAR2"
        },
        {
            "name": "tstamp",
            "type": "DATE"
        },
        {
            "name": "price",
            "type": "NUMBER"
        }
    ],
    "links": [
        {
            "rel": "collection",
            "href": "http://192.168.56.101:8080/ords
                /rmougprov/metadata-catalog/"
```

# DEMO 02a: REST requests 4 Modify StockTicker Entry


RMOUG Demo Modify StockTicker Entry

PUT http://192.168.56.101:8080/ords/rmougprov/tab-StockTicker/ORCL,2016-01-08T21%3A01%3A00Z

BODY PARAMS AUTH HEADERS JSON

```
{
    "price": 250
}
```

RESPONSE HEADERS 475 B 1961 ms SUCCESS 200


```
{
    "symbol": "ORCL",
    "tstamp": "2016-01-08T21:01:00Z",
    "price": 250,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/ords
                /rmougprov/tab-StockTicker/ORCL,2016
                -01-08T21%3A01%3A00Z"
        },
        {
            "rel": "edit",
            "href": "http://localhost:8080/ords
                /rmougprov/tab-StockTicker/ORCL,2016
                -01-08T21%3A01%3A00Z"
        },
        {
            "rel": "describedby",
            "href": "http://localhost:8080/ords
                /rmougprov/metadata-catalog/tab
                -StockTicker/item"
        },
        {
            "rel": "collection"
```

# DEMO 02a: REST requests 5 Create StockTicker Entry


RMOUG Demo Create StockTicker Entry

PUT http://192.168.56.101:8080/ords/rmougprov/tab-StockTicker/RMOUG,2016-01-08T21:01:55Z

BODY PARAMS AUTH HEADERS JSON

```
{
    "price": 50000
}
```

RESPONSE HEADERS 1564 ms SUCCESS 200

```
{
    "symbol": "RMOUG",
    "tstamp": "2016-01-08T21:01:55Z",
    "price": 50000,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/ords
                /rmougprov/tab-StockTicker/RMOUG,2016
                -01-08T21%3A01%3A55Z"
        },
        {
            "rel": "edit",
            "href": "http://localhost:8080/ords
                /rmougprov/tab-StockTicker/RMOUG,2016
                -01-08T21%3A01%3A55Z"
        },
        {
            "rel": "describedby",
            "href": "http://localhost:8080/ords
                /rmougprov/metadata-catalog/tab
                -StockTicker/item"
        },
        {
            "rel": "collection"
```

# DEMO 02a: REST requests 6 Delete StockTicker Entry


RMOUG Demo Delete StockTicker Entry

DELETE http://192.168.56.101:8080/ords/rmougprov/tab-StockTicker/TDC,2016-01-19T02%3A04%3A00Z

BODY PARAMS AUTH HEADERS JSON

```
{
    "symbol": "RMOUG",
    "tstamp": "2016-01-08T21:01:55Z"
}
```

RESPONSE HEADERS 17 B 289 ms SUCCESS 200

```
{
    "rowsDeleted": 0
}
```

# The JSON produced by ORDS is NOT schemaless

## Oracle has defined a new Media Type

- application/vnd.oracle.resource+json

## Whitepaper

- http://www.oracle.com/webfolder/technetwork/tutorials/appdevinfo/New%20REST%20Media%20Type.pdf

# DEMO 03: - http_rest_response


```
-- Create the types to support the table function.
drop type http_res-)
t_response_tab;
drop type http_rest_response_row;

create type http_rest_response_row
  as object
  ( id           NUMBER
  , url          VARCHAR2(4000)
  , response     clob
);
/
show errors

create type http_rest_response_tab
  is table of http_rest_response_row;
/
show errors


