LEAD WITH CURIOSITY

REST Connector in the world of IoT

Christof Schwarz
Principal Solution Architect

15-May-2019

Qonnections
QLIK GLOBAL CONFERENCE

Agenda

- Introduction to REST
- Prepare your work with an API
- Working with Qlik REST Connector
  - Demystifying the Rest Connector
  - Workflow for connecting to a new REST API
  - Examples: Google Spreadsheet API
- Some Qlik Script tricks
  - Transposing Data
  - ISO-Date handling
  - Paging Techniques

LIVE!

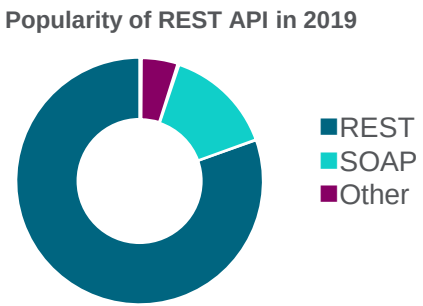
Introduction to REST

Qonnections
QLIK GLOBAL CONFERENCE

Before we start ...

REST vs SOAP

- REST
  - Representational State Transfer
  - started to spread in 2005
  - is a design style
  - Uses an URI to access information
  - Many different data formats
- SOAP
  - Simple Object Access Protocol
  - Is a protocol (“envelope”)
  - A service interface (calling functions)
  - More overhead
  - XML data format



Before we start ...

JSON vs XML

XML
Extensible Markup Language

```
<?xml version="1.0" encoding="UTF-8"?>
<authentication-context>
   <username>my_username</username>
   <password>my_password</password>
   <id>32443</id>
   <validationfactors>
      <validationfactor>
         <name>remote_address</name>
         <value>127.0.0.1</value>
      </validationfactor>
   </validationfactors>
</authentication-context>
```

JSON
JavaScript Object Notation

```
{
  "username" : "my_username",
  "password" : "my_password",
  "id" : 32443,
  "validationfactors" : {
     "validationfactor" : [
        { "name" : "remote_address",
          "value" : "127.0.0.1" }
     ]
   }
}
```

Strict notation!

Before we start ...

JSON vs XML

Language

```
oding="UTF-8"?>
>
ame</username>
ord</password>


ddress</name>
</value>


t>
```

JSON

JavaScript Object Notation

```
{
  "username" : "my_username",
  "password" : "my_password",
  "id" : 32443,
  "validationfactors" : {
    "validationfactor" : [
      { "name" : "remote_address",
        "value" : "127.0.0.1" }
    ]
  }
}
```

Strict notation!

```
{
  username : "my_username",
  password : "my_password",
  id : 32443,
  validationfactors : {
    validationfactor : [
      { name : 'remote_address',
        value : '127.0.0.1' }
    ]
  }
}
```

No relaxed notation ...

Prepare your work with an API

Read API Documentation
Play with the API using a tool

Qonnections
QLIK GLOBAL CONFERENCE

Ask for the documentation

Documention

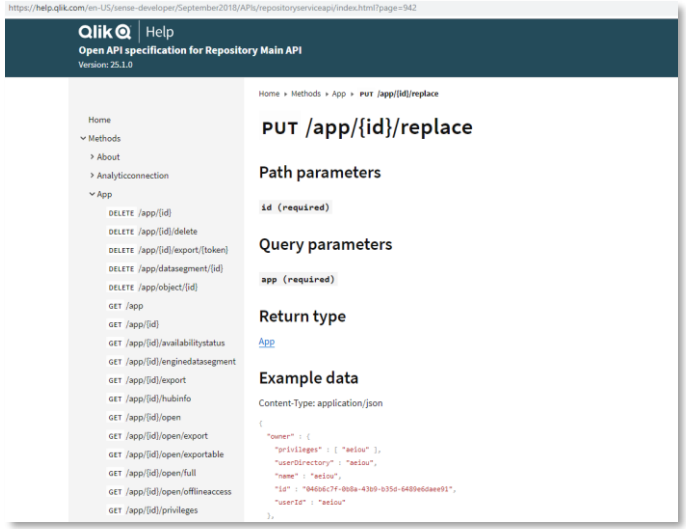
- Describe what methods (endpoints) are doing and which parameters they need
- Often created with Swagger

Sending Request

- Method (GET, POST, PUT ...)
- Path parameters
- QueryString parameters
- http-headers
- Body

