

# OData Protocol Guide

Last Modified on 09/01/2022 5:04 am EDT

## Overview

This article enables developers to use the OData protocol to access data from client technologies. OData support is available from Cora SeQuence V7.7 and later.

## Prerequisites

- Knowledge of product Data Model and Forms.
- Familiarity with the OData Service.

## OData service overview

OData is a standardized protocol for creating and consuming data APIs. OData builds on core protocols like HTTP and commonly accepted methodologies like REST. The result is a uniform way to expose full-featured data APIs.

OData has the following characteristics:

- Data access over RESTful web service or Atom feeds
- Built-in URI-based query syntax
- Supports client-side libraries for .NET (DataSvcUtil.exe), AJAX, Silverlight, and can be consumed from any external location.

## OData in Cora SeQuence

The Form Data Model can now be exposed as the OData service.

Cora SeQuence includes an implementation of a Representational State Transfer (REST) web service that uses the OData (version 2.0) protocol to perform CRUD operations on the Forms data, or to evaluate Stored Procedures and Service queries. You can use this to access Sequence data from client technologies.

We implement the OData specification in the following ways:

- Based on the OData v2 specification (<http://www.odata.org>)
- Built on top of the WCF Data Services v3.5 ([http://msdn.microsoft.com/enus/library/cc668810\(v=vs.90\).aspx](http://msdn.microsoft.com/enus/library/cc668810(v=vs.90).aspx))

Supported functionality:

- Table queries support CRUD operations including batches.
- Lookup queries are exposed as read only entity sets.
- Service and SP Queries are exposed as operations.

The exposed OData Service allows accessing the metadata based on the underlying data model.

For example:

```

<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
- <edmx:Edmx Version = "1.0" xmlns:edmx="http://schemas.microsoft.com/ado/2022/06/edmx">
  - <edmx:DataServices xmlns:m="http://schemas.microsoft.com/ado/2022/08/dataservices/metadata" m:DataServiceVersion="1.0">
    - <Schema Namespace="wf.S77Features.S77Features" xmlns:d="http://schemas.microsoft.com/ado/2022/08/dataservices" xmlns:edmx="http://schemas.microsoft.com/ado/2022/05/edmx">
      - <EntityType Name="UACT1">
        - <Key>
          <PropertyRef Name="fldId" />
        </Key>
        <Property Name="fldId" Type="Edm.Int32" Nullable="false" />
        <Property Name="fldWfld" Type="Edm.Int32" Nullable="false" />
        <Property Name="fldActId" Type="Edm.Int32" Nullable="false" />
        <Property Name="fldAllId" Type="Edm.Int32" Nullable="true" />
        <Property Name="StringField" Type="Edm.String" Nullable="true" />
        <Property Name="Int32Field" Type="Edm.Int32" Nullable="true" />
        <Property Name="DateField" Type="Edm.DateTime" Nullable="true" />
        <Property Name="DateTimeField" Type="Edm.DateTime" Nullable="true" />
        <Property Name="TrueFalseField" Type="Edm.Boolean" Nullable="true" />
      </EntityType>
      + <EntityType Name="UACT2">
      + <EntityType Name="UACT3">
      + <EntityType Name="NWCategories">
      + <EntityType Name="NWCustomers">
      + <EntityType Name="NWProducts">
      + <EntityType Name="NWShippers">
      + <ComplexType Name="ProductByCategory">
      + <Association Name="UACT3_Customer">
      + <EntityContainer Name="ODataService" m:IsDefaultEntityContainer="true">
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>

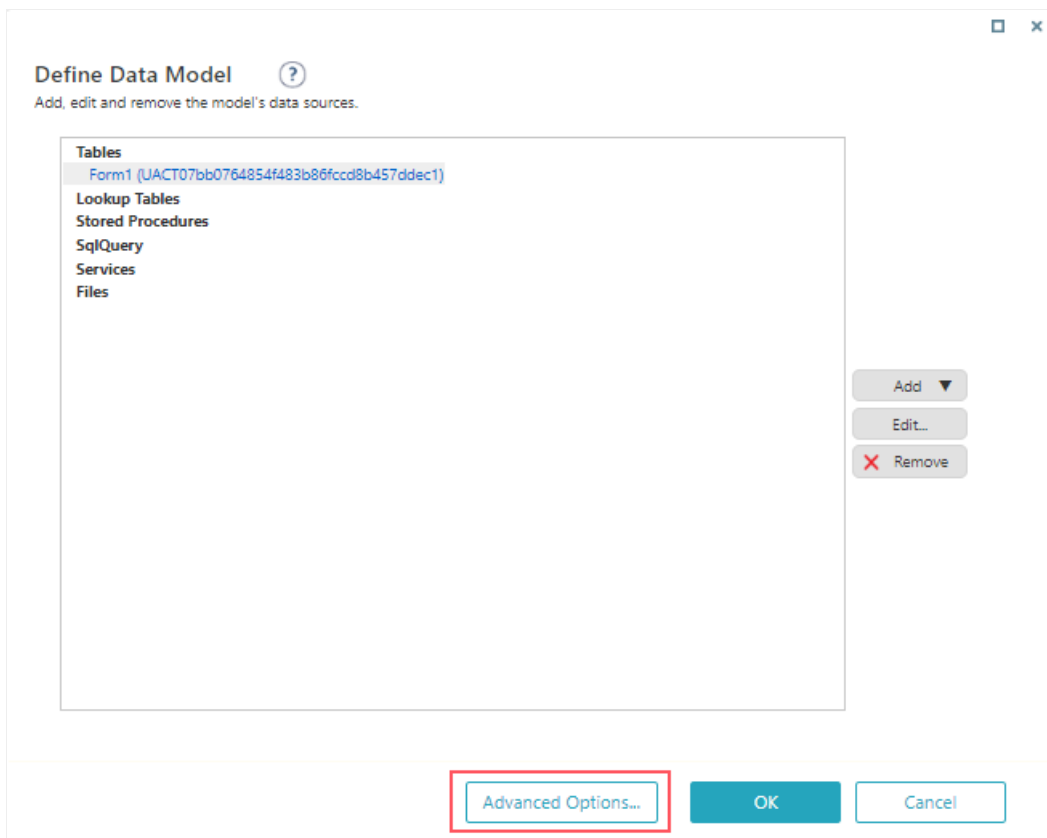
```

