

ODATA CREATION

SEST ZG628T DISSERTATION

by

SUGANDHA VERMA
2015sp93011

Dissertation Work carried out at

SAP Labs, Bangalore

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Rajasthan) India

June, 2017

SEST ZG628T DISSERTATION

ODATA CREATION

Submitted in partial fulfilment of the requirements of

M.S. Software Engineering Degree Program

by

SUGANDHA VERMA
2015sp93011

Under the supervision of

John Alex William, Associate Architect

Dissertation work carried out at

SAP Labs, Bangalore

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)

June, 2017

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Dissertation entitled ODATA Creation and submitted
by Sugandha Verma ID No. 2015sp93011 in partial fulfilment of the requirements
of SEST ZG628T Dissertation, embodies the work done by her under my supervision.



Signature of Supervisor
John Alex William
Associate Architect

Date: June 30,2017

Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be but incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned the efforts with success. It is my privilege to express gratitude and respect to all those who guided me in the completion of this project.

I express my gratitude to **Mr. John Alex William**, Associate Architect, Cloud for Customer Analytics, SAP Labs India Pvt. Ltd., Bangalore for many insightful discussions on design approach, reviews, and, for his inputs.

I would also like to thank SAP Labs India Pvt. Ltd, Bangalore for providing all the necessary infrastructure, software for accomplishing this project and a special thanks to Birla Institute of Technology for opening a door to completion of my Post-Graduation.

Sugandha Verma

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

SEST ZG628T DISSERTATION

FOURTH SEMESTER 2016-2017

Dissertation Title: ODATA Creation

Name of Supervisor : John Alex William

Name of Student : Sugandha Verma

BITS ID No. of Student: 2015sp93011

ABSTRACT

This project contributes to Cloud for Customer Analytics Framework's team. For an analytics framework's team, Dashboards, Reports, Data sources, KPIs play important role for its CRM customers to help them analyze data related to their customers; accessibility to ODATA for these objects will make their day-to-day lives easier by helping them in an extensible manner. The goal is to create ODATA for 'Data Sources' wherein ODATA for Reports has already been exposed to the customers. Reports are built on top of data sources. Hence ODATA for data sources will help customers to access data for 1 level down the hierarchy.

OData (Open Data Protocol) is an OASIS standard that defines a set of best practices for building and consuming RESTful APIs. OData helps to focus on the business logic while building RESTful APIs without having to worry about the various approaches to define request and response headers, status codes, HTTP methods, URL conventions, query options, etc. OData RESTful APIs are easy to consume.

The intension is to create ODATA for analytical Data Sources which will include access to the data through ODATA URLs to get Data Source list, metadata of a particular Data Source.

John Alex William

Signature of Supervisor

Sugandha Verma

Signature of student

TABLE OF CONTENTS

S.No	Content	Page No.
1	Introduction to ODATA	03
1.1	Data Model	06
2.	ODATA Creation for Data Source	08
2.1	Technical view	08
2.2	Function Module for Data Sources' Metadata	09
2.3	Function Module for Data Source List	10
3	Integration of Function Modules to fetch data in EDM format	12
4	Conclusions	13
5	Directions for future work	13
6	Bibliography/ References	13

List of Abbreviations used

API - Application Programming Interface

ABAP – Advanced Business Application
Programming FM - Function Module

C4C - Cloud for Customer

WC - Work Centre

List of Figures

1. Example of JSON data.
2. Example of XML data.
3. Data Model for ODATA
4. Glimpse of Business Analytics WC in SAP Hybris C4C.
5. Initial page when the metadata of DS loads in EDM format
6. Expanded view of metadata of DS in XWL format
7. Initial page when DS list loads with 10 records
8. Expanded view of ODATA for DS list
9. ODATA in SAP Hybris Cloud for customer

CHAPTER 1

Introduction to ODATA

Open Data Protocol (OData) is an open [protocol](#) which allows the creation and consumption of queryable and interoperable [RESTful](#) APIs in a simple and standard way. The protocol enables the creation and consumption of [REST](#) APIs, which allow Web clients to publish and edit resources, identified using URLs and defined in a [data model](#), using simple HTTP messages. OData shares some similarities with [JDBC](#) and with [ODBC](#); like ODBC, OData is not limited to [relational databases](#).