Receiving Answer

- Response code
- Body

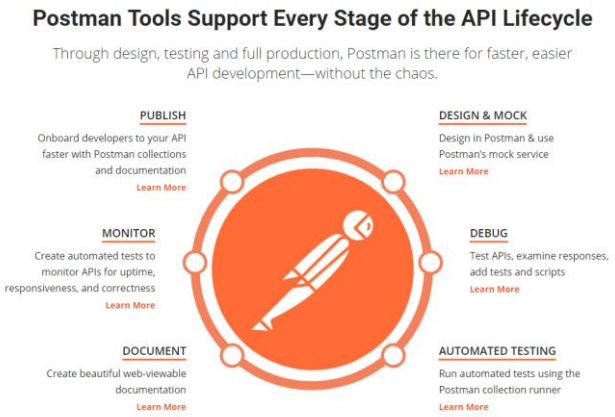


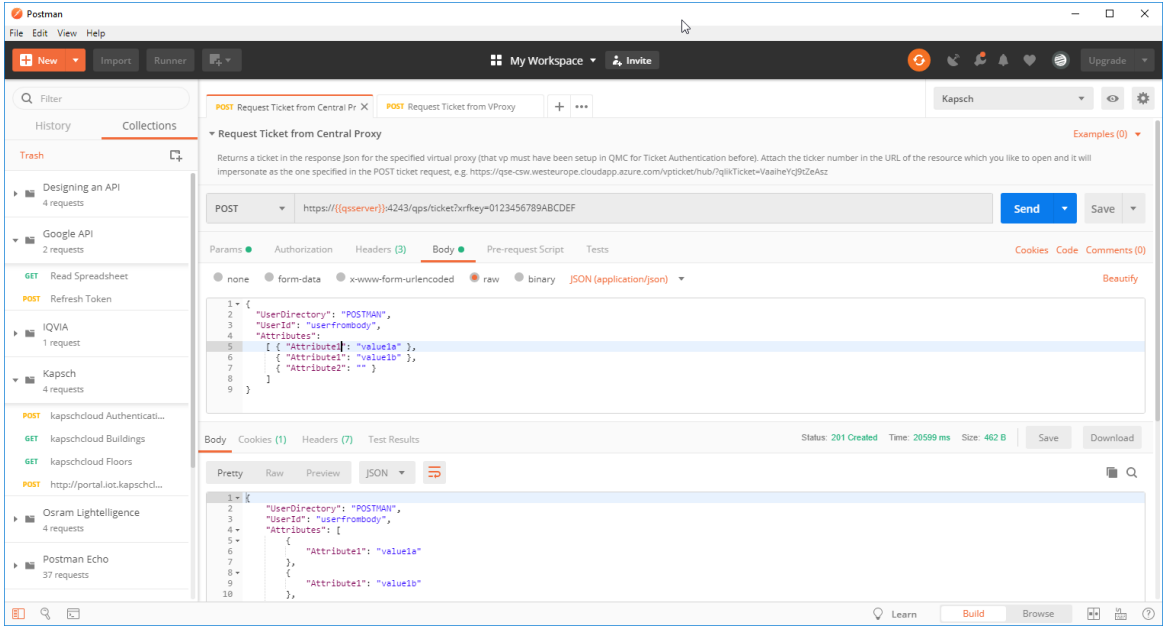
Example https://help.qlik.com/en-US/sense-developer/April2019/APIs/repositoryserviceapi

Play with the API using a tool

For example: Postman

www.getpostman.com





Play with API

Save in collections with variables

Echo Server

Mock Server

Copy Code

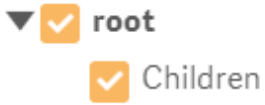
Working with Qlik REST Connector

Qlik REST Connector under the hood

De-mystify the generated script

Json Response

```
[
    {
        "Wife": "Martina",
        "Husband": "Christof",
        "Children": [
            {"name": "Julia"},
            {"name": "John"}
        ]
    },{
        "Wife": "Mary",
        "Husband": "Alexander"
    }
]
```



REST Connector Wizard

```
RestConnectorMasterTable:
SQL SELECT
    "Wife",
    "Husband",
    "__KEY_root",
    (SELECT
        "name",
        "__FK_Children"
    FROM "Children" FK "__FK_Children")
FROM JSON (wrap on) "root" PK "__KEY_root";
```

Field names are tolerant. The LOAD doesn‘t break if you attempt to load a non-existing key.

