LENS2ODATA

7/27/2016

Lens2odata: Query, Search, and Review OData sources

lens2odata

Revision History:

Date	Version	Description	Author
7/27/2016	1.0	Initial Document	PJL

lens2odata

Table of Contents

Figures

Tables

INTRODUCTION

Objectives

The objectives of lens2odata are to provide a simple method of OData query construction driven by the metadata provide by OData services

- Provides metamodel-driven OData query construction
  - Eliminates any configuration required to expose any OData service to lens2odata
- Allows searches to be saved and rerun
  - Allows ease of use by casual users
- Allows queries to be pinned to 'Lens' dashboard panels
  - Provides simple-to-use dashboard
- Searches can be parameterized
  - Allows for easy configuration of queries
- Compatible with odata2sparql, a service that exposes any triple store as an OData service

Concept of Operation

Lens2odata consists of 3 primary pages with which users interact:

1. Query: is the page in which users can
   a. add new OData services,
   b. compose queries,
   c. save those queries for reuse, and
   d. pin the queries as result fragments on a Lens
2. Search: is the page in which users can
   a. select an existing query, and execute that query to explore the results
   b. from where they can navigate to Lens pages for specific entities or collections of entities
3. Lens: are the pages, composed by users, which display fragments of details, optionally grouped into tabs, about a specific entity or collections of entities. Fragments can either be forms or tables. Other fragment layouts are being added.

Navigation between these pages are shown in the diagram below. Specifically these navigation paths are:

1. Toggle between Search and Query to explore how the results would appear to a casual user
2. Navigate to a concept's Lens from Query preview hyperlinks
3. Navigate to a concept's Lens from Search results' hyperlinks

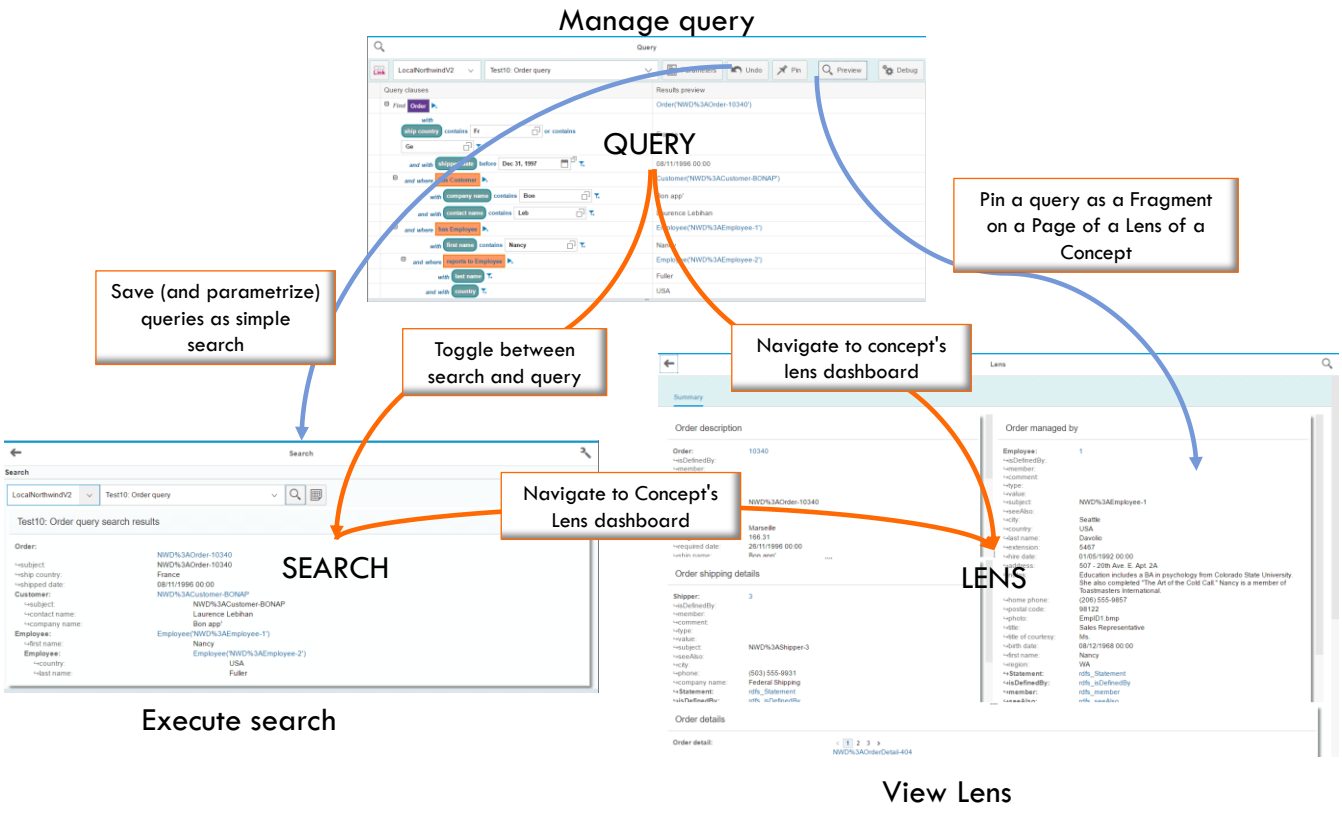


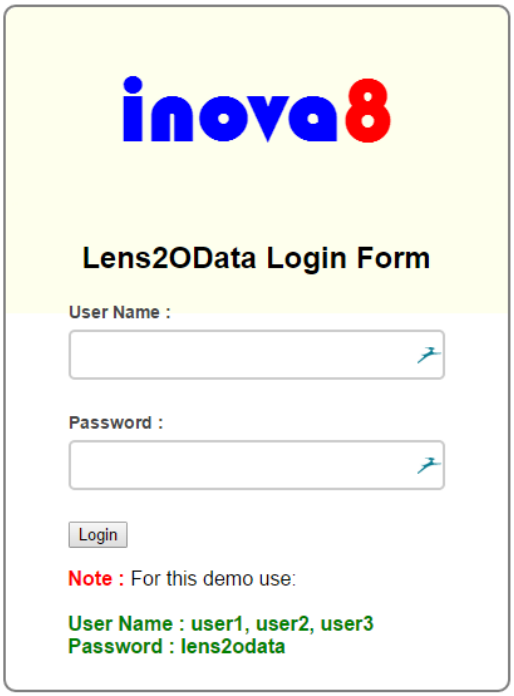
FIGURE 1: LENS2ODATA NAVIGATION

QUICK START

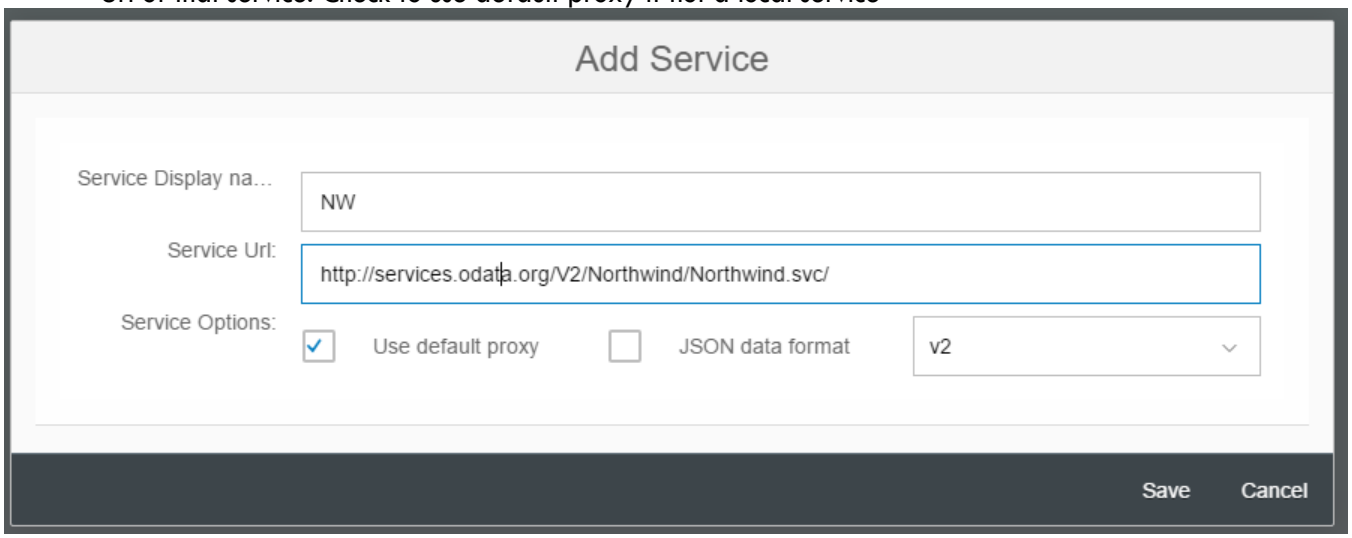
The following assumes that you have neither logged on before, nor that the lens2odata administrator has predefined any services and/or queries.

