



CUSTOMER BETA PROGRAM
Denodo OData Service

30 July 2015

CONFIDENTIAL

THE BETA PROGRAM

Project Status

- Unpublished (first BETA)
- Developed by **Denodo Labs**
- To be distributed through a specific channel during BETA
- Future **DenodoConnect Enterprise** component
 - Distributed from the Support Site

BETA Program: Timeline

- GA intended publishing date: **27 August 2015**
- Feedback channel: email
 - Not the Support Site (unpublished software!)
 - **Daniel Fernández dfernandez@denodo.com**

THE DENODO ODATA SERVICE

Artifact Overview

- A **Java Web Application** (.war)
 - Deployed on e.g. an Apache Tomcat server
- Additional **northbound** interface for Denodo Platform
- Multi-database – serves an entire Denodo installation
- Southbound: connects to VDP via JDBC

OData support

- Supports **OData 2.0 (partially)**

- <http://www.odata.org/documentation/odata-version-2-0/overview/>
- <http://www.odata.org/documentation/odata-version-2-0/uri-conventions/>

- **OData 4.0** is an OASIS standard since 2014

- Why 2.0?

- OData 4 clients can talk to OData 2 servers
- Salesforce LightningConnect
- Technical constraint: unstable dependencies

ODataServ vs RESTfulWS

- Similar in concept to the RESTfulWS

`http://localhost:9090/denodo-restfulws`

- Authentication, URLs...
- But no HTML interface!

denodo [Logout](#)

View name
country

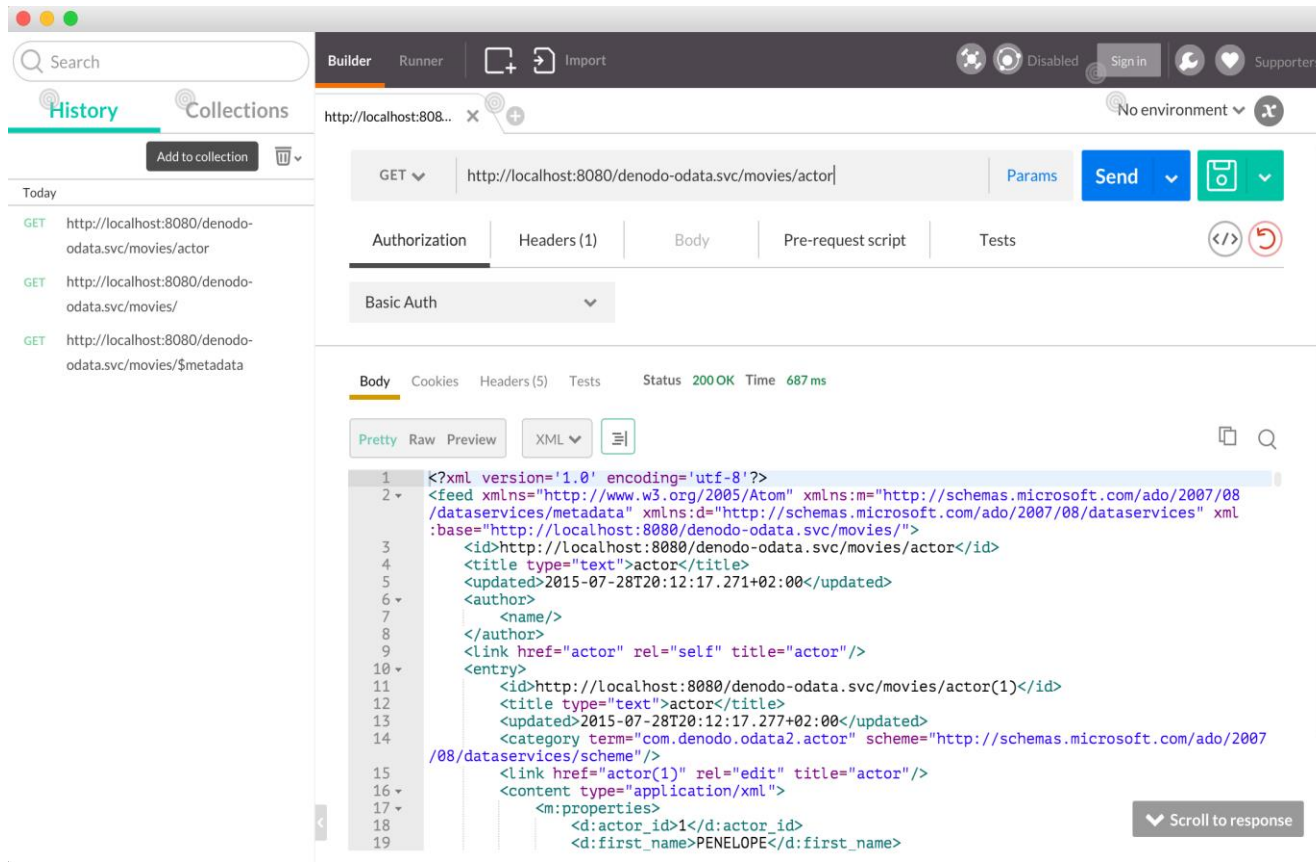
Related views
[city](#)