Qlik REST Connector under the hood

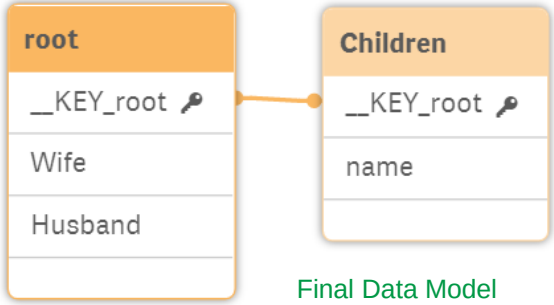
name	__FK_Children	Wife	Husband	__KEY_root	__extra_
Julia	1	-	-	-	-
John	1	-	-	-	-
-	-	Martina	Christof	1	-
-	-	Mary	Alexander	2	-

RestConnectorMasterTable (temporary)

```
[Children]:
LOAD    [name],
    [__FK_Children] AS [__KEY_root]
RESIDENT RestConnectorMasterTable
WHERE NOT IsNull([__FK_Children]);


[root]:
LOAD    [Wife],
    [Husband],
    [__KEY_root]
RESIDENT RestConnectorMasterTable
WHERE NOT IsNull([__KEY_root]);


DROP TABLE RestConnectorMasterTable;
```



Final Data Model

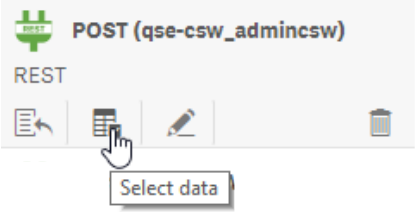
Embrace Qlik Scripting

It takes a program logic to interact with REST APIs

- In most cases, it is not a static, single call of a REST URI
  - The „Select Data“ wizard alone won‘t do the job
  - A series of calls are needed, which build on each other
  - Embrace the capabilities of Qlik Scripting

- For example
  - First of all, get a token for the next calls
  - Complex paging scenarios
  - Transposing results