Login to lens2odata

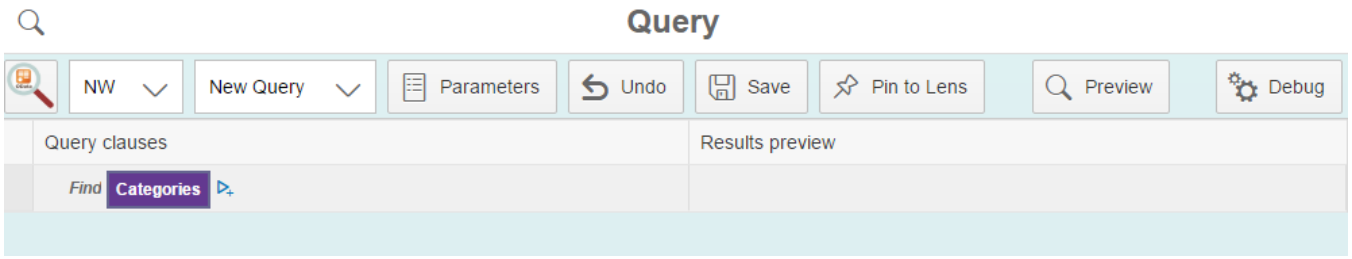
1. Navigate to the Url provided by your administrator for lens2odata, http://<server>/lens2odata
2. Enter users name and password



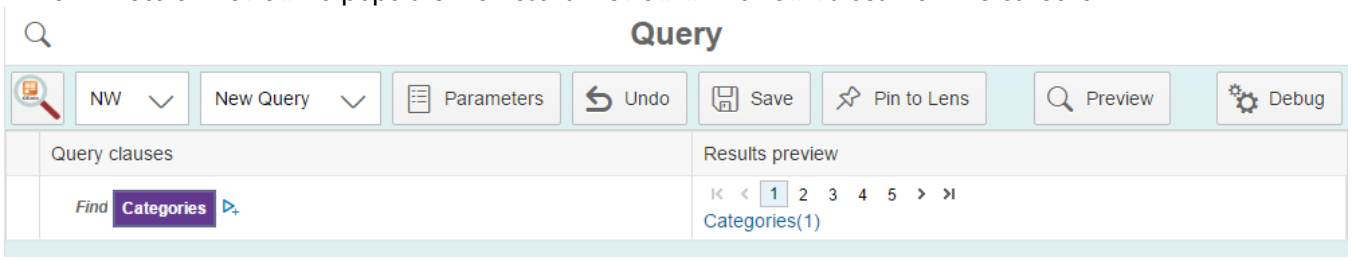
3. Since no service has been previously setup, you will be prompted to enter the service display name and Url of that service. Check to use default proxy if not a local service



4. After 'save', as long as your service was validated, you will immediately enter the Query page with a new query initialized with the first collection found in the service



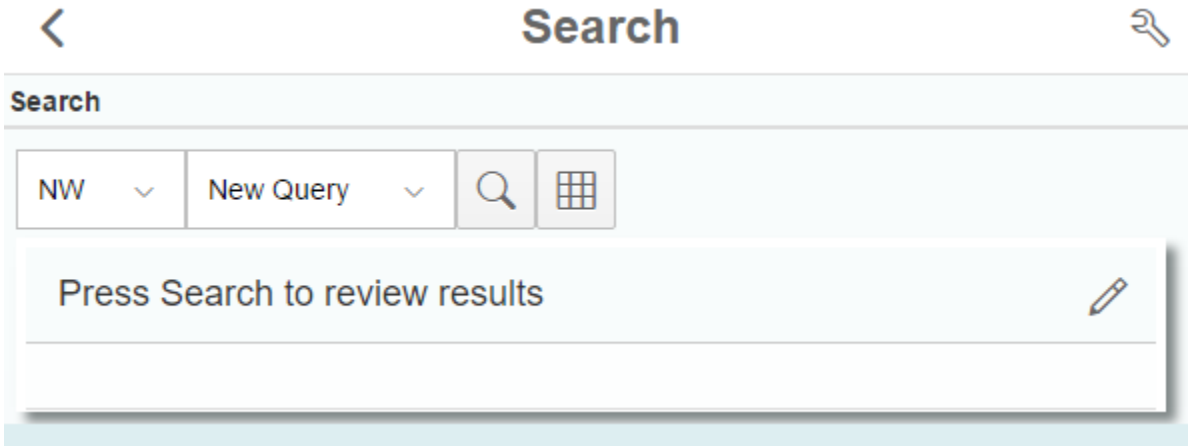
5. Execute 'Preview' to populate the Results Preview with a few values from the collection:



6. You are now ready to:
   a. View the query via Search
   b. Explore the results further via Lens
   c. Expand the query with more values and filters

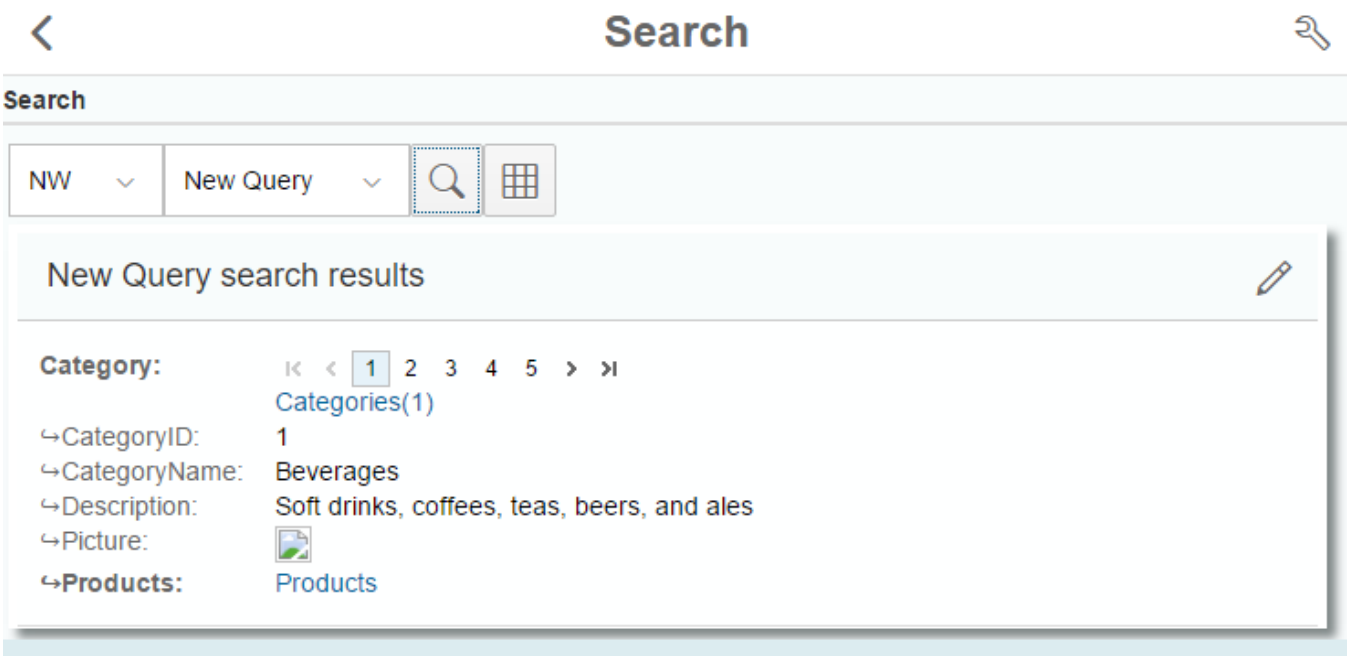
   Let's explore Search first of all

7. Click on  at the top-left of the Query page to navigate to the search page:



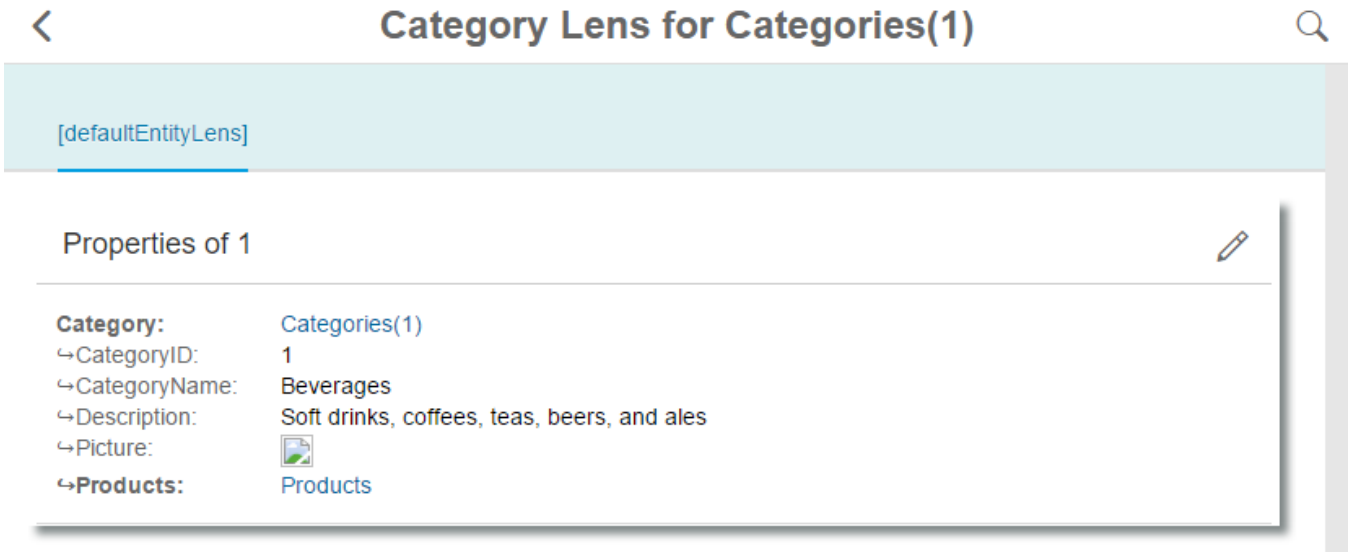
   Search is the page that general users will access. From this page they can select a predefined query, and execute that query to start their discovery journey.

