

## OData and SOAP endpoint Microsoft Dynamics

You can use JavaScript web resources to access Microsoft Dynamics CRM 2015 and Microsoft Dynamics CRM Online 2015 Update data from within the application. There are two web services available, each provides specific strengths. The following table describes the appropriate

### Note

**In the previous version of CRM, the names “REST endpoint for web resources” and “SOAP endpoint for web resources” were used to describe the OData endpoint and the modern app SOAP endpoint, respectively. Because these endpoints are no longer only available for use in web resources, we have changed the names.**

web service to use depending on the task you need to perform.

| Task   | Web Service              |
|--|--------------------------|
| Create, Retrieve, Update and Delete records. | OData endpoint           |
| Associate and Disassociate records           | OData endpoint           |
| Assign Records                               | Modern app SOAP endpoint |
| Retrieve Metadata                            | Modern app SOAP endpoint |
| Execute Messages                             | Modern app SOAP endpoint |
|  |                          |

Both of these web services can use the authentication provided by the Microsoft Dynamics CRM application within web resources without the need to include any code to implement authentication.

The OData endpoint provides a “RESTful” web service using OData to provide a programming environment that is familiar to many developers. It is the recommended web service to use for tasks that involve creating, retrieving, updating and deleting records. However, in this release of Microsoft Dynamics CRM the capabilities of this web service are limited to these actions.

The modern app SOAP endpoint provides access to all the messages defined in the Organization service. However, only the types defined in the Web Services Description Language (WSDL) will be returned. There is no strong type support. While the modern app SOAP endpoint is also capable of performing create, retrieve, update and delete operations, the OData endpoint provides a better developer experience for client application extensions. In this release of Microsoft Dynamics CRM, the modern app SOAP endpoint provides an alternative way to perform operations that the OData endpoint isn’t capable of.

Use the OData endpoint with web resources

The OData endpoint with web resources provides an alternative interface to work with Microsoft Dynamics CRM 2015 and Microsoft Dynamics CRM Online 2015 Update data. You can use the

OData endpoint to execute HTTP requests by using a service that is based on a Uniform Resource Identifier (URI).

1. What is OData?
2. Microsoft Dynamics CRM implementation of REST
3. Third-party tools to generate code

### What is OData?

The OData endpoint uses the Open Data protocol. This protocol implements a “RESTful” design pattern. REST represents Representational State Transfer. REST is an architectural style in which every resource is addressed by using a unique URI. In Microsoft Dynamics CRM, a resource can be an entity collection or a record.

REST works the way the Internet works. You interact with resources by using HTTP verbs such as **GET**, **POST**, **MERGE**, and **DELETE**. Various libraries can be used to process the HTTP requests and responses. REST provides a standard interface that you can use with any programming language. REST allows for either synchronous or asynchronous processing of operations. The capability to perform asynchronous operations makes REST well suited for web resources and scripts used in Microsoft Dynamics CRM 2015 and Microsoft Dynamics CRM Online 2015 Update.

### Microsoft Dynamics CRM implementation of OData

Microsoft Dynamics CRM 2015 and Microsoft Dynamics CRM Online 2015 Update uses the Windows Communication Foundation (WCF) Data Services framework to provide an Open Data Protocol (OData) endpoint that is a REST-based data service. This endpoint is called the **Organization Data Service**. In Microsoft Dynamics CRM, the service root URI is:

### Copy

[Your Organization Root URL]/xrm/services/2011/organizationdata.svc

OData sends and receives data by using either ATOM or JavaScript Object Notation (JSON). ATOM is an XML-based format usually used for RSS feeds. JSON is a text format that allows for serialization of JavaScript objects.

To provide a consistent set of URIs that corresponds to the entities used in Microsoft Dynamics CRM, an Entity Data Model (EDM) organizes the data in the form of records of "entity types" and the associations between them.

### **OData Entity Data Model**

The Microsoft Dynamics CRM EDM is described in an OData Service Metadata document available at the following path:

## Copy

[Your Organization Root URL]/xrm/services/2011/organizationdata.svc/\$metadata

This XML document uses conceptual schema definition language (CSDL) to describe the available data. You will download this document and use it to generate typed classes when you use managed code or as a reference for available objects when you use JavaScript.