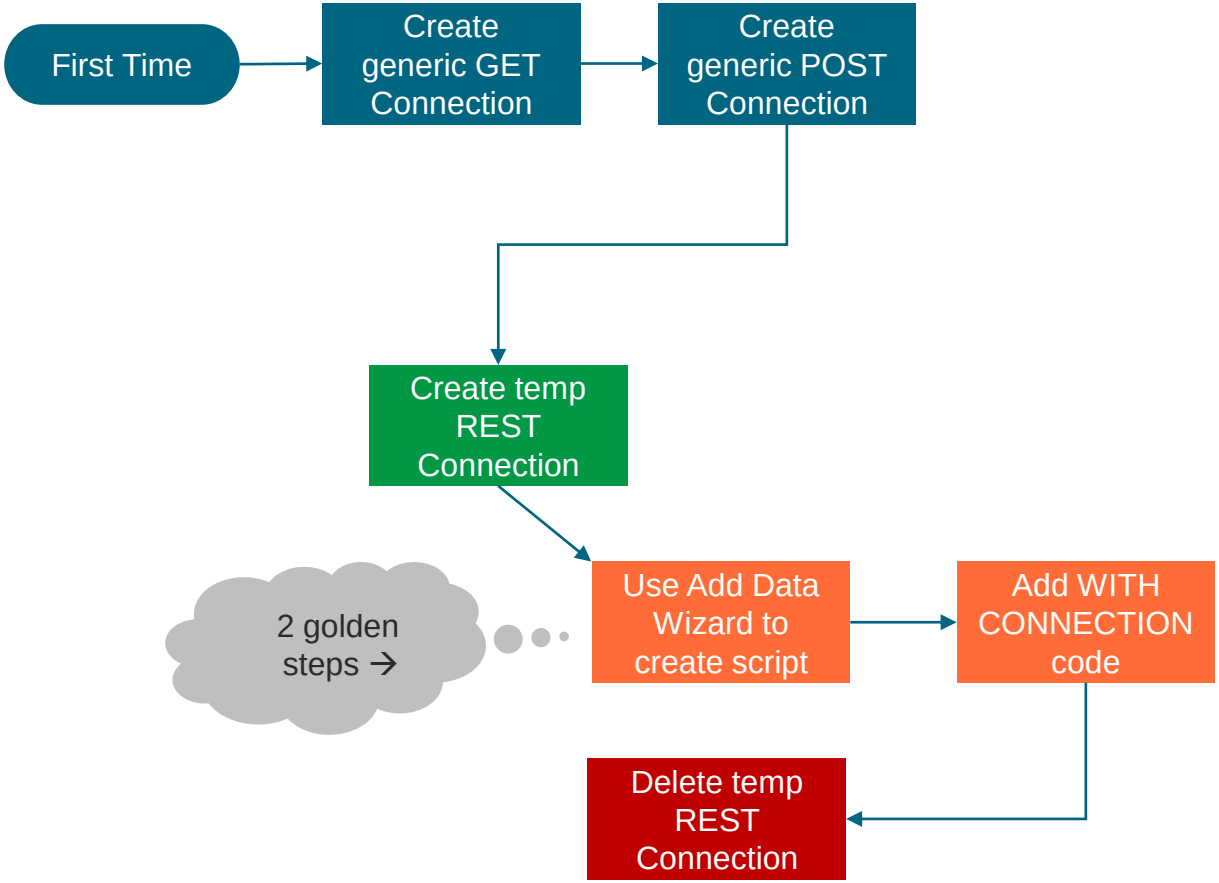
Select Data wizard



+ Scripting

POST → GET → GET → GET → ...

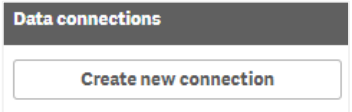
Workflow for working with REST Connector



Set up 2 placeholder REST-connections

- A **GET** request (https://postman-echo.com/get)
- A **POST** request (https://postman-echo.com/post)
- Leave the settings empty for Query-strings, Http-Heades, Body
  - you will <u>later</u> parameterize all those with script
- Why two requests?
  - You cannot parameterize the http-method itself (GET/POST) with script
  - The http-method will come from this script line just before the SELECT ...

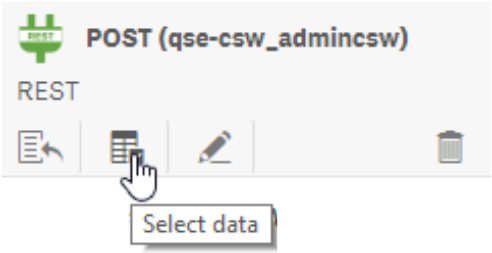
```
LIB CONNECT TO 'get_connection';
LIB CONNECT TO 'post_connection';
```



Note:

The „Create New Connection“ dialog can only be safed when there was a proper REST response


```
SQL SELECT
    "__KEY_root",
    "rootfield",
    (SELECT
        "id",
        "name",
        "__FK_data"
    FROM "data" FK "__FK_data")
FROM JSON (wrap on) "root" PK "__KEY_root"
```