8. Click on  to populate the results form:



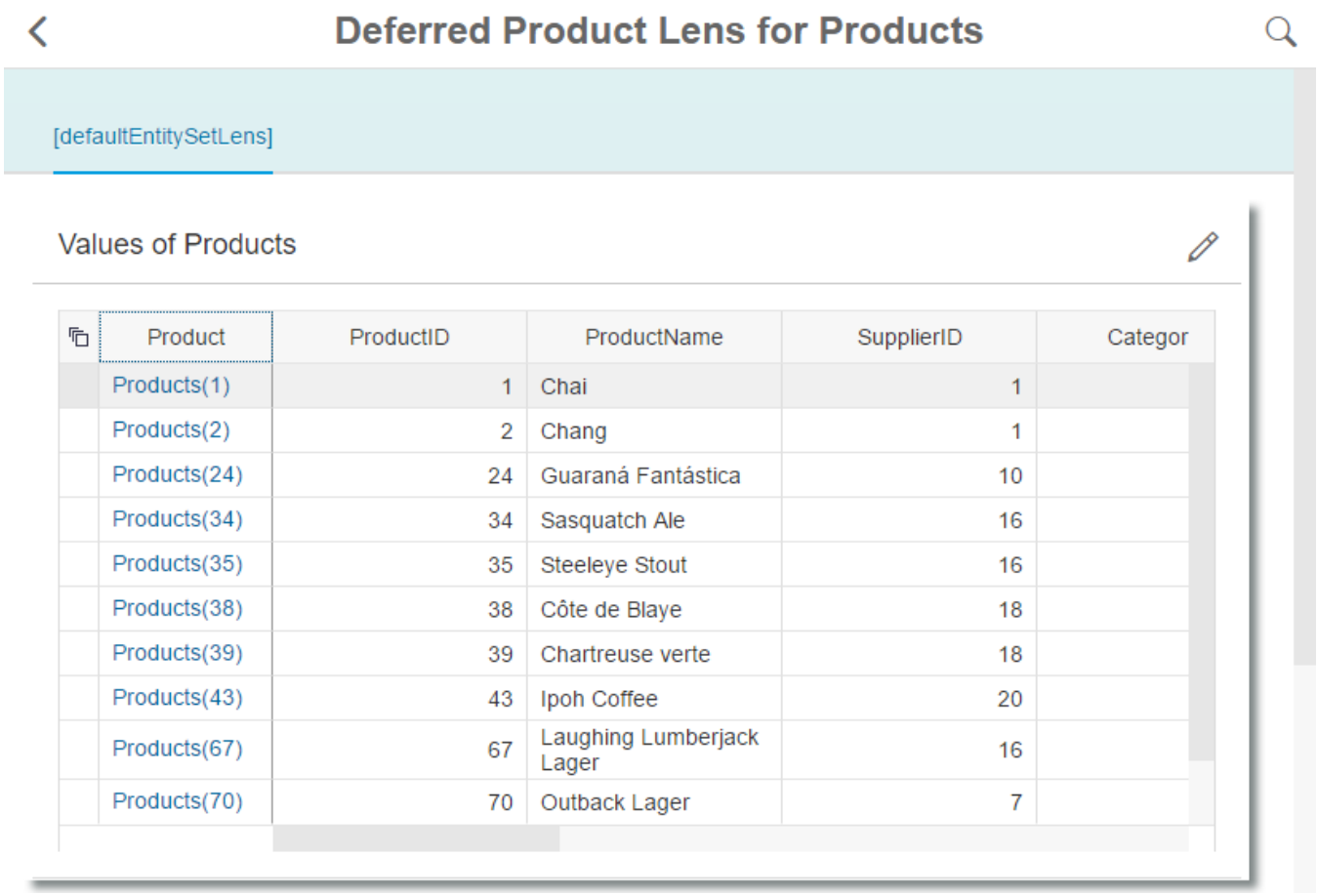
This shows more details than the Query page because, in the absence of any specific definition about what details of the instances of collections should be displayed, search will display whatever it can find.

9. Click on the Url "Categories(1)" to navigate to the Lens for that type of instance.



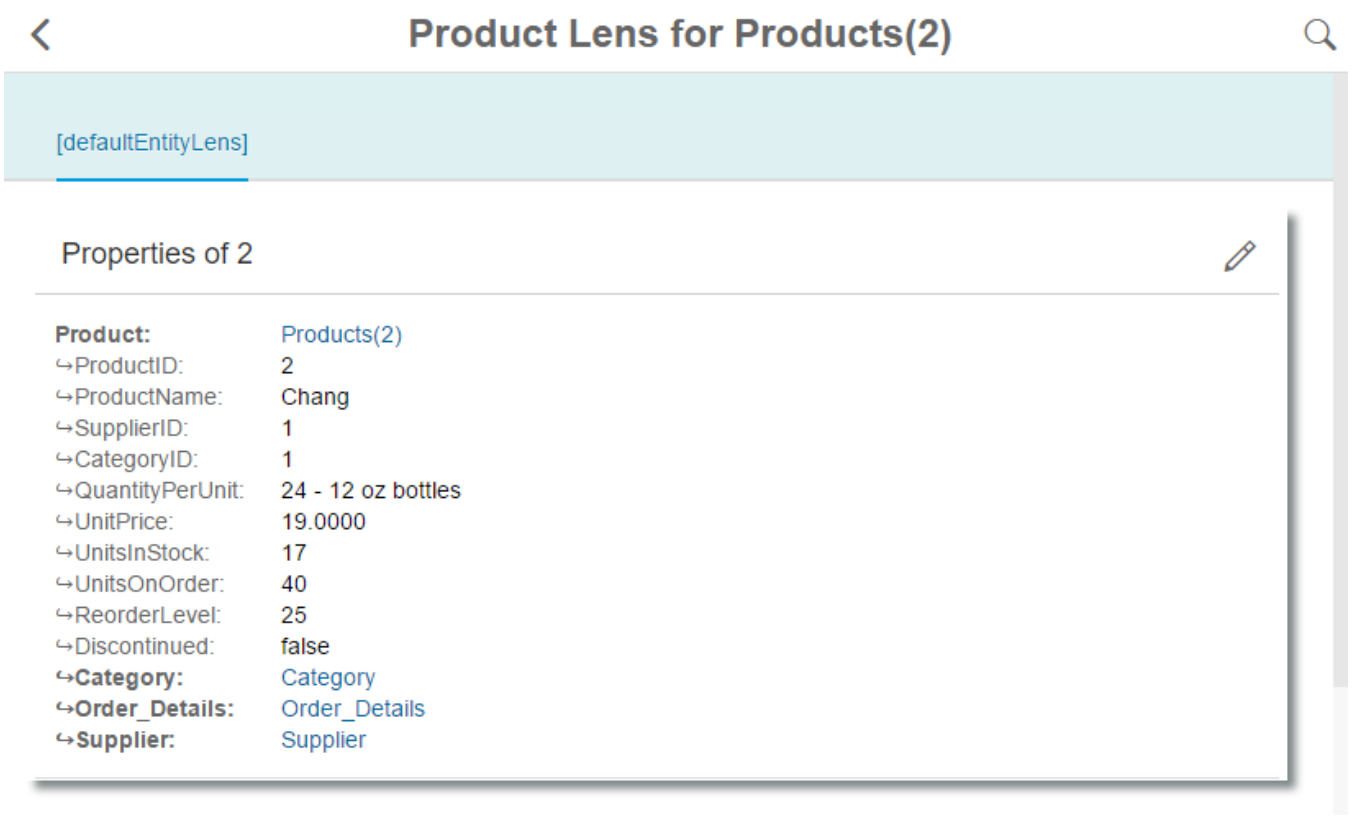
The Lens page is an information dashboard that can be constructed for any type of instance that is discovered. In this case, since no specific lens page has been setup for 'Category' types of instance, a default page has been used with a single tab.

10. There is another Uri on this page "Products". Navigating this link will take you to a similar lens page, but one for a collection of instances:



Product	ProductID	ProductName	SupplierID	Categor
Products(1)	1	Chai	1	
Products(2)	2	Chang	1	
Products(24)	24	Guaraná Fantástica	10	
Products(34)	34	Sasquatch Ale	16	
Products(35)	35	Steeleye Stout	16	
Products(38)	38	Côte de Blaye	18	
Products(39)	39	Chartreuse verte	18	
Products(43)	43	Ipoh Coffee	20	
Products(67)	67	Laughing Lumberjack Lager	16	
Products(70)	70	Outback Lager	7	

11. This Lens for Products shows a list of instances of products. The first column is a Uri to the lens page for the individual product. Click one to navigate to its Lens:



12. You are now discovering information by navigating through the data, having started by querying a collection. Next steps would be:
    - The original query was not particularly specific. Lens2odata allows that query to be further refined by specifying which attributes should be displayed, adding navigation properties to other entities, adding filters to the query to restrict results or even parameterizing the query so that a user can execute the saved query, modifying the results just by supplying parameter values.
    - The Lens dashboard page can be further refined by adding more fragments of information on a tab, or adding more tabs to the lens page to logically group the information.