create or replace function http_rest_response (p_url  in  varchar2)
  return http_rest_response_tab pipelined
as
  l_http_request   UTL_HTTP.req;
  l_http_response  UTL_HTTP.resp;
  l_rownum         number := 0;
  l_url            varchar2(4000) := p_url;
  l_hasmore        varchar2(  10) := 'true';
  l_clob           clob;
  l_text           varchar2(32767);
begin
  while coalesce(l_hasmore, 'false') = 'true'
  loop
    l_rownum := l_rownum +1;
    -- Initialize the CLOB.
    DBMS_LOB.createtemporary(l_clob, false);
    -- Make a HTTP request and get the response.
    l_http_request  := UTL_HTTP.begin_request(l_url);
    l_http_response := UTL_HTTP.get_response(l_http_request);
    -- Copy the response into the CLOB.
    begin
      loop
        UTL_HTTP.read_text(l_http_response, l_text, 32766);
        DBMS_LOB.writeappend (l_clob, length(l_text), l_text);
      end loop;
    exception
      when UTL_HTTP.end_of_body
        then UTL_HTTP.end_response(l_http_response);
    end;
    -- Piping ROw
    pipe row(http_rest_response_row(l_rownum, l_url, l_clob));
    --
    select hasMore, href
      into l_hasMore
         , l_url
      from json_table ( l_clob, '$'
                        columns ( hasMore varchar2  path '$.hasMore'
                                , nested path '$.links[*]'
                                  columns ( rel varchar2 path '$.rel'
                                          , href varchar2 path '$.href' )
                                  )
                      ) j
     where hasmore='true'
       and rel='next';
    -- Relase the resources associated with the temporary LOB.
    DBMS_LOB.freetemporary(l_clob);
  end loop;
exception
  when others then
    UTL_HTTP.end_response(l_http_response);
    -- Relase the resources associated with the temporary LOB.
    DBMS_LOB.freetemporary(l_clob);
    raise;
end http_rest_response;
/
show errors
```

Script: „03 - rmougCons - http_rest_response.sql“

# DEMO 03a: create ACL


```
/*

ORA-29273: HTTP request failed
ORA-24247: network access denied by access control
list (ACL)

*/

-- Execute as dba
grant execute on utl_http to rmougCons;
grant execute on dbms_lock to rmougCons;

BEGIN
  DBMS_NETWORK_ACL_ADMIN.drop_acl (
    acl          => 'local_rest_acl_file.xml'   );
end;
/



BEGIN
  DBMS_NETWORK_ACL_ADMIN.create_acl (
    acl          => 'local_rest_acl_file.xml',
    description  => 'Grant Access to REST Services
on Host localhost',
    principal    => 'RMOUGCONS',
    is_grant     => TRUE,
    privilege    => 'connect',
    start_date   => SYSTIMESTAMP,
    end_date     => NULL);
end;
/

begin
  DBMS_NETWORK_ACL_ADMIN.assign_acl (
    acl          => 'local_rest_acl_file.xml',
    host         => 'localhost',
    lower_port   => 8080,
    upper_port   => NULL);
end;
/

commit;
```

Script: „03a - dba - create ACL.sql“

# DEMO 04: Building the View


```
ALTER SESSION SET TIME_ZONE = 'MST';


create or replace view StockTicker_ORDS
as
select symbol
     , to_number(price, '999999999999D99999999','nls_numeric_characters=''.,''' )  price
     , to_date(tstamp, 'YYYY-MM-DD"T"HH24:MI:SS"Z"') tstamp_json
     , cast( to_timestamp_tz(to_char(to_date(tstamp, 'YYYY-MM-DD"T"HH24:MI:SS"Z"'), 'YYYY-MM-DD HH24:MI:SS "UTC"'), 'YYYY-MM-DD HH24:MI:SS TZR')
             at time zone sessiontimezone as date ) tstamp
     , selfurl
  from table(http_rest_response('http://localhost:8080/ords/rmougprov/tab-StockTicker/')) t
     , json_table(t.response, '$.items[*]'
                  columns ( symbol  varchar2 path '$.symbol'
                          , tstamp  varchar2 path '$.tstamp'
                          , price   varchar2 path '$.price'
                          , selfurl varchar2 path '$.links[0].href'
                          )
                ) j
  ;

/* Test */
select symbol, tstamp, price
  from StockTicker_ORDS
minus
select symbol, tstamp, price
  from rmougprov.StockTicker;

select symbol, tstamp, price
  from rmougprov.StockTicker
minus
select symbol, tstamp, price
  from StockTicker_ORDS
;
```

Script: „04 - rmougCons - Building the View.sql“

# DEMO 05: View DML