Use Add Data Wizard to create script

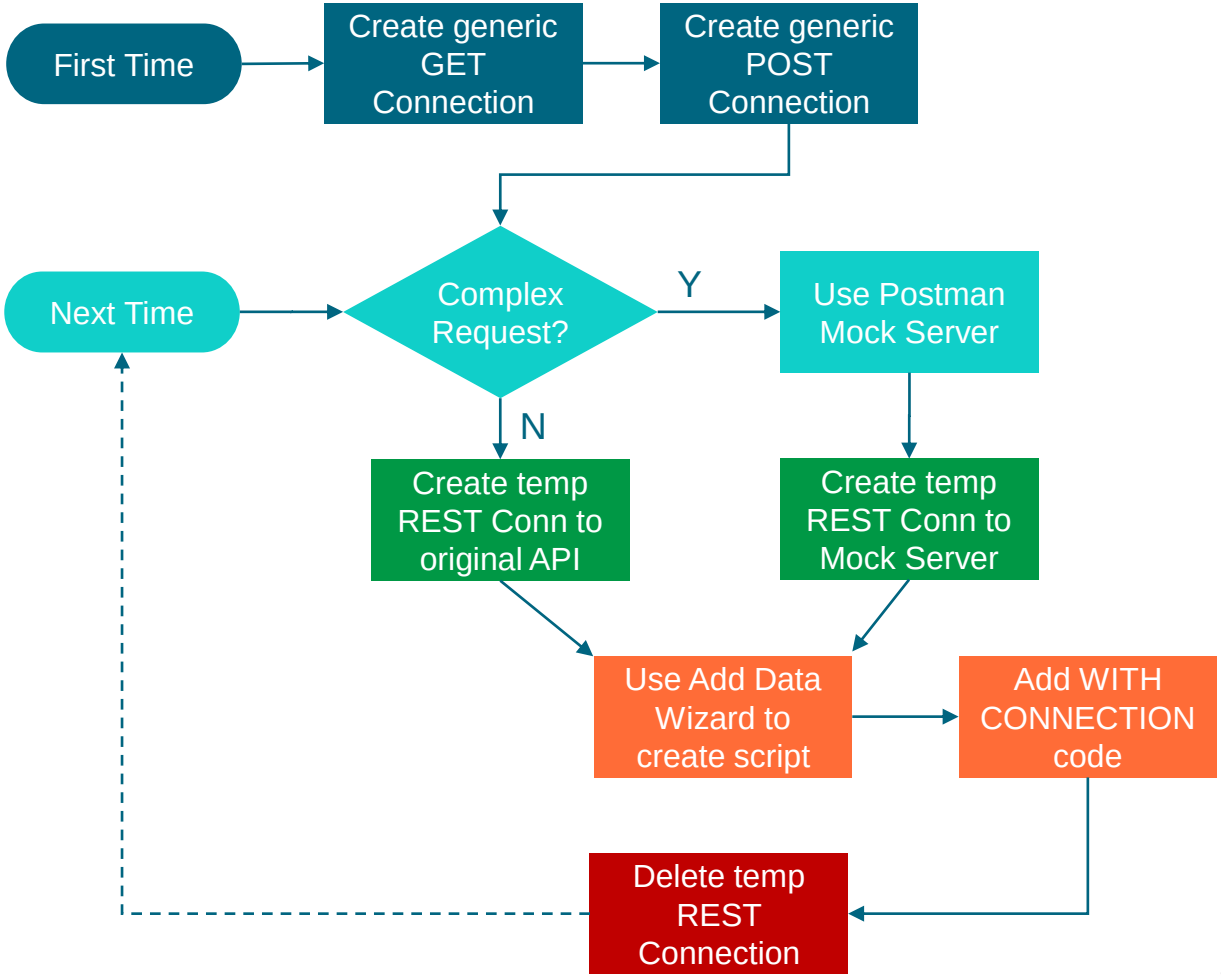
Add WITH CONNECTION code

```
WITH CONNECTION (
    URL "$(vBaseAPIurl)/users/$(vAPIuserId)"
    ,QUERY "tenent" "qliktrainees"
    ,HTTPHEADER "Content-Type" "application/json"
    ,HTTPHEADER "Authorization" "Bearer $(vToken)"
    //,HTTPHEADER "X-HTTP-Method-Override" "PUT",
    ,HTTPHEADER "cookie" "$(vCookie)"
    ,BODY "{""path"":""$(vAttribute)""}"
);
```

Insert all API parameters

2 golden steps

Workflow for working with REST Connector



Some Qlik Script Tricks

Request and use bearer token,
ISO-Date handling, Transposing Data, Paging

Qonnections
QLIK GLOBAL CONFERENCE

Receiving an access token

And use it as bearer authentication in subsequent calls

1) Get the token

```
LIB CONNECT TO 'REST POST Request';
Authentication:
SQL SELECT
    "access_token",
    "token_type",
    "expires_in",
    "expiration_time",
    "error_message"
FROM JSON (wrap on) "root"
WITH CONNECTION (
    URL "$(vBaseAPIurl)/api/Authentication"
    ,QUERY "username" "#####"
    ,QUERY "password" "#####"
);

LET vToken = FieldValue('access_token', 1);
DROP TABLE Authentication;
TRACE New Token is $(vToken);
```

1

2) Use the token

```
LIB CONNECT TO 'REST GET Request';

RestConnectorMasterTable:
SQL SELECT
    ...
FROM JSON (wrap on) "root"
WITH CONNECTION (
    URL "$(vBaseAPIurl)/api/endpoint"
    ,HTTPHEADER "Authorization" "Bearer $(vToken)"
    ...
);
```

2

Even better: Try the old token first, get a new only if the old doesn‘t work anymore
→ https://github.com/ChristofSchwarz/qs_script_rest_api/blob/master/sub_try_request.md

Transposing Arrays

If data is returned in an array matrix like this instead of key-value pairs ...


```
{
    "range": "'Form responses 1'!A1:S125",
    "majorDimension": "ROWS",
    "values": [
        [
            "Timestamp",
            "Passenger",
            "From Airport",
            "To Airport",
            "Date",
            "Operator",
            "Aircraft Type",
            "Flight-Number",
            "Reason for Travel",
            "Company/Private",
            "Travel Class",
            "",
            "Comment"
        ],
        [
            "18/12/2018 16:24:57",
            "Christof",
            "VIE",
            "STR",
            "18/12/2018 16:30:00",
            "Eurowings",
            "A319",
            "",
            "PDS Software & PDS",
            "Company",
            "Economy"
        ],
        [
            "18/12/2018 18:27:02",
            "Christof",
            "Str"
```

Transposing Arrays

... the imported result will have two fields, an „array row“ autoid and a value ...