USER GUIDE

Lens2odata startup

Assuming that the lens2odata application service is running, access to lens2odata is via the Url supplied by your administrator, typically of the form: <server>/lens2odata.
On startup a user will be challenged for their username and password:

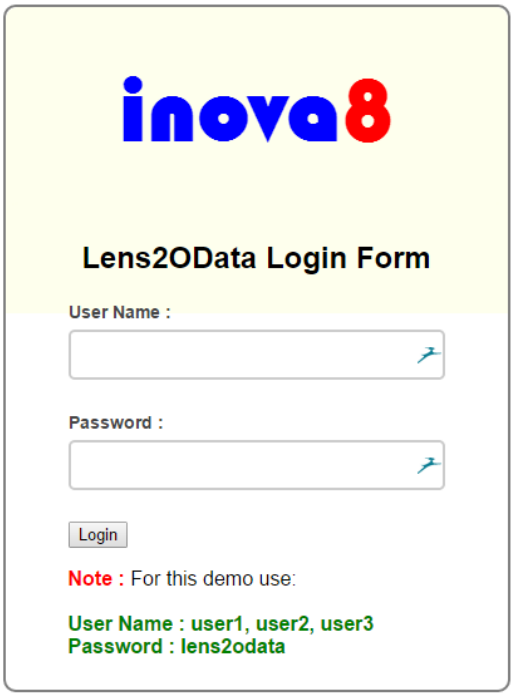


FIGURE 2: LENS2ODATA LOGIN

On login the users saved state will attempt to be recovered. Note that the state of the user's last lens2odata session is saved to the lens2odata server, thus allowing access from different devices. The logic for recovering any prior saved state is as follows:

- If neither the administrator has set up a default set of services and queries, nor the user has saved a prior state, then the application will wait on the user to add a new OData service via the Query page.
- If the administrator has set up a default set of services and queries, then the user's session will be initialized with those services and queries. These will be saved along with any other changes the users makes to the users own lens2odata state.
- If the user has saved a prior lens2odata state, then these services and queries will be merged with any services and queries setup by the administrator.

It is also possible for a user to enter lens2odata Search, Query, or Lens page directly by providing the complete Url of that page.

Query

The Query page is the main management page of lens2odata allowing for:

- The management of services
- The management of queries
- The interactive construction of queries and previewing the query results

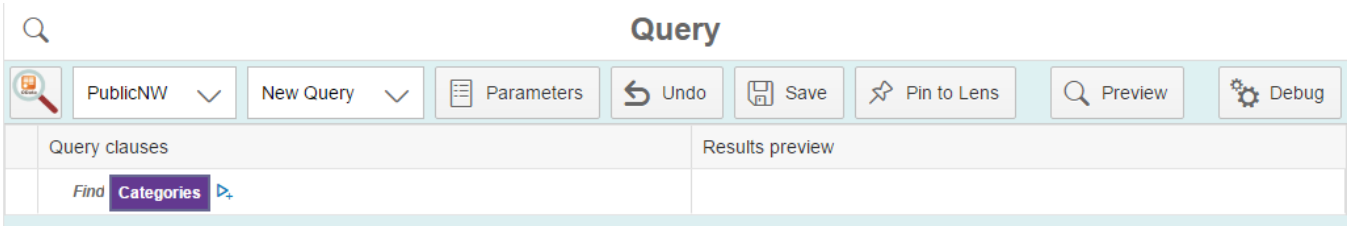


FIGURE 3: QUERY PAGE

Manage Services

The Manage Services menu allows:

- Selection of a pre-verified service
- Deletion of the currently selected service
- Addition of a new service
- Editing the currently selected service

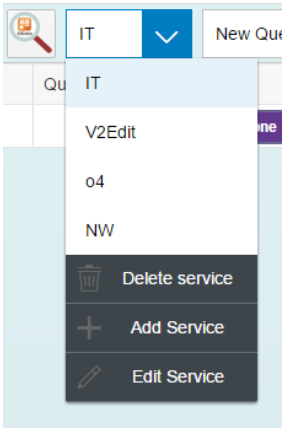


FIGURE 4: MANAGE SERVICES

SELECTION OF A PRE-VERIFIED SERVICE

A pre-verified service can be selected from the drop-down list. When selected lens2odata will verify that the service is available. If available lens2odata will either select the first currently defined query for that service or create a new, but empty, query for that service:

DELETION OF THE CURRENTLY SELECTED SERVICE

A user can opt to delete the currently selected service. Since deleting that service will also delete all saved queries, then the user is prompted for confirmation prior to deletion.

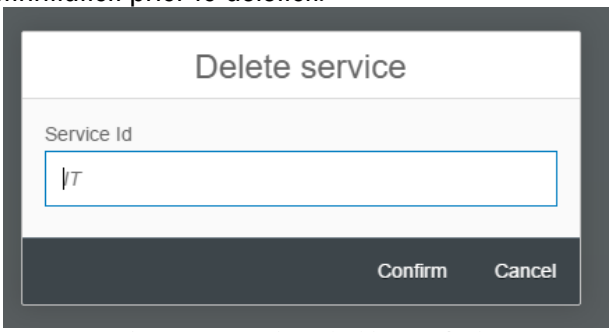


FIGURE 5: DELETE SERVICE CONFIRMATION

ADDITION OF A NEW SERVICE

A new service can be added and verified with Add service:

- Each service requires a Service Display Name that is used as an alias for the service URL.
- The service must have a Service Url. If the default Proxy built into lens2odata is used then this is the public URL. If the default Proxy is not used then the Service Url must be on the same server hosting lens2odata to avoid CORS issues.
- If the service is not located on the lens2odata server, then Use Default proxy should be checked.
- If only JSON data response format is required then check JSON data format.
- Select the maximum service version, currently 2.0, or 4.0

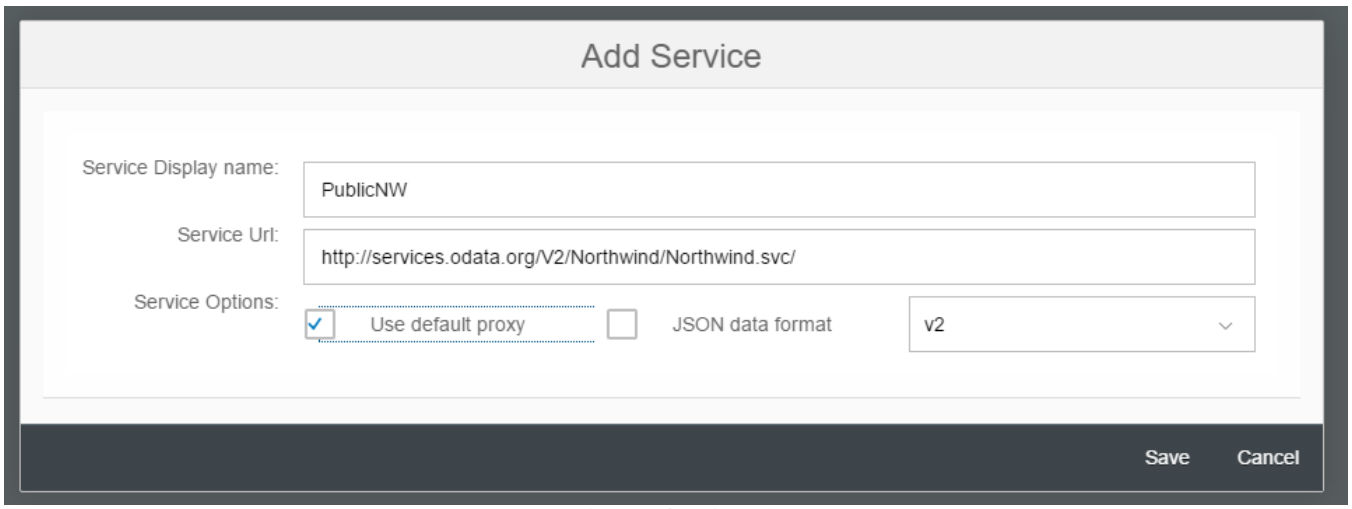


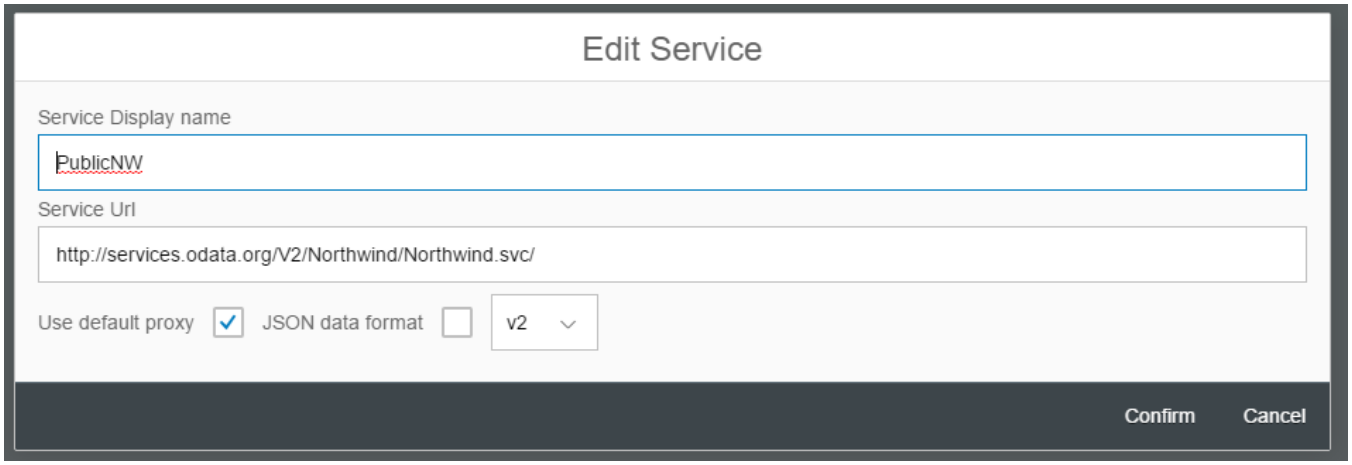
FIGURE 6: ADD SERVICE

When the service is saved, via the Save option, lens2odata validates the service. If the service fails validation the user will be alerted.