ODATA is a RESTful architecture. The Open Data Protocol (OData) enables the creation of REST-based data services, which allow resources, identified using Uniform Resource Locators (URLs) and defined in a data model, to be published and edited by Web clients using simple HTTP messages. This specification defines the core semantics and the behavioral aspects of the protocol.

OData-URL:

specification defines a set of rules for constructing URLs to identify the data and metadata exposed by an OData service as well as a set of reserved URL query options.

OData-CSDL:

specification defines an XML representation of the entity data model exposed by an OData service.

OData-Atom and **OData-JSON** documents specify the format of the resource representations that are exchanged using OData.

REST approach is the implementation of HTTP or the way HTTP is meant to be used. For ODATA, data is uniquely referenced by URL and can be acted upon using HTTP operations (GET, PUT, POST, DELETE, etc). According to this mapping,

To create a resource on the server, use POST. To retrieve a resource, use GET.

To change the state of a resource or to update it, use PUT.

To remove or delete a resource, use DELETE.

The data is usually sent out by the server in XML or JSON format in HTTP implementation. Examples are shown below:

JSON data

```
{
  name: 'James Bond',
  university: {
    name: 'Oxford',
    country: 'UK'
  },
  gpa: 3.5,
  score: '10'
}
```

Fig 1: Example of JSON data

XML Notation

```
<student>
  <name>James Bond</name>
  <university>
    <name>Oxford</name>
    <country>UK</country>
  </university>
  <gpa>3.5</gpa>
  <score>10</score>
</student>
```

Fig 2: Example of XML data

The OData Protocol is an application-level protocol for interacting with data via RESTful interfaces. The protocol supports the description of data models and the editing and querying of data according to those models. It provides facilities for:

- Metadata: a machine-readable description of the data model exposed by a particular data provider.
- Data: sets of data entities and the relationships between them.
- Querying: requesting that the service perform a set of filtering and other transformations to its data, then return the results.
- Editing: creating, updating, and deleting data.
- Operations: invoking custom logic
- Vocabularies: attaching custom semantics

The OData Protocol is different from other REST-based web service approaches in that it provides a uniform way to describe both the data and the data model. This improves semantic interoperability between systems and allows an ecosystem to emerge.

design principles:

- Prefer mechanisms that work on a variety of data stores. In particular, do not assume a relational data model.
- Extensibility is important. Services should be able to support extended functionality without breaking clients unaware of those extensions.
- Follow REST principles.
- OData should build incrementally. A very basic, compliant service should be easy to build, with additional work necessary only to support additional capabilities.
- Keep it simple. Address the common cases and provide extensibility where necessary.

Reasons to use OData Service:

1. OData is based on the REST architecture, so we can retrieve data based on an URL.
2. It also supports HTTP, Atom Pub as well as JSON format.
3. It has support for any type of data source. Even you can use a custom class as a data source.
4. No need to create a proxy service object. So, it is lightweight to use.
5. You can create your own custom methods and expose it.
6. Since it is lightweight, the interaction between server and client is fast. Thus, performance is good.

1.1 Data Model

The central concepts in the EDM are entities, relationships, entity sets, actions, and functions.

Entities are instances of entity types (e.g. `Customer`, `Employee`, etc.).

Entity types are named structured types with a key. They define the named properties and relationships of an entity. Entity types may derive by single inheritance from other entity types.

The key of an entity type is formed from a subset of the primitive properties of the entity type.

Complex types are keyless named structured types consisting of a set of properties. These are value types whose instances cannot be referenced outside of their containing entity. Complex types are commonly used as property values in an entity or as parameters to operations.

Properties declared as part of a structured type's definition are called *declared properties*. Instances of structured types may contain additional undeclared *dynamic properties*. A dynamic property cannot have the same name as a declared property. Entity or complex types which allow clients to persist additional undeclared properties are called *open types*.

Relationships from one entity to another are represented as navigation properties. Navigation properties are generally defined as part of an entity type, but can also appear on entity instances as undeclared *dynamic navigation properties*. Each relationship has a cardinality.