```
set define off
create or replace trigger StockTicker_ORDS_IUD
instead of insert or update or delete on StockTicker_ORDS
for each row
declare
  c_ords_url constant varchar2(256) := 'http://localhost:8080/ords/rmougprov/tab-
StockTicker/';

  procedure http_rest_request ( p_url          varchar2
                              , p_req_body     varchar2
                              , p_req_method   varchar2
                              )
  as
    l_http_request   UTL_HTTP.req;
    l_http_response  UTL_HTTP.resp;
    l_text           varchar2(32767);
  begin
    lg('p_req_method: '||p_req_method||' URL: '||p_url||' p_req_body:
'||nvl(p_req_body, '**NULL**'));
    l_http_request := utl_http.begin_request (url=> p_url, method => p_req_method);
    utl_http.set_header ( r      =>  l_http_request
                        , name   =>  'Content-Type'
                        , value  =>  'application/json');
    utl_http.set_header ( r      =>   l_http_request
                        , name   =>   'Content-Length'
                        , value  =>   length(p_req_body));
    utl_http.write_text ( r      =>   l_http_request
                        , data   =>   p_req_body);
    l_http_response := utl_http.get_response(l_http_request);
    loop
      utl_http.read_text(l_http_response, l_text, 32766);
    end loop;
    lg('1: resp'||l_text);
    utl_http.end_response(l_http_response);
  exception
    when utl_http.end_of_body
      then utl_http.end_response(l_http_response);
           lg('2: resp'||l_text);
  end http_rest_request;
```

```
begin
  if deleting then
     lg('Deleting...');
     http_rest_request(:old.selfurl, '{}', 'DELETE');
    null;
  end if;
  if updating
    then
    null;
  end if;
  if inserting then
    null;
  end if;
end;
/
show errors
```

Script: „05 - rmougCons - View DML.sql“

# DEMO 06: Generator in Action


Script: „06 - generator.sql“

# DEMO 07: Performance


```
/* cleanup
drop database link rmoug_prov;
drop table st_perf_dbl        purge;
drop table st_perf_rest_25    purge;
drop table st_perf_rest_100   purge;
drop table st_perf_rest_500   purge;
drop table st_perf_rest_1000  purge;
drop table st_perf_rest_5000  purge;

*/

create database link rmoug_prov
  connect to rmougprov identified by rmoug
  using '//192.168.56.101/orcl';


set timing on
set feedback on
create table st_perf_dbl
  as select * from stockticker@rmoug_prov;