EDITING THE CURRENTLY SELECTED SERVICE

The currently selected service can be edited using the edit Service option. Any parameters can be changed:

- Service Display Name,
- Service Url,
- Use of Default proxy,
- JSON data format, and
- OData version.



Confirm commits the changes after verification of the service, whilst cancel reverts back to the original values.

Manage Queries

The Manage Query menu allows:

- Selection of an existing named query
- Deletion of the currently selected query
- Addition of a new query
- Copying (cloning) of the currently selected query under a new name
- Reset of the query down to its simplest form (selection of a collection)
- Edit the query details (name etc)

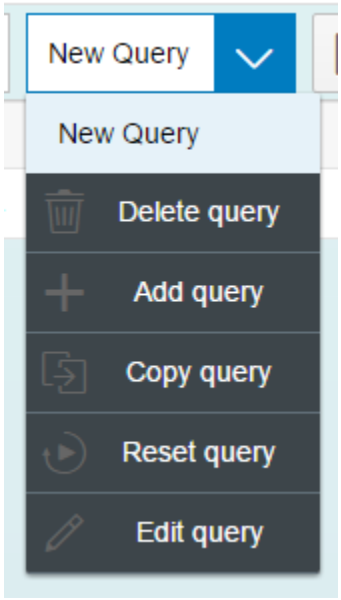


FIGURE 7: MANAGE QUERIES

SELECTION OF AN EXISTING NAMED QUERY

An existing query for the service can be selected from the drop-down list. Once selected the Query Editing tree will be populated with the current state of the query:

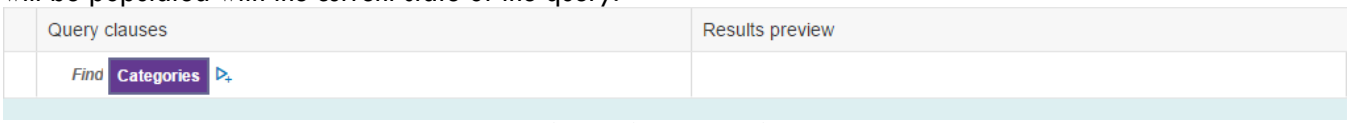


FIGURE 8: QUERY EDITING TREE

DELETION OF THE CURRENTLY SELECTED QUERY

A user can opt to delete the currently selected query. Since deleting that query is unrecoverable, then the user is prompted for confirmation prior to deletion.

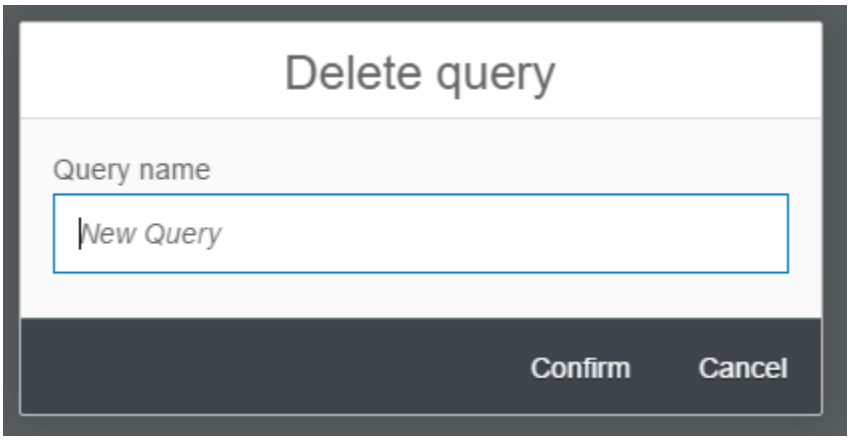


FIGURE 9: DELETE QUERY CONFIRMATION

Note that the query is only deleted from the currently cached services and queries. If the user does not save the state of this cache to server storage, then the query can be recovered by refreshing lens2odata, at which time the saved services and queries are downloaded.

ADDITION OF A NEW QUERY

A user can add a new named query with Add Query within which a new display name is assigned for the query:

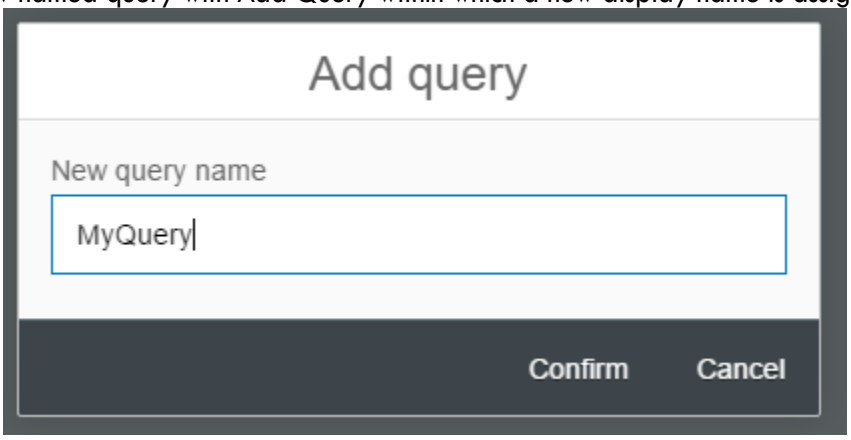


FIGURE 10: ADD QUERY

Note that internally lens2odata uses a unique key to identify each query. Thus the query name can be changed without loss of any integrity.

Once the new query display name has been confirmed, the Query Editing Tree will display the new query, initialize with the first collection from the currently selected service:

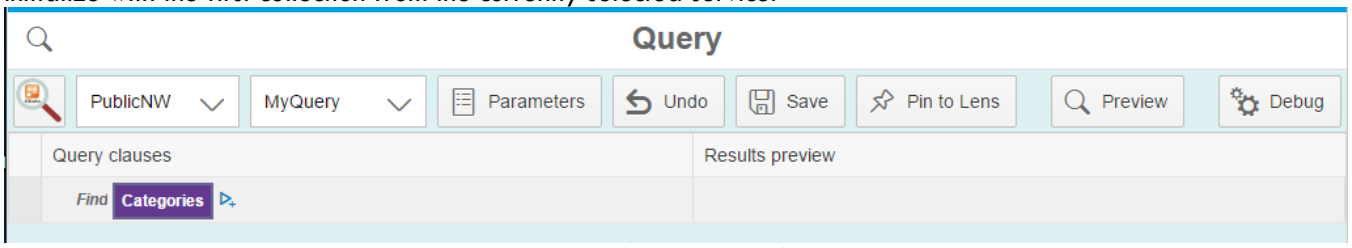


FIGURE 11: ADD QUERY INITIALIZED

COPYING (CLONING) OF THE CURRENTLY SELECTED QUERY UNDER A NEW NAME

Copy Query creates an exact copy (clone) of the currently elected query using a new display name (and correspondingly a new internal identifier):

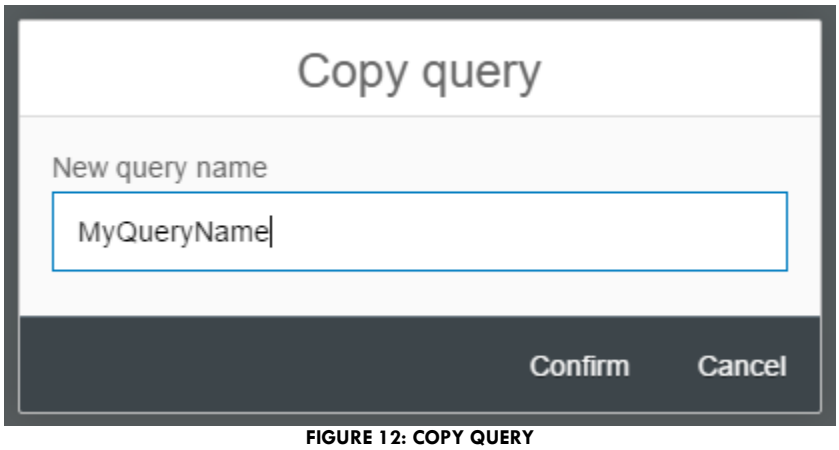


FIGURE 12: COPY QUERY

If confirmed the copied query will become the currently active query and be displayed in the Query editing tree.