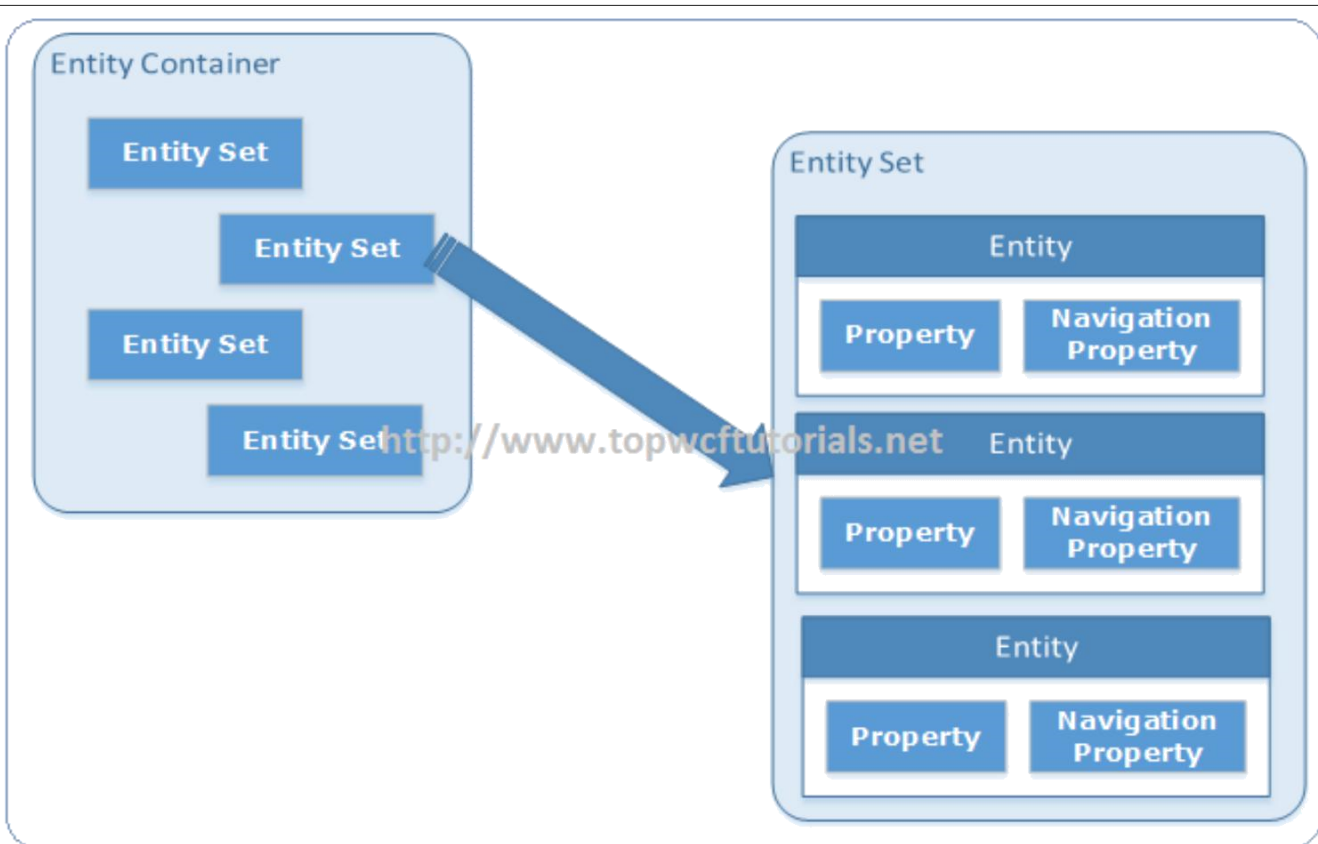


Fig 3: Data Model for ODATA

Enumeration types are named primitive types whose values are named constants with underlying integer values.

Type definitions are named primitive types with fixed facet values such as maximum length or precision. Type definitions can be used in place of primitive typed properties, for example, within property definitions.

Entity sets are named collections of entities. An entity's key uniquely identifies the entity within an entity set. If multiple entity sets use the same entity type, the same combination of key values can appear in more than one entity set and identifies different entities, one per entity set where this key combination appears. Each of these entities has a different entity-id.