## Limitations

The OData endpoint provides an alternative to the SOAP endpoint, but there are currently some limitations.

**Only Create, Retrieve, Update, and Delete actions can be performed on entity records.**

- Messages that require the **Execute** method can't be performed.
- Associate and disassociate actions can be performed by using navigation properties.

**The OData protocol isn't fully implemented. Some system query options are not available.**

For more information, see [OData system query options using the OData endpoint](#).

**You can't use late binding with managed code against custom entities, attributes, or relationships that did not exist when the code was written.**

You will typically use WCF Data Services Client Data Service classes while programming by using managed code. These classes allow for early binding so that you get strongly typed classes at design time. The only entities available to you are those defined in the system when the classes were generated. This means that you can't use late binding to work with custom entities, attributes, or relationships that were not included in the WCF Data Services Client Data Service classes when they were generated.

## Third-party tools to generate code

The community of developers who work with Microsoft Dynamics CRM have created some tools that demonstrate how the OData endpoint works and can generate code you can use. The following tools provide capabilities to generate code to work with CRM data using the OData endpoint.

### CRM REST Builder

Use this tool to build requests to retrieve data and perform operations using a variety of JavaScript libraries.

Thanks to Microsoft Dynamics CRM MVP [Jason Lattimer](#) for building this tool.

## **OData Query Designer**

Included with the Dynamics XRM Tools project, the OData Query Designer allows you to create queries to retrieve data using the OData endpoint.

Use the Modern app SOAP endpoint for modern applications with web resources

Unlike the REST endpoint for web resources, the SOAP endpoint uses the Organization service. This is the same service used when writing applications that exist outside of the Microsoft Dynamics CRM 2015 and Microsoft Dynamics CRM Online 2015 Update application. The differences are:

- Requests are sent to a different URL: <organization URL>/XRMServices/2011/Organization.svc/web.
- Authentication for web resources is provided by the application.

If you use this endpoint outside the application, you must implement authentication. More information: [Authenticate the user with the web services](#)

### [Using the SOAP endpoint with JavaScript](#)

---

With JavaScript, you will be using **XmlHttpRequest** to **POST** requests to the service. The body of the request must contain the XML appropriate for the message you are using. You must also parse the XML returned in a response. You have several options for using the SOAP endpoint with JavaScript:

- Create a library yourself using the procedure described in [Walkthrough: Use the Modern app SOAP endpoint with JavaScript](#).
- Use the **Sdk.Soap.js** sample library described below.
- Use open source libraries.

### **Sample: Sdk.Soap.js library**

**Sdk.Soap.js** is a sample that demonstrates a JavaScript library that makes it easier to write code that uses the SOAP endpoint for modern applications with web resources. This library was created using the **SOAPLogger** sample application described in [Walkthrough: Use the Modern app SOAP endpoint with JavaScript](#) to capture and observe the XML data passed to the server. **Sdk.Soap.js** includes object definitions that correspond with classes and methods from the SDK assemblies. These include separate libraries for more than 200 messages you can use.

**Sdk.Soap.js** is for use with web resources only. It does not provide the capability to authenticate from outside the application.

**Sdk.Soap.js** provides a programming model that is based on how you write C# code using the assemblies included in the SDK. You can write code using the late bound or early bound styles. To use the early bound style you must generate JavaScript libraries representing classes for the entities you will use in your code. Use the [Sdk.Soap.js Entity Class Generator](#) to generate the libraries you need. No additional libraries are required to write code using the late-bound style. More information: [Entity programming \(early bound vs. late bound vs. developer extensions\)](#)

If you use actions to create custom messages, you can use the [Sdk.Soap.js Action Message Generator](#) to generate JavaScript libraries for each action in your organization and then use **Sdk.Soap.js** to call those actions in your JavaScript code. More information: [Create your own actions](#)

[Sdk.Soap.js Samples](#) is a separate set of samples that demonstrates using **Sdk.Soap.js** with both late bound and early bound styles, messages, and queries.