RESET OF THE QUERY DOWN TO ITS SIMPLEST FORM (SELECTION OF A COLLECTION)

Reset Query will reduce the query down to its simplest form, removing any navigation of properties from the query:

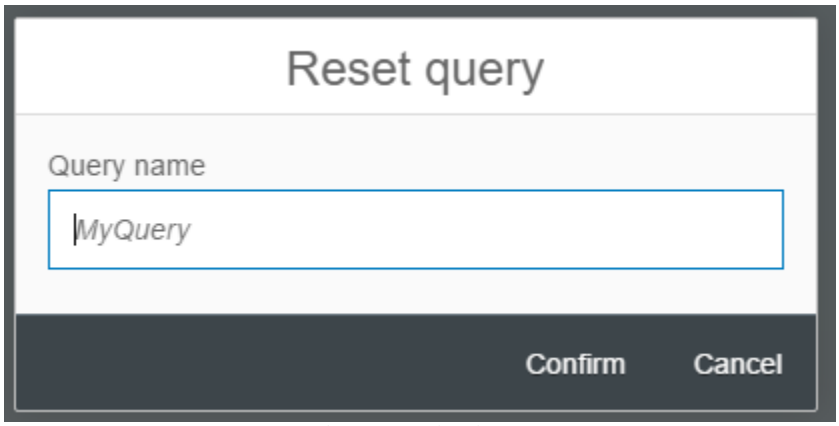


FIGURE 13: RESET QUERY

If confirmed the Query Editing Tree is reset to display only the simplified query

EDIT THE QUERY DETAILS

Edit Query allows the query display name to be changed:

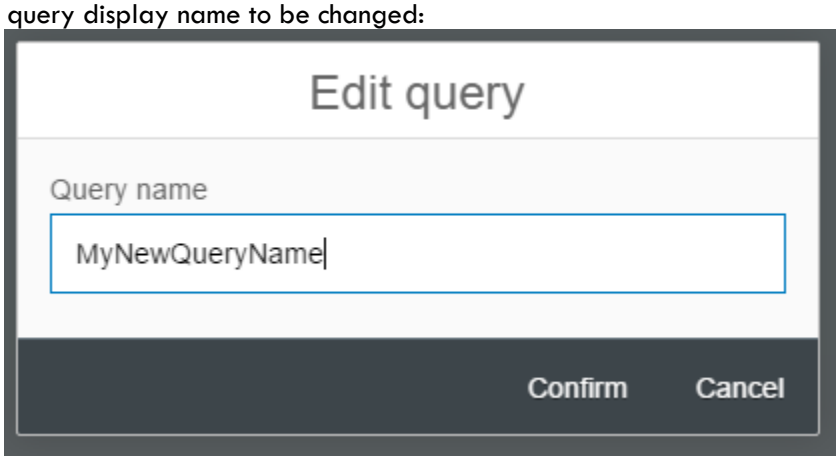


FIGURE 14: EDIT QUERY

Query Editing Tree

The Query Editing Tree is the area within the Query page where the user can compose their query and preview the results.
The Query Editing Tree will always be initialized with a Collection from which the user wishes to find results.

FIND COLLECTION

A 'Collection' is always the starting point of any lens2odata query.

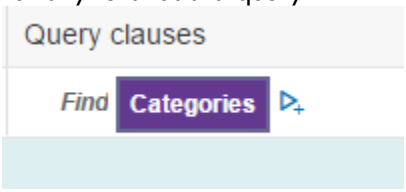


FIGURE 15: FIND COLLECTION

The collection will always be initialized, so if a user wishes to select a different collection they can click on the currently displayed collection

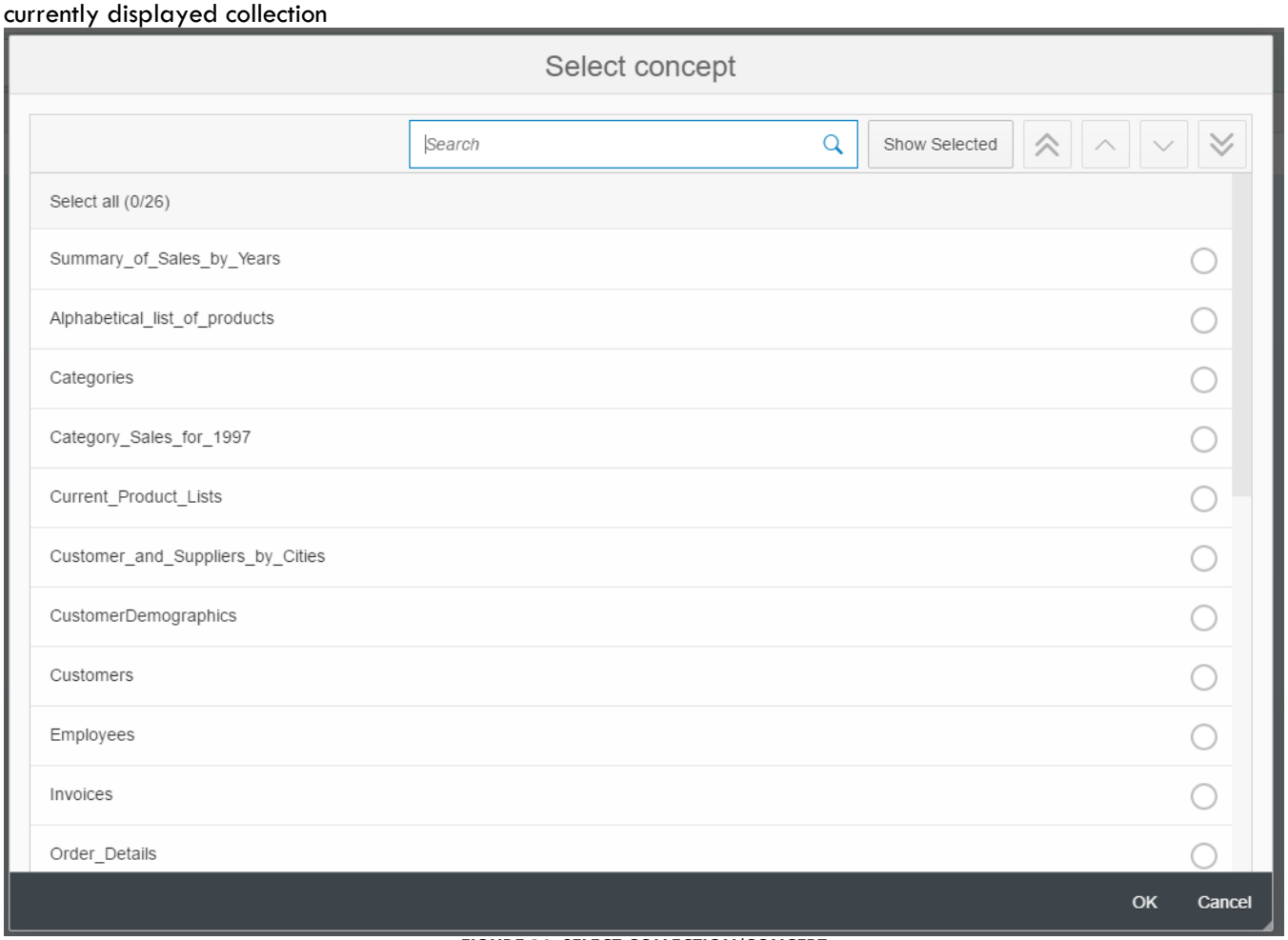


FIGURE 16: SELECT COLLECTION/CONCEPT

The select Collection/Concept dialog allows one to search for a collection/concept, but only one can be selected. If selected the collection/concept will replace the currently selected collection/concept.

PREVIEW RESULTS

The user can now preview the contents of the collection using the Preview button. This will return a limited set of results. This helps compose the query, eliminating queries with no results early on in the query construction process

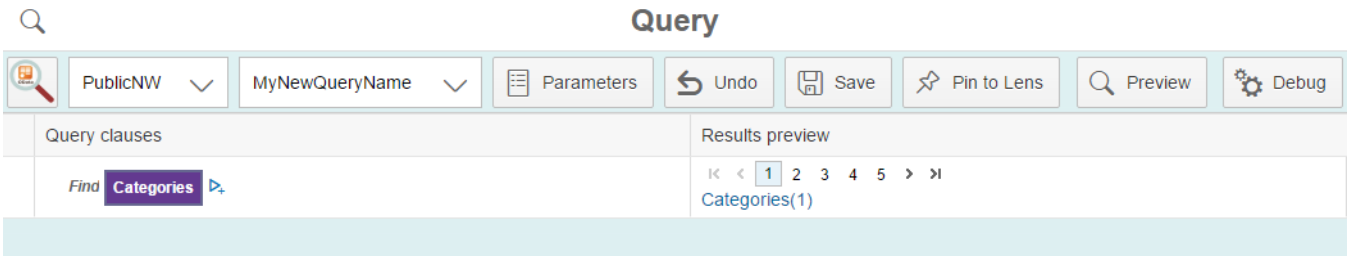


FIGURE 17: PREVIEW COLLECTION

Since only the identity of the collection has been selected in the Query Tree, the Results preview only shows that identity, but as a navigable Url. Thus if the user wants to see more results about an instance in this collection they can:

- Navigate to the Lens: Simply clicking on the Url in the results preview column will take the user to a Lens page that will display more details of selected instance. Even if a Lens page has not been defined for the particular instance type, a default will be displayed showing all results:

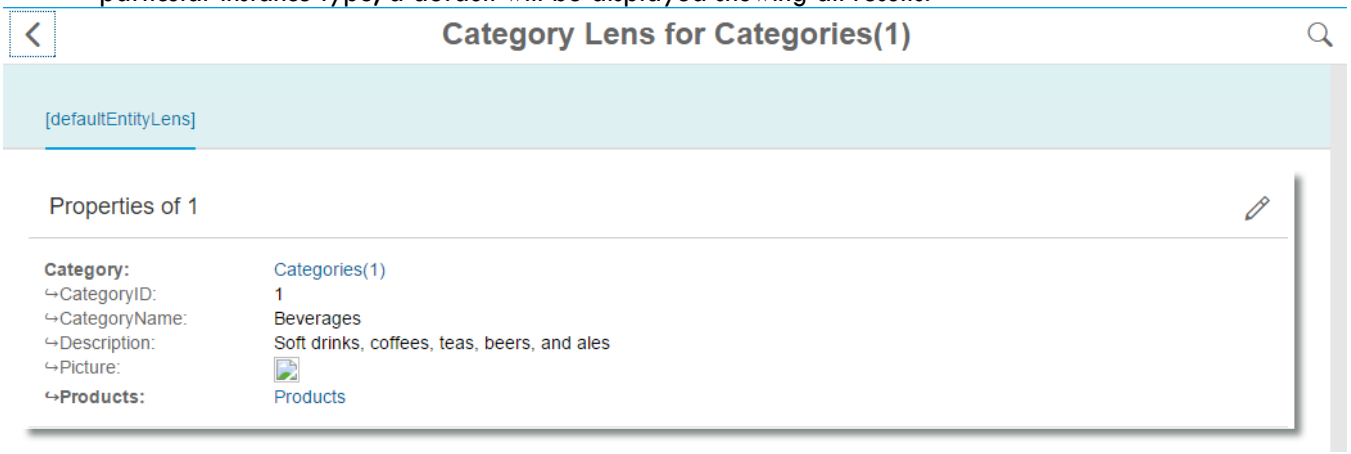


FIGURE 18: DEFAULT LENS PAGE FOR ENTITY INSTANCE

Note that even the entity instances on the default lens page can be used to navigate to more details, such as Products in the above example

- Navigate to the Search page: A user can navigate to the 'Search page using the  icon at top-left on the Query page. The search page is intended for users that do not want to manipulate the query, just execute that query with, optionally some query parameters.

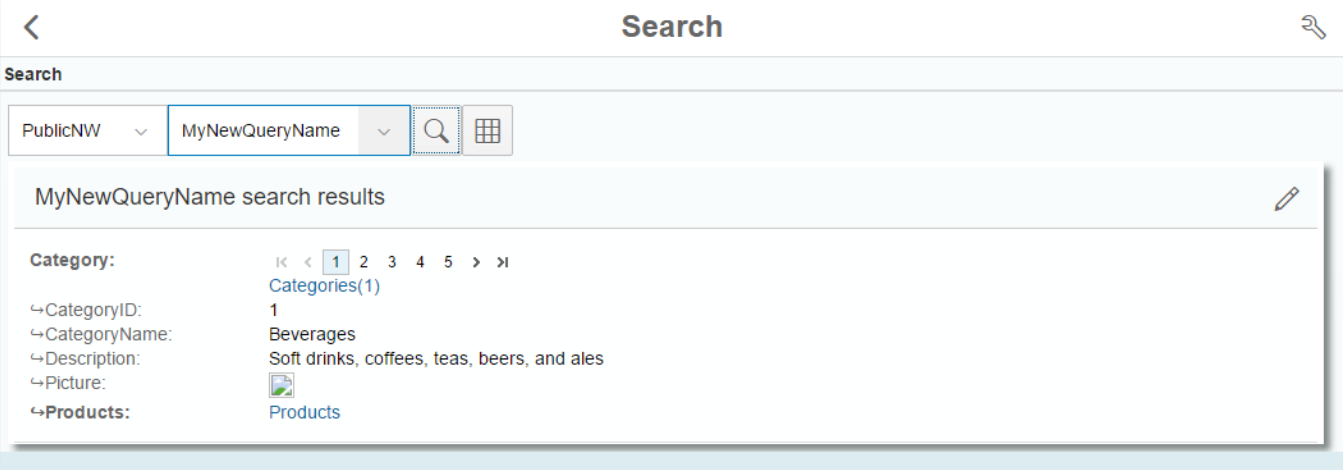


FIGURE 19: SEARCH USING COMPOSED QUERY

EXPANDING DETAILS OF COLLECTION

Once the collection/concept has been selected, one can request more information to be displayed about the chosen collection/concept using the 'expand' button.
This will display a dialog with two panes:

- Navigation Properties/Links: navigation properties or links to other, related, entities. Sometimes known as foreign keys

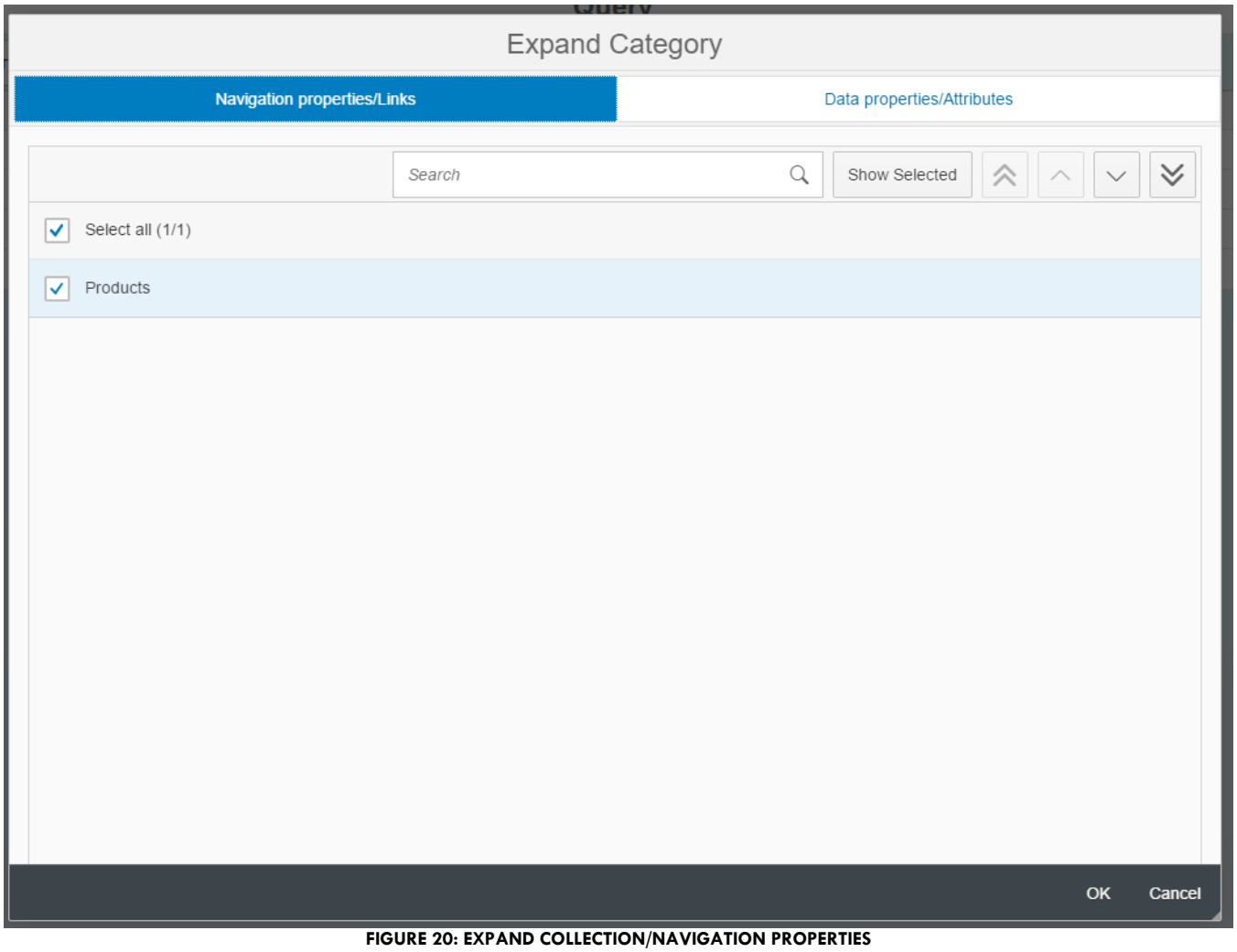


FIGURE 20: EXPAND COLLECTION/NAVIGATION PROPERTIES

- Data Properties/Attributes: properties or attributes of data values associated with the entity. These could be simple scalar properties (strings, dates, numbers, etc.), or complex properties involving dependent attributes. An example of a complex property might be an Address with City, Zip etc.