[Search](#)

| country_id | country | last_update | Self | cities |
|------------|----------------|---------------------|----------------------|------------------------|
| 1 | Afghanistan | 2006-02-15 03:44:00 | Self | cities |
| 2 | Algeria | 2006-02-15 03:44:00 | Self | cities |
| 3 | American Samoa | 2006-02-15 03:44:00 | Self | cities |
| 4 | Angola | 2006-02-15 03:44:00 | Self | cities |
| 5 | Anguilla | 2006-02-15 03:44:00 | Self | cities |
| 6 | Argentina | 2006-02-15 03:44:00 | Self | cities |
| 7 | Armenia | 2006-02-15 03:44:00 | Self | cities |
| 8 | Australia | 2006-02-15 03:44:00 | Self | cities |
| 9 | Austria | 2006-02-15 03:44:00 | Self | cities |
| 10 | Azerbaijan | 2006-02-15 03:44:00 | Self | cities |

Recommended Tools

■ API Tools: Postman, HTTPRequester...



Authentication

■ RESTfulWS-like URLs

```
http[s]://{server}:{port}/denodo-odata.svc/{database}
```

```
https://localhost:8080/denodo-odata.svc/movies
```

■ RESTfulWS-like authentication

GET ▾

http://localhost:8080/denodo-odata.svc/movies/

Authorization

Headers (1)

Body

Pre-request script

Tests

Basic Auth ▾

Username

:

odata

Password

:

•••••

The authorization header will be generated and added as a custom header.

☒ Show Password

☒ Save helper data to request

Clear

Update request

Metadata: Service Document /

- OData Service root for each database
- Lists the top-level feeds – i.e. the available views

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <service xml:base="http://localhost:8080/denodo-odata.svc/movies/" xmlns="http://www.w3.org/2007/app" xmlns:atom="http://www.w3
  .org/2005/Atom">
3   <workspace>
4     <atom:title>Default</atom:title>
5     <collection href="actor">
6       <atom:title>actor</atom:title>
7     </collection>
8     <collection href="address">
9       <atom:title>address</atom:title>
10    </collection>
11    <collection href="category">
12      <atom:title>category</atom:title>
13    </collection>
14    <collection href="city">
15      <atom:title>city</atom:title>
16    </collection>
17    <collection href="country">
18      <atom:title>country</atom:title>
19    </collection>
20    <collection href="customer">
21      <atom:title>customer</atom:title>
22    </collection>
23    <collection href="film">
24      <atom:title>film</atom:title>
25    </collection>
26    <collection href="film_category">
27      <atom:title>film category</atom:title>
```

Metadata: Entity Data Model / \$metadata

- Describes the entire database using CSDL (XML)

