Microsoft Dynamics 365-REST API

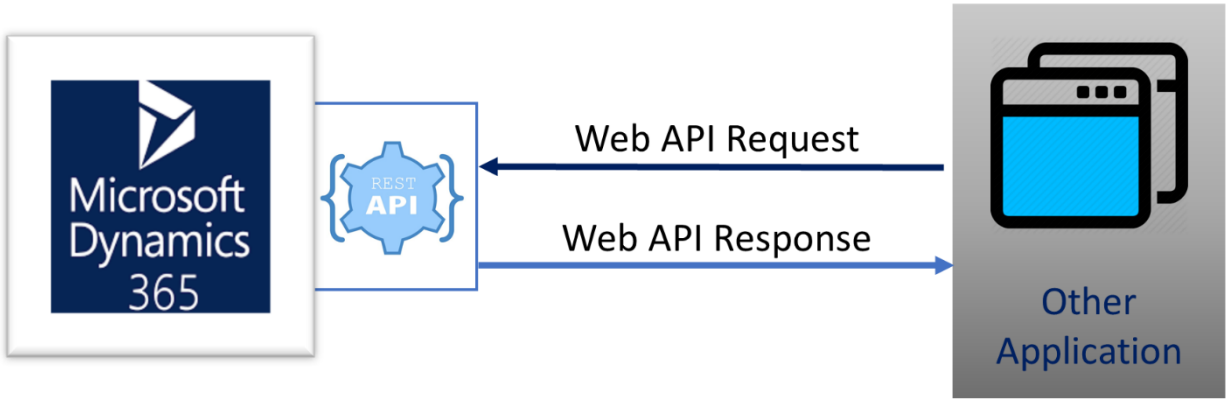
Cognitive Convergence
http://www.cognitiveconvergence.com
+1 4242530744
shahzad@cognitiveconvergence.com

Cognitive Convergence is Subject Matter Expert in Office 365, Dynamics 365, SharePoint, Project Server, Power Platform: Power Apps-Power BI-Power Automate-Power Virtual Agents. Our Microsoft Dynamics 365 Consulting, Development, Customization, Integration services and solutions, can help companies maximize business performance, overcoming market challenges, achieving profitability and providing best customer service.

Contents

OBJECTIVE

Dynamics 365 provides a vast possibility for Rest API and OData protocols. The Dynamics 365 Web API implements the OData (Open Data Protocol), version 4.0, an OASIS standard for building and consuming RESTful APIs over rich data sources, in this case, Dataverse. The REST API can be used with both cloud deployments and on-premises deployments.

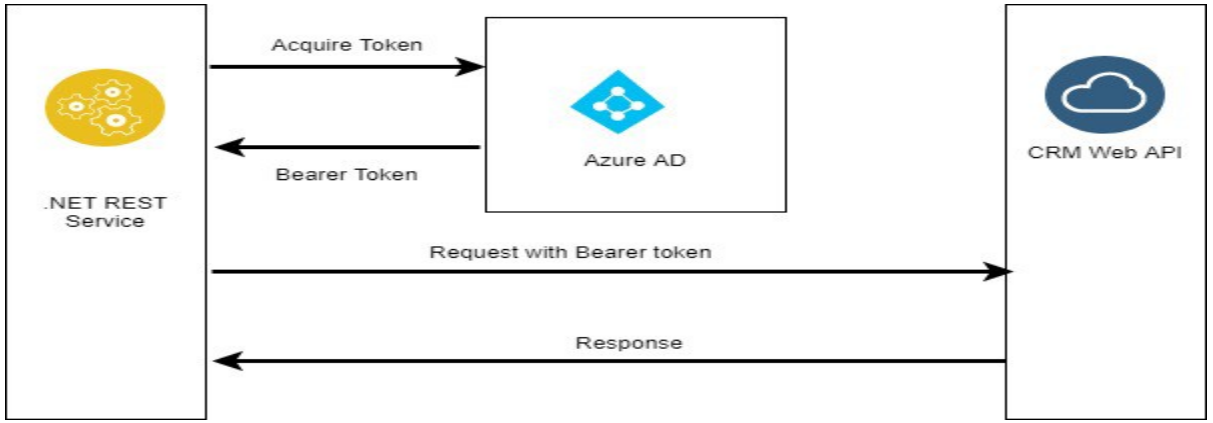


DYNAMICS 365 CUSTOMER ENGAGEMENT REST APIS

Dynamics 365 unifies the capabilities of CRM business software and ERP systems by providing intelligent applications that seamlessly work together in the cloud. With Dynamics 365, previous Dynamics CRM functionality is included as a part of a suite of intelligent business applications. The apps that make up the customer relationship management (CRM) portion of the suite are referred to as "Customer Engagement" and include Sales, Customer Service, Field Service, Project Service Automation, and their related services.