CHAPTER 2

ODATA Creation for Data Source

The C4C Analytics team work on frameworks like KPIs, Dashboards, Reports and Data Sources which help customers analyze their data. Since we have already seen the reasons to go for ODATA, it has been implemented for Reports already. Data Sources are used to build reports. There are many kinds of data sources like combined data sources, basic data source, joined data source and cloud data sources which are provided by the business analytics work center in SAP Hybris C4C CRM product.

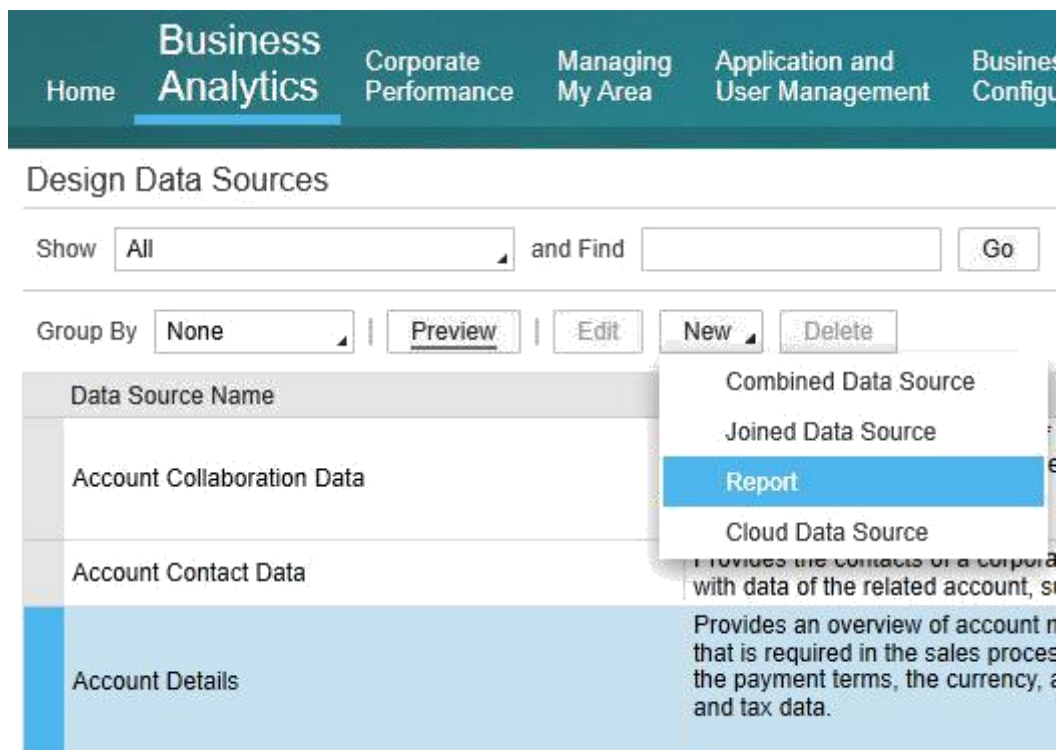


Fig 4: Glimpse of Business Analytics WC in SAP Hybris C4C.

Using ODATA created exclusively for data sources, customers can consume it easily for any other application since the ODATA is easily consumable and provides better performance.

2.1 Technical View

ODATA implementation for Data Sources is done in 2 major function modules through SAP Netweaver platform.

1. Function Module for Data Source Metadata.
2. Function Module for Data Source List.

2.2 Function Module for Data Sources' Metadata

The module1 is for Data Source Metadata. The metadata for a data source essentially provides a view on the MDAV and its master data as a set of related tables. Personalization and parametrization is not supported currently for MDAV metadata. ABAP language is used to create objects in SAP Netweaver platform. Hence, the FMs are created in ABAP.

```
<edmx:Edmx xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" Version="1.0">
  <edmx:DataServices xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" m:DataServiceVersion="1.0">
    <Schema xmlns="http://schemas.microsoft.com/ado/2008/09/edm" Namespace="sapbyd">
      <EntityType xmlns:sap="http://www.sap.com/Protocols/SAPData" Name="DSCODOPPHQueryResult" sap:semantics="aggregate" sap:label="Opportunity Header">
      <EntityContainer Name="cc_home_analytics" m:IsDefaultEntityContainer="true">
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>
```