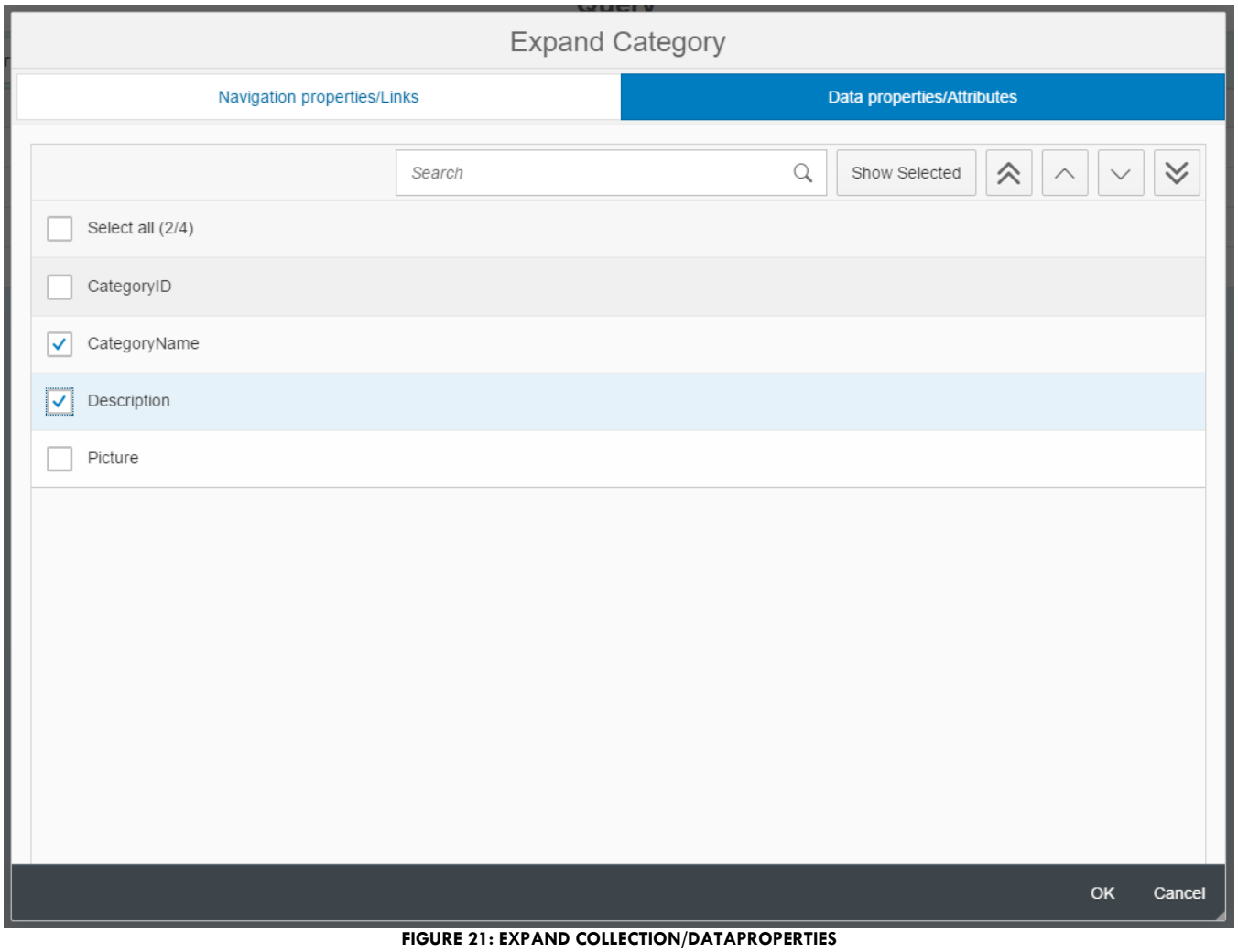


FIGURE 21: EXPAND COLLECTION/DATAPROPERTIES

Multiple navigation properties and/or data properties can be selected from both panes. If confirmed, with OK, these will be added to the existing query as dependencies of the original collection:

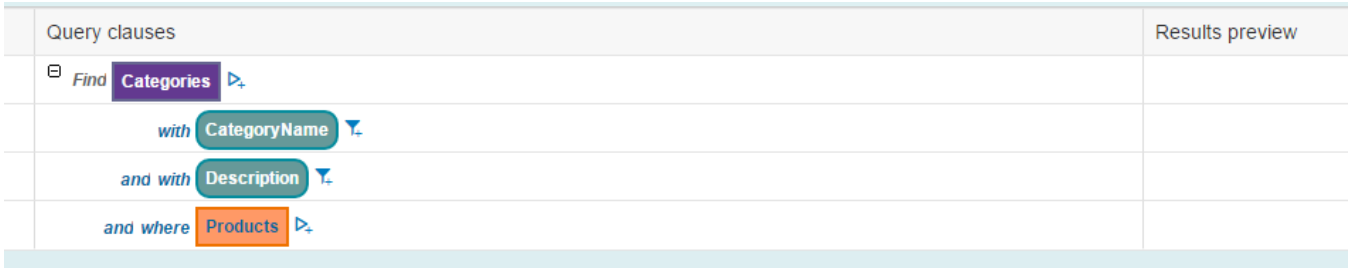


FIGURE 22: QUERY WITH EXPANDED NAVIGATION AND DATA PROPERTIES

The user can now preview these results using the Preview button. This will return a limited set of results. This helps compose the query, eliminating queries with no results early on in the query construction process

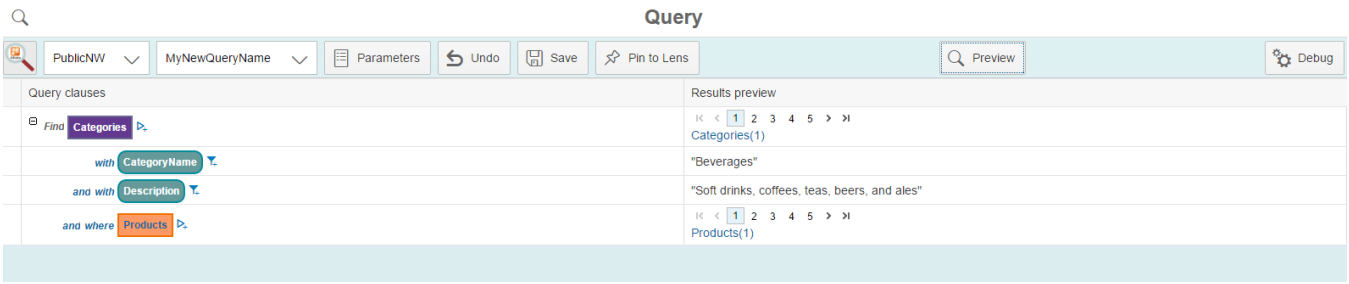


FIGURE 23: PREVIEW QUERY RESULTS

COMPLEX PROPERTIES

DATA PROPERTY FILTERS

Lens2odata provides the ability to add one or more filter conditions on a particular data property within the query. Filters are added by clicking the filter icon at the left of a data property clause:

FILTER PARAMETERS

One of the objectives of lens2odata is to publish pre-defined queries that another user can select and execute. To make this capability more useful, lens2odata allows the values within a data property filter clause to be parameterized. This means that instead of entering a scalar value in the filter, a scalar value is assigned to a parameter. When a user selects to execute a Search based on such a query, then the user will be prompted to enter values for the assigned parameters. Additionally the creator of the query can assign default values for any parameter which will be used in the absence of a user overriding the default value.

FUNCTION IMPORT PARAMETERS

OData specification defines 'FunctionImports' which are similar to database views in that they are a function that may take parameters and return results. The results take the form of a collection of instances of entitytypes (entitysets).
FunctionImports in this regard are very similar to Collections and navigation Properties: they appear in the list of collections from which a user can select, and they might appear in the list of navigation properties a user can select when expanding the query.

Search

Search is the start of a user's search for, and exploration of, data published by the OData service. Search provides an easier to use but less powerful user interface for users. It allows users to select an OData service, chose a predefined query for that service (constructed in the Query page), and then execute that query after entering and prerequisite parameters values for the query.

Lens

Lens provides the information dashboards for the OData services.

ADMINISTRATOR GUIDE

Deployment

1. Lens2oadata is available as an autonomous WAR file.
2. It is built using the OpenUI5 application development framework, however by default the required files are included within the WAR. This avoids any problems of accessing libraries outside of any firewall.

File Locations

1. Ensure that the base/config directory contains at least lenses.json, and optionally queries.json.
2. Queries.json contains the 'reference' queries that will supersede any users' variants of those queries.
3. Users can save their own queries in a directory on the application server /inova8/<users>/
4. The reference queries.json and that of any user uses the same JSON format, so one could copy a particular user's queries.json to the base/config directory to be used as the 'reference' copy.

Authentication and Security

Transport Security

1. The config/servers.xml should define an SSLenabled connector, preferably on port 8443
2. When using lens2odata to access an OData service, such odata2sparql that is on the same host then https://<host>:8443/odata2sparql/2.0/... should be used to avoid a 'mixed mode' error

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS" />
```

User Authentication

1. BASIC authentication, at least, should be defined in the web.xml
   a. Authentication is required so that users may save their query and lens configurations on the server in ../../inova8/config/<user>
2. Users should be defined in config/tomcat-users.xml
3. Users require the role 'lens2odata'

```
<role rolename="lens2odata"/>
<user password="lens2odata" roles="lens2odata" username="user1"/>
```

GLOSSARY

The following is a catalog of the working terms and their associated definitions. These terms will probably change in the final version:

Term	Description	References
Collection	One of the collections of entity instances published by the OData service. They are collections of entitySets published by the EntityContainer or returned by FunctionImports.	
Concept	A synonym for Collection	
EntityType		
Property	An attribute of an entitytype. The attribute might be a scalar or complex type	
NavigationProperty	A property that links one entitytype to another. Commonly referred to as a foreign-key in the RDBMS world, or ObjectProperty in the semantic world.	
ComplexType	A property value that is composed of a set of dependent properties. A typical example is an Address complex type that consists of Street, City Zip etc properties.	

TABLE 1: GLOSSARY