Customer Engagement provides the following REST APIs:

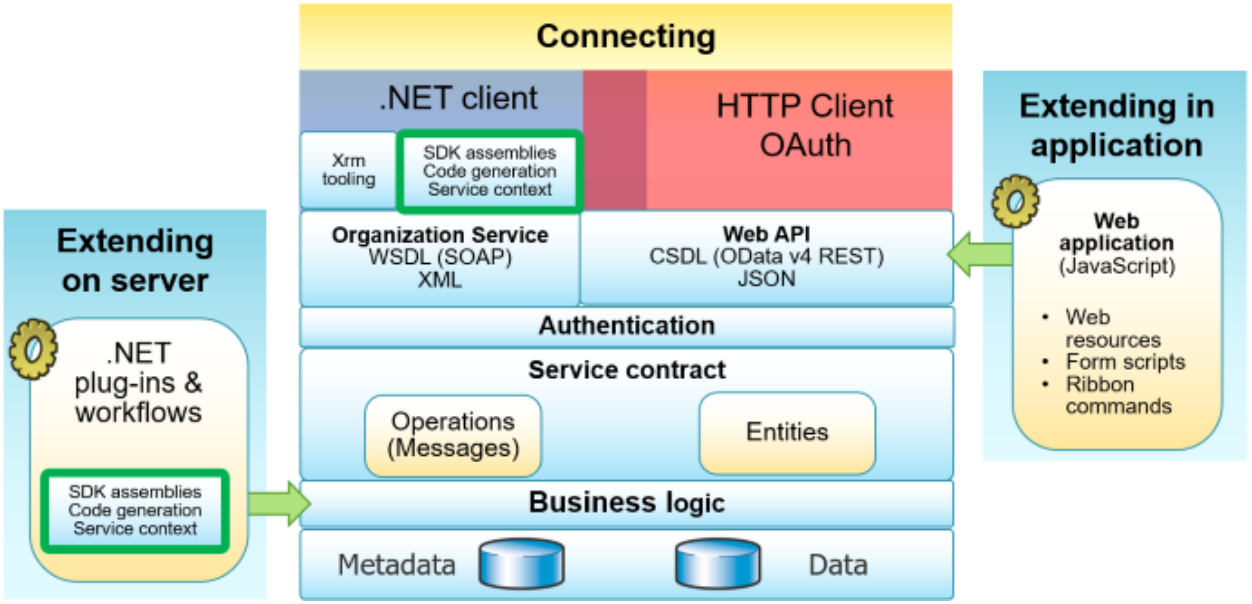
- Web API
- Online Management API (deprecated)



DYNAMICS 365 WEB SERVICES

Web services provide APIs that are used to write programs for Microsoft Dynamics 365 (online & on-premises).

Microsoft Dynamics 365 provides different Web services, which include the following:

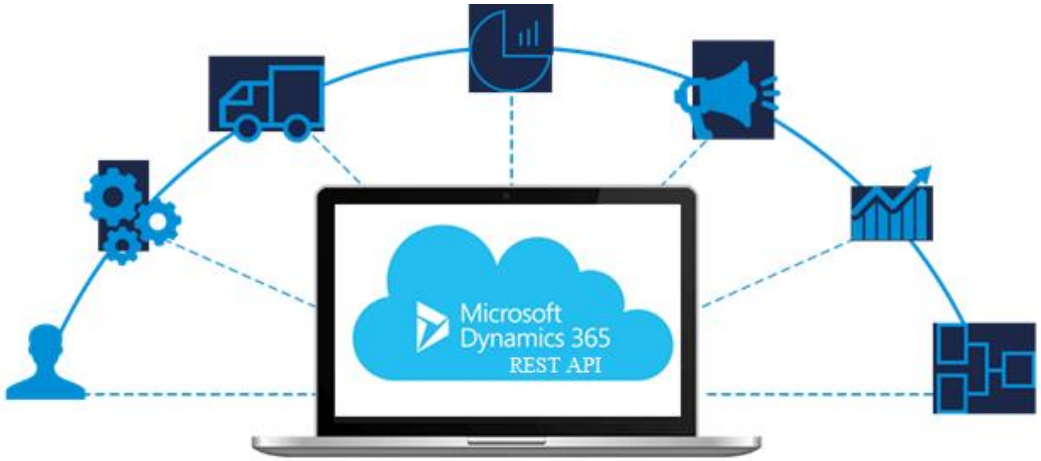


Web API

The Web API for Microsoft Dynamics 365 (online & on-premises), provides a development experience that can be used across a wide variety of programming languages, platforms, and devices.

As the Web API is built on open standards, Microsoft do not provide assemblies for specific developer experience. The option of composing HTTP requests for specific operations or third-party libraries to generate classes for whatever language or platform is being used by the developer, is available.

Due to these exceptional qualities, Web API will be going to replace the Organizational services and Organizational Data Services.



Organization Services

The Organization service, also sometimes known as the "SOAP endpoint," has been available since Microsoft Dynamics CRM 2011. It's the web service that most developers working with Microsoft Dynamics 365 are already familiar with. The Organization service is optimized for use with .NET. The Microsoft Dynamics 365 SDK provides a set of assemblies and tools to generate strongly typed classes and proxies that streamline the development process with a better development experience using Microsoft Visual Studio.

Plug-ins or workflow assemblies on the server use the Organization service. Input and output parameters use specific classes defined with the assemblies that support the Organization service.

Organization Data services

The Organization Data service, also sometimes known as the “OData” or “REST” endpoint has also been available since Microsoft Dynamics CRM 2011. This service implemented the OData v2 standard. These services are going to be deprecated. In the form of JavaScript, embedded in the web resources as client-side, OData actions can be performed easily. The CRUD implementation will be discussed in this paper.

Discovery Web Services

Discovery web services provide a way for a client to detect which servers and organizations are available for a user to connect to base on their user account. You can choose to use either the Web API Discovery service or the IDiscoveryService web service. The Web API Discovery service provides the same benefits as the Web API, it is easier to consume for a wider range of programming languages, platforms, and devices.

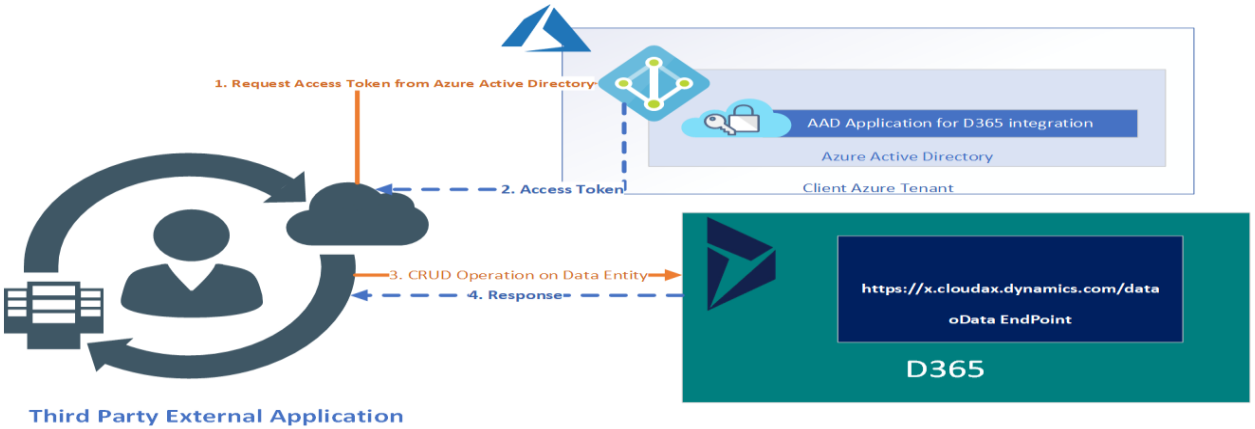
Deployment Web Services

For Dynamics 365 (on-premises) actions can be performed to manage your deployment programmatically using the Deployment web service. These are essentially the same operations that are performed on the server using the Deployment manager tool client installed on the server. Create, import, or delete organizations as well as apply certain settings in code can be used. This may be useful when you want to automate certain processes if you are providing a hosting service or if you want to automate creation of environments for testing.