Fig 5: Initial page when the metadata of DS loads in EDM format

```
<edmx:Edmx xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" Version="1.0">
  <edmx:DataServices xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" m:DataServiceVersion="1.0">
    <Schema xmlns="http://schemas.microsoft.com/ado/2008/09/edm" Namespace="sapbyd">
      <EntityType xmlns:sap="http://www.sap.com/Protocols/SAPData" Name="DSCODOPPHQueryResult" sap:semantics="aggregate" sap:label="Opportunity Header">
        <Key>
          <PropertyRef Name="ID" />
        </Key>
        <Property Name="CNODE_ID" Type="Edm.String" Nullable="true" sap:label="Node ID" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALED_ST_MNTH_N" Type="Edm.String" Nullable="true" sap:label="Month of Predecessor Sales Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALED_ST_QUART_N" Type="Edm.String" Nullable="true" sap:label="Quarter of Predecessor Sales Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALED_ST_WK_DY_N" Type="Edm.String" Nullable="true" sap:label="Weekday of Predecessor Sales Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALED_ST_YMNTN_N" Type="Edm.String" Nullable="true" sap:label="Year/Month of Predecessor Sales Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALED_ST_YQRT_N" Type="Edm.String" Nullable="true" sap:label="Year/Quarter of Predecessor Sales Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALED_ST_YR_N" Type="Edm.String" Nullable="true" sap:label="Year of Predecessor Sales Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALED_ST_YR_WK_N" Type="Edm.String" Nullable="true" sap:label="Year/Week of Predecessor Sales Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALNLED_ST_MNTH_N" Type="Edm.String" Nullable="true" sap:label="Month of Predecessor Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALNLED_ST_QUART_N" Type="Edm.String" Nullable="true" sap:label="Quarter of Predecessor Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALNLED_ST_WK_DY_N" Type="Edm.String" Nullable="true" sap:label="Weekday of Predecessor Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALNLED_ST_YMNTN_N" Type="Edm.String" Nullable="true" sap:label="Year/Month of Predecessor Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALNLED_ST_YQRT_N" Type="Edm.String" Nullable="true" sap:label="Year/Quarter of Predecessor Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALNLED_ST_YR_N" Type="Edm.String" Nullable="true" sap:label="Year of Predecessor Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CALNLED_ST_YR_WK_N" Type="Edm.String" Nullable="true" sap:label="Year/Week of Predecessor Lead" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CBTD_EXT_REF_ID" Type="Edm.String" Nullable="true" sap:label="Reference Document ID" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CBTD_REF_ROLE_CODE" Type="Edm.String" Nullable="true" sap:label="Business Transaction Document Relationship Role" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CBTD_REF_TYPE_CODE" Type="Edm.String" Nullable="true" sap:label="Reference Type" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="CCAL_DAY" Type="Edm.DateTime" Nullable="true" sap:label="Calendar Day" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="COBA_DISTCHN_CD" Type="Edm.String" Nullable="true" sap:label="Distribution Channel" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="COBA_SALESGROUP" Type="Edm.String" Nullable="true" sap:label="Sales Group ID" sap:aggregation-role="dimension" sap:filterable="true" />
        <Property Name="COBA_SALESGROUP_UUID" Type="Edm.String" Nullable="true" sap:label="Sales Group" sap:aggregation-role="dimension" sap:filterable="true" sap:filter-restriction="multi-value" />
      </EntityType>
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>
```