```
<EntityType Name="city">
  <Key>
    <PropertyRef Name="city_id"/>
  </Key>
  <Property Name="city_id" Type="Edm.Int32" Nullable="false"/>
  <Property Name="city" Type="Edm.String" Nullable="true" MaxLength="2000000000"/>
  <Property Name="country_id" Type="Edm.Int32" Nullable="true"/>
  <Property Name="last_update" Type="Edm.String" Nullable="true" MaxLength="65536"/>
</EntityType>
```


- 5 kinds of artifact:
 - Entity Types
 - Associations
 - Entity Sets
 - Association Sets
 - Imported Functions

Mapping OData with VDP

| Denodo ODATA Service | VDP |
|---|-------------------------|
| Entity Type | View Definition |
| <i>Entity Type > Property</i> | <i>View Column</i> |
| <i>Entity Type > Navigation Property</i> | <i>Association Role</i> |
| Association | Association Definition |
| Entity Set | View Data |
| Association Set | (Associated data) |
| Imported Functions | - |

One view, one collection (entity set)

- Each view in the VDP database → collection

| city | | |
|---|-------------|------|
|  | city_id | int |
| | city | text |
| | country_id | int |
| | last_update | text |

<http://localhost:8080/denodo-odata.svc/movies/city>

```
<entry>
  <id>http://localhost:8080/denodo-odata.svc/movies/city(1)</id>
  <title type="text">city</title>
  <updated>2015-07-28T20:28:11.882+02:00</updated>
  <category term="com.denodo.odata2.city" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <link href="city(1)" rel="edit" title="city"/>
  <content type="application/xml">
    <m:properties>
      <d:city_id>1</d:city_id>
      <d:city>A Corua (La Corua)</d:city>
      <d:country_id>87</d:country_id>
      <d:last_update>2006-02-15 03:45:25</d:last_update>
    </m:properties>
  </content>
</entry>
```

ATOMPub vs JSON: `$format`

- Data can be returned in AtomPub (XML) or JSON
- Except the EDM (`/$metadata`) – CSDL is XML

- Request parameter:

`http://server/denodo-odata.svc/movies/city?$format=json`

- HTTP Header:

`Accept: application/json`

\$format=atom (default)

```
468         <d:city_id>31</d:city_id>
469         <d:city>Arak</d:city>
470         <d:country_id>46</d:country_id>
471         <d:last_update>2006-02-15 03:45:25</d:last_update>
472     </m:properties>
473 </content>
474 </entry>
475 <entry>
476     <id>http://localhost:8080/denodo-odata.svc/movies/city(32)</id>
477     <title type="text">city</title>
478     <updated>2015-07-28T20:34:24.163+02:00</updated>
479     <category term="com.denodo.odata2.city" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
480     <link href="city(32)" rel="edit" title="city"/>
481     <content type="application/xml">
482         <m:properties>
483             <d:city_id>32</d:city_id>
484             <d:city>Arecibo</d:city>
485             <d:country_id>77</d:country_id>
486             <d:last_update>2006-02-15 03:45:25</d:last_update>
487         </m:properties>
488     </content>
489 </entry>
490 <entry>
491     <id>http://localhost:8080/denodo-odata.svc/movies/city(33)</id>
492     <title type="text">city</title>
493     <updated>2015-07-28T20:34:24.164+02:00</updated>
494     <category term="com.denodo.odata2.city" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
495     <link href="city(33)" rel="edit" title="city"/>
496     <content type="application/xml">
497         <m:properties>
498             <d:city_id>33</d:city_id>
499             <d:city>Arlington</d:city>
500             <d:country_id>103</d:country_id>
501             <d:last_update>2006-02-15 03:45:25</d:last_update>
502         </m:properties>
503     </content>
```


\$format=json