## Exposing a Form Data Model as an OData Service

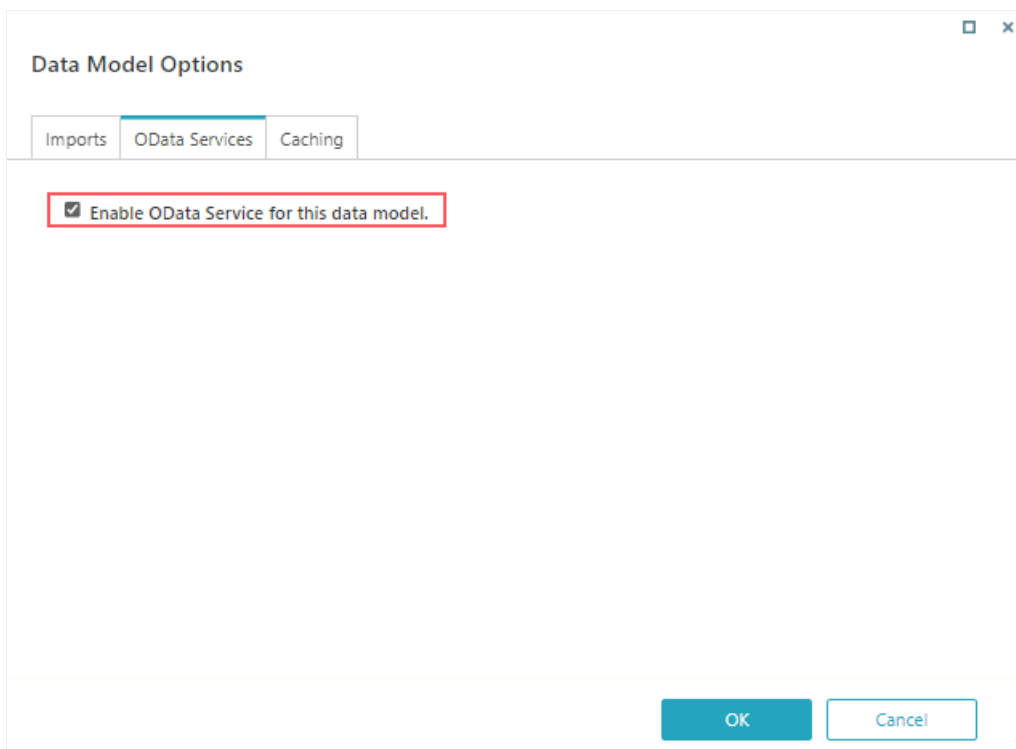
You must explicitly enable the OData Service endpoint using the Data Model wizard. (By default, the OData feature is disabled.)