create table st_perf_rest_1000
as
select symbol
     , to_number(price, '999999999999D99999999','nls_numeric_characters=''.,''' )  price
     , to_date(tstamp, 'YYYY-MM-DD"T"HH24:MI:SS"Z"') tstamp_json
     , cast( to_timestamp_tz(to_char(to_date(tstamp, 'YYYY-MM-DD"T"HH24:MI:SS"Z"'), 'YYYY-MM-DD HH24:MI:SS "UTC"'), 'YYYY-MM-DD HH24:MI:SS TZR')
             at time zone sessiontimezone as date ) tstamp
  from table(http_rest_response('http://192.168.56.101:8080/ords/rmougprov/tab-StockTicker/?limit=1000')) t
     , json_table(t.response, '$.items[*]'
                  columns ( symbol  varchar2 path '$.symbol'
                          , tstamp  varchar2 path '$.tstamp'
                          , price   varchar2 path '$.price'
                          , selfurl varchar2 path '$.links[0].href'
                          )
                ) j
/
```

Script: „07 - rmougCons - Performance.sql“

# Generator

## Use from sqlcl.

- JavaScript based

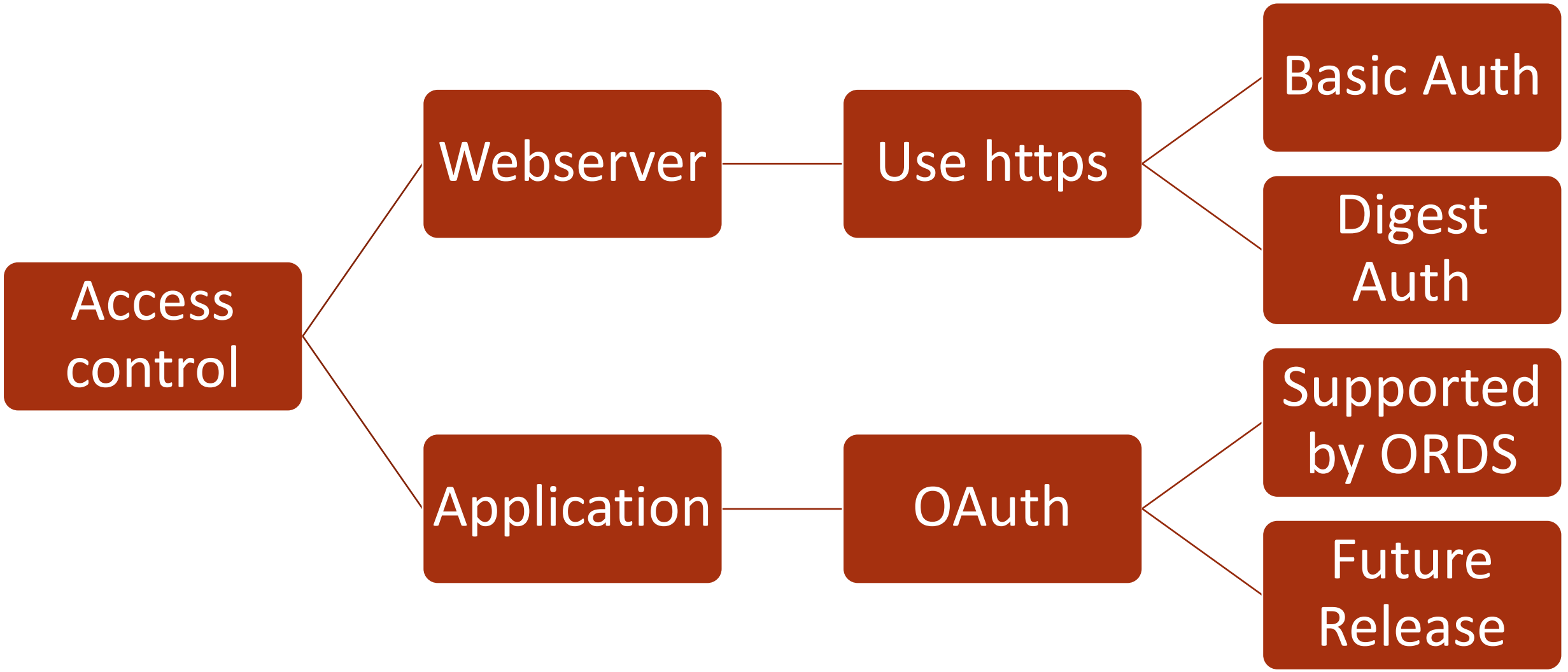
## Parameter:

- View Name
- ORDS Metadata URL
- opional: Parameter for URL

```
script ../generator/generator
          v_stock
          http://192.168.56.101:8080/ords/rmougprov/metadata-catalog/tab-
StockTicker/
          ?limit=5
-- call from single line
```

# Performance Analysis

# Security


Access control
Webserver
Use https
Basic Auth
Digest Auth
Application
OAuth
Supported by ORDS
Future Release

# Limitations / Future enhancements

## Limitations

simple data types only

- No Spatial
- No Object Types

## Planned enhancements

- OAuth
- Complex data types
- Move from AutoREST to custom implementation
- 11g Version with PL/JSON and apex_web_service

# How do you get it?

| | |
|---|---|
| conference website | Slides and scripts uploaded to |
| | http://www.rmoug.org/training/training-days-2016/ |
| GitHub | Latest version always here |
| | You can help enhance the generator – I'm accepting pull requests |
| | https://github.com/its-people/rest-db-links.git |
| notification | Twitter: @RobbieDatabee  |
| | Blog: http://www.its-people.de/blog  |

# Conclusion

Only a little magic is needed.

Performance is worse, but acceptable in most cases.

With REST and JSON, you can reach out from your database to the Internet.

**Database Links can be replaced by a modern Architecture.**

we make the difference
www.its-people.de

# Herzlichen Dank für Ihre Aufmerksamkeit !

## Questions ?





**its-people GmbH**
Frankfurt Tel. 069 2475 2100
Hamburg Tel. 040 2360 8808
Köln Tel. 0221 1602 5204
München Tel. 089 5484 2401

**its-people ERP Beratungsgesellschaft mbH**
Frankfurt Tel. 069 2475 1980

**www.its-people.de** **info@its-people.de**