```
336     "_metadata": {
337         "id": "http://localhost:8080/denodo-odata.svc/movies/city(31)",
338         "uri": "http://localhost:8080/denodo-odata.svc/movies/city(31)",
339         "type": "com.denodo.odata2.city"
340     },
341     "city_id": 31,
342     "city": "Arak",
343     "country_id": 46,
344     "last_update": "2006-02-15 03:45:25"
345 },
346 {
347     "_metadata": {
348         "id": "http://localhost:8080/denodo-odata.svc/movies/city(32)",
349         "uri": "http://localhost:8080/denodo-odata.svc/movies/city(32)",
350         "type": "com.denodo.odata2.city"
351     },
352     "city_id": 32,
353     "city": "Arecibo",
354     "country_id": 77,
355     "last_update": "2006-02-15 03:45:25"
356 },
357 {
358     "_metadata": {
359         "id": "http://localhost:8080/denodo-odata.svc/movies/city(33)",
360         "uri": "http://localhost:8080/denodo-odata.svc/movies/city(33)",
361         "type": "com.denodo.odata2.city"
362     },
363     "city_id": 33,
364     "city": "Arlington",
365     "country_id": 103,
366     "last_update": "2006-02-15 03:45:25"
367 },
368 {
369     "_metadata": {
370         "id": "http://localhost:8080/denodo-odata.svc/movies/city(34)",
371         "uri": "http://localhost:8080/denodo-odata.svc/movies/city(34)",
372         "type": "com.denodo.odata2.city"
373     },
374     "city_id": 34,
375     "city": "Arlington",
376     "country_id": 103,
377     "last_update": "2006-02-15 03:45:25"
378 },
379 }
```

Querying by primary key: (id)

- A specific ID can be specified between params:

`/denodo-odata.svc/movies/city(32)`

The screenshot shows a REST client interface. At the top, the method is 'GET' and the URL is 'http://localhost:8080/denodo-odata.svc/movies/city(32)'. Below the URL bar, there are tabs for 'Authorization', 'Headers (2)', 'Body', 'Pre-request script', and 'Tests'. The 'Authorization' tab is selected, showing 'Basic Auth'. Below the tabs, the 'Body' tab is selected, showing the JSON response. The response is formatted as 'Pretty' and is a JSON object with the following structure:

```
1 {
2   "d": {
3     "_metadata": {
4       "id": "http://localhost:8080/denodo-odata.svc/movies/city(32)",
5       "uri": "http://localhost:8080/denodo-odata.svc/movies/city(32)",
6       "type": "com.denodo.odata2.city"
7     },
8     "city_id": 32,
9     "city": "Arecibo",
10    "country_id": 77,
11    "last_update": "2006-02-15 03:45:25"
12  }
13 }
```

Associations

- Associations are defined as logical entities in VDP

Association name

☒ Referential constraint

End point 'city'

Role name

Role description

Role precondition [Set precondition](#)

Multiplicity

End point 'country'

Role name

Role description

Role precondition [Set precondition](#)

Multiplicity

Condition mappings [+](#)

☒ ☐ `city.country_id = country.country_id`

| city | | |
|------|-------------|------|
| | city_id | int |
| | city | text |
| | country_id | int |
| | last_update | text |

| country | | |
|---------|-------------|------|
| | country_id | int |
| | country | text |
| | last_update | text |

Navigating associations: /navprop

- Equivalent to RESTfulWS navigations
- Equivalent to SELECT_NAVIGATIONAL queries in VQL

/denodo-odata.svc/movies/country(22)/cities