To enable the OData Service:

1. From a Form activity, open the Data Model wizard.
2. Click **Advanced Options**.



3. From the OData Services tab, select **Enable OData Services for this data model**.



4. Click **OK**. The OData Service is enabled for this Form's Data Model.

### Endpoint format

The OData Service can be found using this endpoint format:

```
https://[sequence app]/sequenceservices/odata/activitydata.svc/wf/[workflow space]/[workflow | {active}]/[activity]/act([activity instance ID]) | msg([message instance ID])
```

You can use Message instance ID instead of Activity instance ID as msg(542).

Alternatively, in forms, you can use `sq:ODataEntityDataSource` control, that provides the client side API for accessing the OData Service endpoint address.

For example:

```
<sq:ODataEntityDataSource runat="server" ID="UserDetailsDataSource"
DataModelPath="UserDetails"></sq:ODataEntityDataSource>
<script type="text/javascript">
    function showUserDetails(userId, container) {
        var serviceUrl = $find("<%= { @container.ClientID } %>_UserDetailsDataSource").get_serviceUrl() + "tblE
mployees(" + userId + ")";
        $sq.getJSON(
            serviceUrl,
            function (data) {
                });
        }
    }
</script>
```

#### NOTE

To be exposed by the OData Service, a table-based query must have a non-nullable, primary key field defined.

## Querying the OData Service

You can query and filter the OData Service result using the following options:

Option	Description
orderby	Specifies how to order the entities.
skip	Skips number of data items.
top	At most returns top number of data items.
filter	Applies the filtering condition.
format	Specifies the response format, including JSONP.
sqsubmit	Specifies the form submit option. You can add the query string parameter to save/submit your form.

## Operators

You can use the following operators with OData:

Option	Description
eq	Equal
ne	Not equal
gt	Greater than
ge	Greater than or equal to
lt	Less than
le	Less than or equal to
and	Logical AND
or	Logical OR

For a complete operation reference page, see [this link](#).

#### Syntax examples

Action	Syntax
Fetch a product by the primary key	/Products(1)
Fetch top 10 most expensive Products	/Products?\$top=10&\$orderby=UnitPrice desc
Fetch Products by Category	/Products?\$filter=CategoryName eq 'Beverages'
Invoke SP Query	/ProductsByCategory?categoryId=1
Select only the ProductName and UnitPrice properties from the Products table	/Products\$select=ProductName,UnitPrice

#### Tool for Simplifying Constructing OData URLs

odataquerybuilder is a cross-browser JavaScript library that allows you to easily create OData queries and visualize the result(s).

For more information, see [this link](#).

#### Enabling OData Service after upgrading to a later version

To use the OData service after upgrading to a later version, copy this section to the web.config files:

```

<location path="SequenceServices/odata">
  <system.web>
    <httpHandlers>
      <remove verb="*" path=".svc" />
      <add verb="*" path="*.svc" validate="false" type="PNMsoft.Sequence.Data.Services.Activation.DataService
HttpHandler, PNMsoft.Sequence.Data.Services, Version=9.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dc
a1" />
    </httpHandlers>
  </system.web>
  <system.webServer>
    <handlers>
      <remove name="DataServiceHttpHandler" />
      <add name="DataServiceHttpHandler" verb="*" path="*.svc" type="PNMsoft.Sequence.Data.Services.Activ
ation.DataServiceHttpHandler, PNMsoft.Sequence.Data.Services, Version=9.0.0.0, Culture=neutral, PublicKeyToken
=0a1a1b90c1c5dca1" precondition="integratedMode" />
    </handlers>
  </system.webServer>
  <sequence.engine>
    <authentication impersonate="false"></authentication>
    <data.services>
      <handlers>
        <add type="PNMsoft.Sequence.Data.Services.EntityBoundActivityDataServiceRequestHandler, PNMsoft.Se
quence.Data.Services, Version=9.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1" />
      </handlers>
    </data.services>
  </web>
  <security>
    <xsrProtection enabled="true" />
  </security>
</web>
</sequence.engine>
</location>

```

## Troubleshooting tips

If you experience issues or errors with OData, try the following:

- Look for errors in the Event Viewer on the server where the service is consumed.
- Use a tool that enables tracing HTTP packets (an optional tool is [Telorik Fiddler](#)).

### NOTE

The product team doesn't guarantee this tool.

- Check the trace during the OData Service call.
- Read data going out from Sequence in JSON format.
- Check for HTTP response code (200-OK).
- Check for error messages.
- Check the response for incoming data.

Download a sample OData PoC, a [.json](#)  and a [package](#) , which includes the requests to update and push forward a Form and a Task.