Fig 6: Expanded view of metadata of DS in XWL format

2.3 Function Module for Data Source List

The module creates a list of 10 DS in EDM format for a WC. The number is restricted to 10 to avoid the dump caused due to fetching of large number of records.

```
<app:service xmlns:app="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom" xml:base="https://a6p-cust206.dev.sapbydesign.com/sap/byd/odata/cc_home_mdav_list_analytics.svc/">
  <app:workspace xmlns:sap="http://www.sap.com/Protocols/SAPData" sap:semantics="things">
    <atom:title> Things </atom:title>
  </app:workspace>

  <app:workspace xmlns:sap="http://www.sap.com/Protocols/SAPData" sap:semantics="data">
    <atom:title> Data </atom:title>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Service Category" href="DSSCCATSSICATQueryResults">
      <atom:title> DSSCCATSSICATQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Service Category Catalog" href="DSSCCATSSICTQueryResults">
      <atom:title> DSSCCATSSICTQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Incidents Analysis" href="DSSITSAMSAPSRBQueryResults">
      <atom:title> DSSITSAMSAPSRBQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="SAP Service Delivery and Service Tasks" href="DSSITSAMSVCBQueryResults">
      <atom:title> DSSITSAMSVCBQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Cost Center Master Data (obsolete)" href="DSSNOMSCOSTCTQueryResults">
      <atom:title> DSSNOMSCOSTCTQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Job Master Data" href="DSSNOMsJOBQueryResults">
      <atom:title> DSSNOMsJOBQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Business Residence Master Data (obsolete)" href="DSSNOMSPERMSTQueryResults">
      <atom:title> DSSNOMSPERMSTQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Account Master Data for Target Group" href="DSCCOCUSTOMER_TGQueryResults">
      <atom:title> DSCCOCUSTOMER_TGQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Project KPIs" href="DSCCOPS_CPM_PRJ_KPIQueryResults">
      <atom:title> DSCCOPS_CPM_PRJ_KPIQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Project Info for Financial Data" href="DSCCOPS_OPP_PRJ_CUST_KPIQueryResults">
      <atom:title> DSCCOPS_OPP_PRJ_CUST_KPIQueryResults </atom:title>
    </app:collection>
  </app:workspace>
</app:service>
```

Fig 7: Initial page when DS list loads with 10 records

```
<app:service xmlns:app="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom" xml:base="https://a6p-cust206.dev.sapbydesign.com/sap/byd/odata/cc_home_mdav_list_analytics.svc/">
  <app:workspace xmlns:sap="http://www.sap.com/Protocols/SAPData" sap:semantics="things">
    <atom:title> Things </atom:title>
  </app:workspace>

  <app:workspace xmlns:sap="http://www.sap.com/Protocols/SAPData" sap:semantics="data">
    <atom:title> Data </atom:title>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Service Category" href="DSSCCATSSICATQueryResults">
      <atom:title> DSSCCATSSICATQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Service Category Catalog" href="DSSCCATSSICTQueryResults">
      <atom:title> DSSCCATSSICTQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Incidents Analysis" href="DSSITSAMSAPSRBQueryResults">
      <atom:title> DSSITSAMSAPSRBQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="SAP Service Delivery and Service Tasks" href="DSSITSAMSVCBQueryResults">
      <atom:title> DSSITSAMSVCBQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Cost Center Master Data (obsolete)" href="DSSNOMSCOSTCTQueryResults">
      <atom:title> DSSNOMSCOSTCTQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Job Master Data" href="DSSNOMsJOBQueryResults">
      <atom:title> DSSNOMsJOBQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Business Residence Master Data (obsolete)" href="DSSNOMSPERMSTQueryResults">
      <atom:title> DSSNOMSPERMSTQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Account Master Data for Target Group" href="DSCCOCUSTOMER_TGQueryResults">
      <atom:title> DSCCOCUSTOMER_TGQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Project KPIs" href="DSCCOPS_CPM_PRJ_KPIQueryResults">
      <atom:title> DSCCOPS_CPM_PRJ_KPIQueryResults </atom:title>
    </app:collection>

    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:requires-filter="false" sap:label="Project Info for Financial Data" href="DSCCOPS_OPP_PRJ_CUST_KPIQueryResults">
      <atom:title> DSCCOPS_OPP_PRJ_CUST_KPIQueryResults </atom:title>
    </app:collection>
  </app:workspace>
</app:service>
```

Fig 7: Expanded view of ODATA for DS list

Properties like creatable, updatable, deletable, filter, label are shown in the EDM format.

Chapter 3

Integration of Function Modules to fetch data in EDM format

After creation of the Function Modules, they have to be integrated with methods, which builds EDM for the ODATA created. The ODATA created is exposed to the ODATA framework's team for further operations.