```
1 {
2   "d": {
3     "results": [
4       {
5         "_metadata": {
6           "id": "http://localhost:8080/denodo-odata.svc/movies/city(27)",
7           "uri": "http://localhost:8080/denodo-odata.svc/movies/city(27)",
8           "type": "com.denodo.odata2.city"
9         },
10        "city_id": 27,
11        "city": "Antofagasta",
12        "country_id": 22,
13        "last_update": "2006-02-15 03:45:25",
14        "country": {
15          "_deferred": {
16            "uri": "http://localhost:8080/denodo-odata.svc/movies/city(27)/country"
17          }
18        },
19        "addresses": {
20          "_deferred": {
21            "uri": "http://localhost:8080/denodo-odata.svc/movies/city(27)/addresses"
22          }
23        }
24      },
25      {
26        "_metadata": {
27          "id": "http://localhost:8080/denodo-odata.svc/movies/city(127)",
28          "uri": "http://localhost:8080/denodo-odata.svc/movies/city(127)",
29          "type": "com.denodo.odata2.city"
30        },
31        "city_id": 127,
32        "city": "Coquimbo",
33        "country_id": 22,
34        "last_update": "2006-02-15 03:45:25",
35        "country": {
36          "_deferred": {
```

Selection: \$filter

■ Request parameter, WHERE clause in SELECT statement

`/denodo-odata.svc/movies/city?$filter=startswith(city,'Bu') eq true`

```
1 {
2   "d": {
3     "results": [
4       {
5         "__metadata": {
6           "id": "http://localhost:8080/denodo-odata.svc/movies/city(95)",
7           "uri": "http://localhost:8080/denodo-odata.svc/movies/city(95)",
8           "type": "com.denodo.odata2.city"
9         },
10        "city_id": 95,
11        "city": "Bucuresti",
12        "country_id": 78,
13        "last_update": "2006-02-15 03:45:25",
14        "country": {
15          "__deferred": {
16            "uri": "http://localhost:8080/denodo-odata.svc/movies/city(95)/country"
17          }
18        },
19        "addresses": {
20          "__deferred": {
21            "uri": "http://localhost:8080/denodo-odata.svc/movies/city(95)/addresses"
22          }
23        }
24      },
25      {
26        "__metadata": {
27          "id": "http://localhost:8080/denodo-odata.svc/movies/city(96)",
28          "uri": "http://localhost:8080/denodo-odata.svc/movies/city(96)",
29          "type": "com.denodo.odata2.city"
30        },
31        "city_id": 96,
32        "city": "Buenaventura",
```

Selection operators: eq, ne, gt, ge...

- Currently implemented

| | | |
|-----|----|-----|
| eq | gt | lt |
| ne | ge | le |
| and | or | not |

Selection functions

- Currently implemented

`startswith()`

`substringof()`

`indexof()`

Projection: \$select

■ Selection of specific fields

/denodo-odata.svc/movies/customer?\$select=first_name,last_name,email

```
1 {
2   "d": {
3     "results": [
4       {
5         "__metadata": {
6           "id": "http://localhost:8080/denodo-odata.svc/movies/customer(1)",
7           "uri": "http://localhost:8080/denodo-odata.svc/movies/customer(1)",
8           "type": "com.denodo.odata2.customer"
9         },
10        "first_name": "MARY",
11        "last_name": "SMITH",
12        "email": "MARY.SMITH@sakilacustomer.org"
13      },
14      {
15        "__metadata": {
16          "id": "http://localhost:8080/denodo-odata.svc/movies/customer(2)",
17          "uri": "http://localhost:8080/denodo-odata.svc/movies/customer(2)",
18          "type": "com.denodo.odata2.customer"
19        },
20        "first_name": "PATRICIA",
21        "last_name": "JOHNSON",
22        "email": "PATRICIA.JOHNSON@sakilacustomer.org"
23      },
24      {
25        "__metadata": {
26          "id": "http://localhost:8080/denodo-odata.svc/movies/customer(3)",
27          "uri": "http://localhost:8080/denodo-odata.svc/movies/customer(3)",
28          "type": "com.denodo.odata2.customer"
29        },
30        "first_name": "LINDA",
31        "last_name": "WILLIAMS",
32        "email": "LINDA.WILLIAMS@sakilacustomer.org"
33      }
34    ]
35  }
36 }
```


Ordering: \$orderby

■ Order by specific properties...

`/denodo-odata.svc/movies/country?$orderby=country desc`