But what you want is ...

__KEY_values		@Value
1	1	Passenger
1	1	From Airport
1	1	To Airport
1	1	Date
1	1	Operator
2	2	Christof
2	2	VIE
2	2	STR
2	2	18/12/2018 16:30:00
2	2	Eurowings
3	3	Christof
3	3	Str
3	3	Bre
3	3	18/12/2018 18:40:00
3	3	Eurowings
	4	Vie
	4	Christof
	4	Ham

Transposing Arrays

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3

But what you want is ...

this where each element of an array row is are transposed into a column (field) in Qlik.

__KEY_values	@Value
1	Passenger
1	From Airport
1	To Airport
1	Date
1	Operator
2	Christof
2	VIE
2	STR
2	18/12/2018 16:30:00
2	Eurowings
3	Christof
3	Str
3	Bre
3	18/12/2018 18:40:00
3	Eurowings
4	Vie
4	Christof
4	Ham

Transposing Arrays

1	1
1	2
1	3
1	4
1	5
2	1
2	2
2	3
2	4
3	1
3	2
3	3
3	4
3	5

- Introduce a row autoid __X which
  - restarts at 1 and
  - increments when the main key is the same as above

```
RestConnectorMasterTable:
LOAD *,
    If(Len(__FK_values_u0)
        ,If(Peek('__FK_values_u0')=__FK_values_u0, Peek('__X')+1, 1)
        ) AS __X,
;
SQL SELECT
    "__KEY_root",
    (SELECT
        "__FK_values",
        "__KEY_values",
        (SELECT
            "@Value",
            "__FK_values_u0"
        FROM "values" FK "__FK_values_u0" ArrayValueAlias "@Value")
    FROM "values" PK "__KEY_values" FK "__FK_values")
FROM JSON (wrap on) "root" PK "__KEY_root"
WITH CONNECTION (  ...
```

Transposing Arrays

1	2	3	4	5
1	1	1	1	1
2	2	2	2	
3	3	3	3	3

- Use Generic Load to achieve this transpose

```
GENERIC LOAD
    __FK_values_u0, __X, @Value
RESIDENT RestConnectorMasterTable
WHERE __FK_values_u0 > 0;

DROP TABLE RestConnectorMasterTable;
```

Transposing Arrays

1	2	3	4	5
1	1	1	1	1
2	2	2	2	
3	3	3	3	3

- Use Generic Load to achieve this transpose

If the field names are not part of the response ...

```
__FieldNames:
MAPPING LOAD * INLINE [
    1, Timestamp
    2, Passenger
    3, From Airport
    4, To Airport
    5, Date
    6, Operator
    7, Aircraft Type
] (no labels);

GENERIC LOAD
       __FK_values_u0, ApplyMap('__FieldNames', __X), @Value
RESIDENT RestConnectorMasterTable
WHERE __FK_values_u0 > 0;

DROP TABLE RestConnectorMasterTable;
```

Script Snippets → https://github.com/ChristofSchwarz/qs_script_rest_api/blob/master/transposing.md

Transposing Arrays

1	2	3	4	5
1	1	1	1	1
2	2	2	2	
3	3	3	3	3

- Use Generic Load to achieve this transpose

If the field names are in block 1 of the response

```
__FieldNames:
MAPPING LOAD __X, @Value
RESIDENT RestConnectorMasterTable
WHERE __FK_values_u0 = 1 AND Len(@Value);
```

Script Snippets → https://github.com/ChristofSchwarz/qs_script_rest_api/blob/master/transposing.md

```
GENERIC LOAD
       __FK_values_u0, ApplyMap('__FieldNames', __X), @Value
RESIDENT RestConnectorMasterTable
WHERE __FK_values_u0 > 1;

DROP TABLE RestConnectorMasterTable;
```

Date handling

Reading ISO dates

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
`Field` →	2	0	1	8	-	1	1	-	0	1	T	1	4	:	3	2	:	5	1	.	1	2	3	Z
`Left(Field,10)`	2	0	1	8	-	1	1	-	0	1														
`Mid(Field,12,8)`												1	4	:	3	2	:	5	1					
`Mid(Field,12,12)`												1	4	:	3	2	:	5	1	.	1	2	3	