Above screenshots depict the ODATA in EDM format.

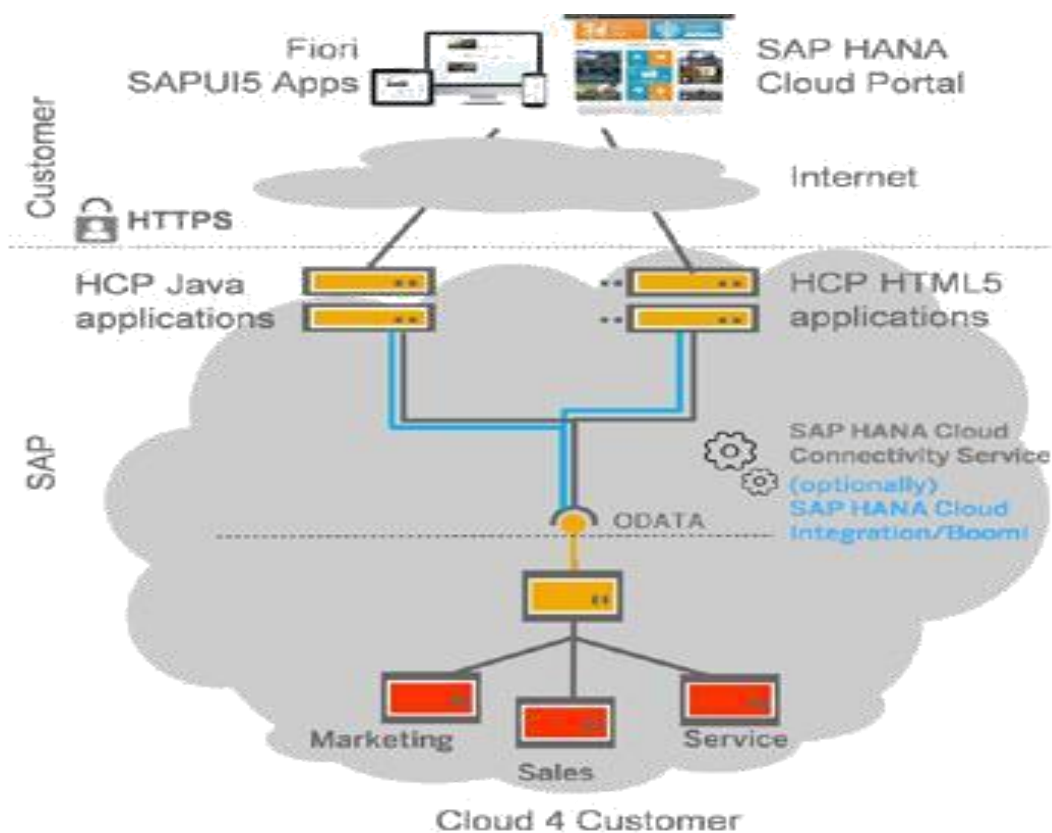


Fig 9: ODATA in SAP Hybris Cloud for customer

Conclusion:

The project would help customers to consume the metadata and list of data sources easily. The ODATA can be consumed in JAVA applications, HTML5 applications etc due to its easy consumption nature.

The main theme behind OData protocol is to provide REST based protocol to Create, Read, Update and Delete style operations. The specifications of OData define the standardization of typed, CRUD interface for manipulation of data. This is achieved by providing the collection of entries. This facilitates the ease of getting, updating and removing of data entries.

The main theme behind OData protocol is to provide REST based protocol to Create, Read, Update and Delete style operations. The specifications of OData define the standardization of typed, CRUD interface for manipulation of data. This is achieved by providing the collection of entries. This facilitates the ease of getting, updating and removing of data entries.

Directions for future work:

The ODATA implementation for other frameworks like KPIs, Dashboards, etc can be done in future.

References

1. Odata and netweaver gateway, Carsten Bonnen, 2013.
2. <https://www.google.co.in/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=ODATA+advantages>
3. <https://help.sap.com/cloud4customer>
4. <https://www.tutorialspoint.com/listtutorial/Understanding-OData-in-simplest-way/5948>
5. <http://www.topwcftutorials.net/2015/11/understanding-odata.html>