```
1 {
2   "d": {
3     "results": [
4       {
5         "_metadata": {
6           "id": "http://localhost:8080/denodo-odata.svc/movies/country(109)",
7           "uri": "http://localhost:8080/denodo-odata.svc/movies/country(109)",
8           "type": "com.denodo.odata2.country"
9         },
10        "country_id": 109,
11        "country": "Zambia",
12        "last_update": "2006-02-15 03:44:00",
13        "cities": {
14          "_deferred": {
15            "uri": "http://localhost:8080/denodo-odata.svc/movies/country(109)/cities"
16          }
17        }
18      },
19      {
20        "_metadata": {
21          "id": "http://localhost:8080/denodo-odata.svc/movies/country(108)",
22          "uri": "http://localhost:8080/denodo-odata.svc/movies/country(108)",
23          "type": "com.denodo.odata2.country"
24        },
25        "country_id": 108,
26        "country": "Yugoslavia",
27        "last_update": "2006-02-15 03:44:00",
28        "cities": {
29          "_deferred": {
30            "uri": "http://localhost:8080/denodo-odata.svc/movies/country(108)/cities"
31          }
32        }
33      }
34    ]
35  }
36 }
```

Pagination: \$top, \$skip, \$skiptoken

- Request-defined pagination:

```
/denodo-odata.svc/movies/customer?$skip=5&$top=3
```

- Server response includes \$skiptoken

```
{  
  "value": [  
    {  
      "__next": "http://localhost:8080/denodo-odata.svc/movies/customer?$skip=5&$top=3&$skiptoken=1"  
    }  
  ]  
}
```

- Max page size: 1000 results

Pagination: \$inlinecount

■ Returns total results

`/denodo-odata.svc/movies/customer?$skip=5&$top=3&$inlinecount=allpages`

```
{
  "d": {
    "__count": "599",
    "results": [
      {
        "__metadata": {
          "id": "http://localhost:8080/denodo-odata.svc/movies/customer(6)",
          "uri": "http://localhost:8080/denodo-odata.svc/movies/customer(6)"
        }
      }
    ]
  }
}
```

Complex types: structs

- Structs are correctly handled by the OData Service

```
<entry>
  <id>http://localhost:8080/denodo-odata.svc/admin/struct_table_film(1M)</id>
  <title type="text">struct_table_film</title>
  <updated>2015-07-29T16:02:30.532+02:00</updated>
  <category term="com.denodo.odata2.struct_table_film" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <link href="struct_table_film(1M)" rel="edit" title="struct_table_film"/>
  <content type="application/xml">
    <m:properties>
      <d:table_id>1</d:table_id>
      <d:film_data m:type="com.denodo.odata2.struct_table_film_film_data">
        <d:id>1</d:id>
        <d:title>ACADEMY DINOSAUR</d:title>
        <d:description>ELIZABETH</d:description>
      </d:film_data>
    </m:properties>
  </content>
</entry>
```

Complex types: arrays

- OData 2.0 does not support arrays, default to String rep