Date handling

Reading ISO dates

```
                                 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
        Field →                  2 0 1 8 - 1 1 - 0 1 T 1 4 : 3 2 : 5 1 . 1 2 3 Z
Date#( Left(Field,10) ,'         Y Y Y Y - M M - D D ' )
Time#( Mid(Field,12,8)                                , ' h h : m m : s s ' )
Time#( Mid(Field,12,12)                               , ' h h : m m : s s . f f f ' )
```

All together:

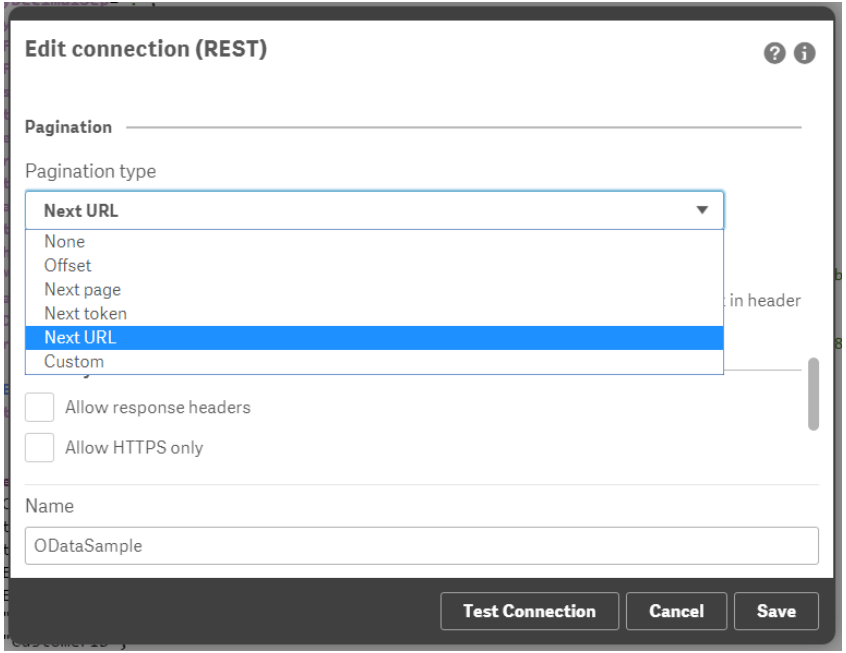
```
Timestamp(
    Date#(Left(Field,10) ,'YYYY-MM-DD’)) + Time#(Mid(Field,12,8),'hh:mm:ss’)
,'$(TimestampFormat)') AS Field
```

Code Snippet → https://github.com/ChristofSchwarz/qs_script_rest_api/blob/master/date_field_processing.md

Paging with REST APIs

Built-in paging types

- There are some paging strategies supported with no coding, e.g.
  - BestBuy
  - Facebook
  - Google Analytics
- REST Connector and Pagination Video (M. Tarallo) https://youtu.be/QICT55_712I



→ https://help.qlik.com/en-US/connectors/Subsystems/REST_connector_help/Content/Connectors_REST/Create-REST-connection/Pagination-scenarios.htm

Paging

Example

- Works out of the box.

GET https://schenck-sims.adamos-dev.com/measurement/measurements

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

Query Params

KEY	VALUE
Key	Value

Body | Cookies | Headers (6) | Test Results

Pretty | Raw | Preview | JSON


```
1  {
   "next": "https://schenck-sims.adamos-dev.com/measurement/measurements?pageSize=5&currentPage=2",
   "self": "https://schenck-sims.adamos-dev.com/measurement/measurements?pageSize=5&currentPage=1",
4  "statistics": {
5      "totalPages": 1433630,
6      "currentPage": 1,
7      "pageSize": 5
8  },
9  "measurements": [
10     {
11         "self": "https://schenck-sims.adamos-dev.com/measurement/measurements/8134601",
12         "time": "2019-03-14T00:00:00.819Z",
13         "id": "8134601",
14         "source": {
15             "self": "https://schenck-sims.adamos-dev.com/inventory/managedObjects/107",
16             "id": "107"
17         },
18         "type": "csd_sim_Universal1",
19         "Status": {
20             "csd_Status": {
21                 "value": 2
22             }
```


Qonnections QLIK GLOBAL CONFERENCE

Paging

Example

- Does not work out of the box
- Dropped querystring “$format”
- Relative url

GET https://services.odata.org/V3/Northwind/Northwind.svc/Orders?$expand=Order_Details&$format=json

Pretty Raw Preview JSON


```
7288                "Quantity": 40,
7289                "Discount": 0
7290            },
7291            {
7292                "OrderID": 10447,
7293                "ProductID": 65,
7294                "UnitPrice": "16.8000",
7295                "Quantity": 35,
7296                "Discount": 0
7297            },
7298            {
7299                "OrderID": 10447,
7300                "ProductID": 71,
7301                "UnitPrice": "17.2000",
7302                "Quantity": 2,
7303                "Discount": 0
7304            }
7305        ],
7306        "OrderID": 10447,
7307        "CustomerID": "RICAR",
7308        "EmployeeID": 4,
7309        "OrderDate": "1997-02-14T00:00:00",
7310        "RequiredDate": "1997-03-14T00:00:00",
7311        "ShippedDate": "1997-03-07T00:00:00",
7312        "ShipVia": 2,
7313        "Freight": "68.6600",
7314        "ShipName": "Ricardo Adocicados",
7315        "ShipAddress": "Av. Copacabana, 267",
7316        "ShipCity": "Rio de Janeiro",
7317        "ShipRegion": "RJ",
            "ShipPostalCode": "02389-890",
            "ShipCountry": "Brazil"
7320    }
7321 ],
7322 "odata.nextLink": "../../Northwind/Northwind.svc/Orders?$expand=Order_Details&$skiptoken=10447"
7323 }
```


Qonnections
QLIK GLOBAL CONFERENCE

Tricks for Paging

OData

Key generation strategy

Current record

1. Set Key Generation strategy to „Current Record“
2. Get the first data page with the REST Connector Wizard
   - (If missing add "odata.nextLink“)
   - Add WITH CONNECTION (...) as always
3. Put all the blocks into a DO-WHILE loop
   - Create variable skiptoken before the loop
   - Add $skiptoken to „WITH CONNECTION“ block
4. Parse the „$skiptoken“ argument from odata.nextLink field

Code Snippets:
https://github.com/ChristofSchwarz/qs_script_rest_api/blob/master/odata.md

Sampe Data:
https://services.odata.org/V3/Northwind/Northwind.svc/Orders?$expand=Order_Details&$format=json

```
LET skiptoken = '';
DO

  RestConnectorMasterTable:
  SQL SELECT
      "odata.metadata",
      "odata.nextLink",
      "__KEY_root",
      ...
      FROM "value" PK "__KEY_value" FK "__FK_value")
  FROM JSON (wrap on) "root" PK "__KEY_root"
  WITH CONNECTION ( QUERY "$skiptoken" "$(skiptoken)" );

  ... Other tables like „values“ and „root“

  nextLink: LOAD Only(odata.nextLink) RESIDENT 'RestConnectorMasterTable';
  LET nextlink = FieldValue('Only(odata.nextLink)', 1);
  DROP TABLE nextLink;
  LET skiptoken = TextBetween(nextlink & '&', '$skiptoken=', '&');
  WHEN Len(nextlink) TRACE [nextLink $skiptoken=$(skiptoken)];

  DROP TABLE RestConnectorMasterTable;

LOOP WHILE Len(nextlink)
```

IoT topics: All together

Advanced Scripting Skills

- How much data do have to load at once? → ODAG? QABDI?
- Can previous data be stored to get delta? → QVD
- Json structures of multiple API calls may differ but should be treated like one → CONCATENATE LOAD
- Data is cut off due to max page size → DO WHILE loops
- Series of sensor values are insufficient to answer business questions → post processing and enrich with flags, duration → RESIDENT LOAD
- Put the heavy-lifting into script

More resources

Help

- https://help.qlik.com/en-US/connectors/Subsystems/REST_connector_help/Content/Connectors_REST/Create-REST-connection/Create-REST-connection.htm

Code Snippets

- https://github.com/ChristofSchwarz/qs_script_rest_api
- https://github.com/ChristofSchwarz/qs_script_rest_api/blob/master/transposing.md

Videos

- REST Connector Deluxe (C. Schwarz) https://youtu.be/7m9ZejlzkkY
- Qlik and REST (M. Tarallo) https://youtu.be/ibCACdF_tPo
- REST Connector and Pagination (M. Tarallo) https://youtu.be/QlCT55_712I
- Google Sheet API with Qlik Script (C. Schwarz) https://youtu.be/l9sk-v_PTf8

Latest version of this presentation

Qonnections

QLIK GLOBAL CONFERENCE

Thank You