```
<entry>
  <id>http://localhost:8080/denodo-odata.svc/admin/people_tab()</id>
  <title type="text">people_tab</title>
  <updated>2015-07-29T16:05:01.002+02:00</updated>
  <category term="com.denodo.odata2.people_tab" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <link href="people_tab()" rel="edit" title="people_tab"/>
  <content type="application/xml">
    <m:properties>
      <d:group_no>100</d:group_no>
      <d:people_column>[1 John Smith 1-650-555-0135 , 2 Diane Smith null ]</d:people_column>
    </m:properties>
  </content>
</entry>
```

Main Limitations

- **Read-only**
- Limited sets of operators and functions
- No support for the **\$expand** command
- Complex types:
 - **No array types** in OData 2.0 ☹️
 - Navigation of associations with complex endpoints

THE TOOLKIT

Denodo OData Service ToolKit

■ Contents

- Denodo OData Service distribution (beta):
 - Denodo OData Service software (beta)
 - Denodo OData Service user manual
- This slide deck

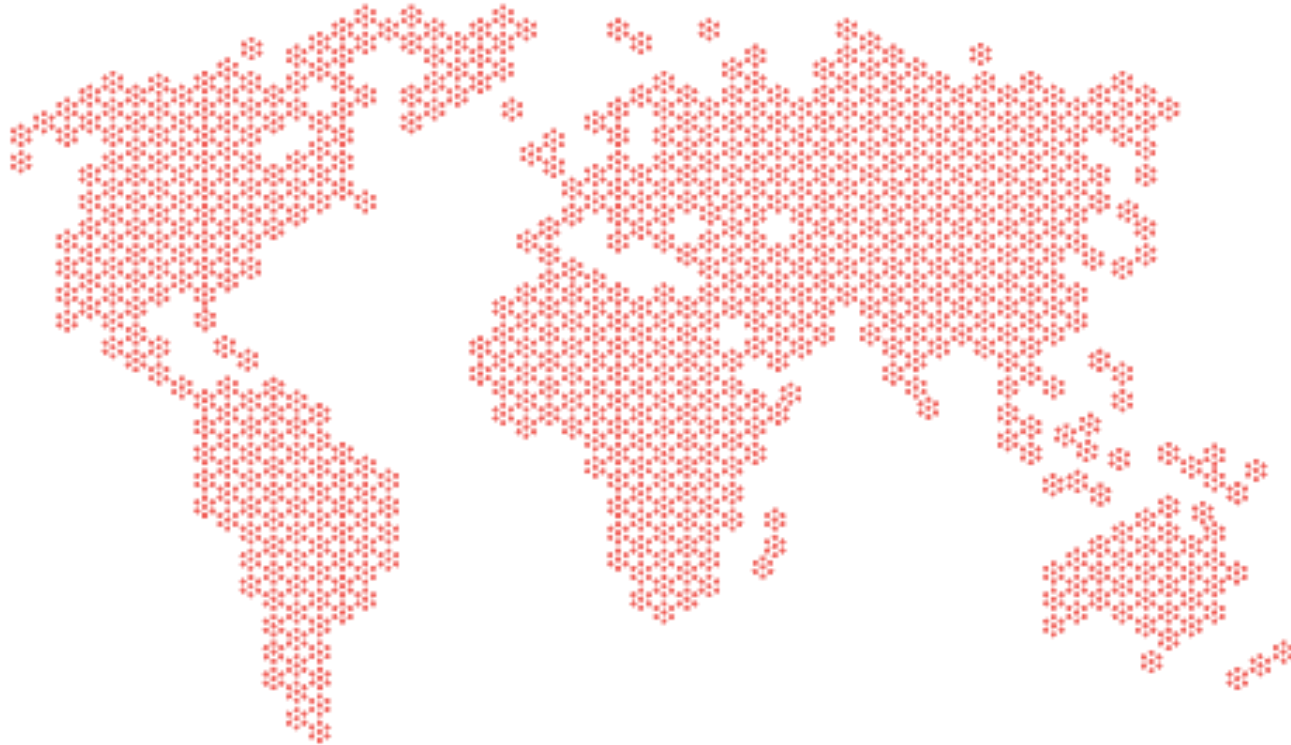
■ Available after this meeting

Q&A

Thank you!



www.denodo.com | info@denodo.com



© **Copyright Denodo Technologies All rights reserved**

Unless otherwise specified, no part of this PDF file may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without prior the written authorization from Denodo Technologies.