OPEN DATA PROTOCOL (ODATA)

OData is a standard protocol for creating and consuming data. The purpose of OData is to provide a protocol that is based on Representational State Transfer (REST) for create, read, update, and delete (CRUD) operations. OData applies web technologies such as HTTP and JavaScript Object Notation (JSON) to provide access to information from various programs.

The public OData service endpoint enables access to data in a consistent manner across a broad range of clients.



Addressing

The following table describes the resources and the corresponding URLs in the Fleet Management sample.

Resource	URL	Description
Service endpoint	[Your organization's root URL]/data/	The root service endpoint for OData entities
Entity collection	[Your organization's root URL]/data/Customers	The collection of all customers
Entity	[Your organization's root URL]/data/Customers("[key]")	A single entity from the entity collection
Navigation property	[Your organization's root URL]/data/Customers("[key]")/Reservations	The navigation from a customer to that customer's reservations
Property	[Your organization's root URL]/data/Customers("[key]")/FirstName	The customer's first name

Supported features from the OData specification

The following are the high-level features that are enabled for the OData service, per the OData specification.

- CRUD support is handled through HTTP verb support for POST, PATCH, PUT, and DELETE.
- Available query options are:
    - $filter
    - $count
    - $orderby
    - $skip
    - $top
    - $expand (only first-level expansion is supported)
    - $select
- The OData service supports serving driven paging with a maximum page size of 10,000.

Filter details

There are built-in operators for $filter:

- Equals (eq)
- Not equals (ne)
- Greater than (gt)
- Greater than or equal (ge)
- Less than (lt)
- Less than or equal (le)
- And
- Or
- Not
- Addition (add)
- Subtraction (sub)
- Multiplication (mul)
- Division (div)
- Decimal division (divby)
- Modulo (mod)

- Precedence grouping ({ })

For example,

```
http://host/service/EntitySet?$filter=StringField eq '\*retail\*'
```

```
http://[baseURI]/data/PurchaseOrderHeadersV2?$filter=dataAreaId eq
'usrt'&cross-company=true
```


Validate methods

The following table summarizes the validate methods that the OData stack calls implicitly on the corresponding data entity.

OData	Methods (listed in the order in which they are called)
Create	• Clear() • Initvalue() • PropertyInfo.SetValue() • Validate Field() • Defaulltrow() • Validatewrite() • Write()
Update	• Forupdate() • Reread() • Clear() • Initvalue() • PropertyInfo.SetValue() for all specified fields in the request • Validatefield() • Defaultrow() • Validatewrite() • Write()
Delete	• Forupdate()

	• Reread() • checkRestrictedDeleteActions() • validatedelete() • Delete()

Examples of querying on OData Endpoint

List all the customers:

[Your organization's root URL]/data/Customers

List first 3 records:

[Your organization's root URL]/data/Customers?$top=3

List all the customers, but show only the first name and last name properties:

[Your organization's root URL]/data/Customers?$select=FirstName,LastName

List all the customers in a JSON format that can be used to interact with JavaScript clients:

[Your organization's root URL]/data/Customers?$format=json

HTTP METHODS

We interact with the Web API by composing and sending HTTP requests. We need to set the appropriate HTTP headers and handle any errors included in the response. HTTP requests can apply a variety of different methods.

When using the web API you will only use the methods listed in the following table.

Method	Usage
GET	Use when retrieving data, including calling functions. The expected Status Code for a successful retrieve is 200 OK.
POST	Use when creating entities or calling actions.
PATCH	Use when updating entities or performing upsert operations.
DELETE	Use when deleting entities or individual properties of entities.

PUT	Use in limited situations to update individual properties of entities. This method isn't recommended when updating most entities. You'll use this when updating model entities.

HTTP HEADERS

Every request should include the Acceptheader value of application/json, even when no response body is expected. Any error returned in the response will be returned as JSON. While the code should work even if this header isn't included, it is recommended to include it as a best practice.

```
Accept: application/json OData-MaxVersion: 4.0 OData-Version: 4.0
```

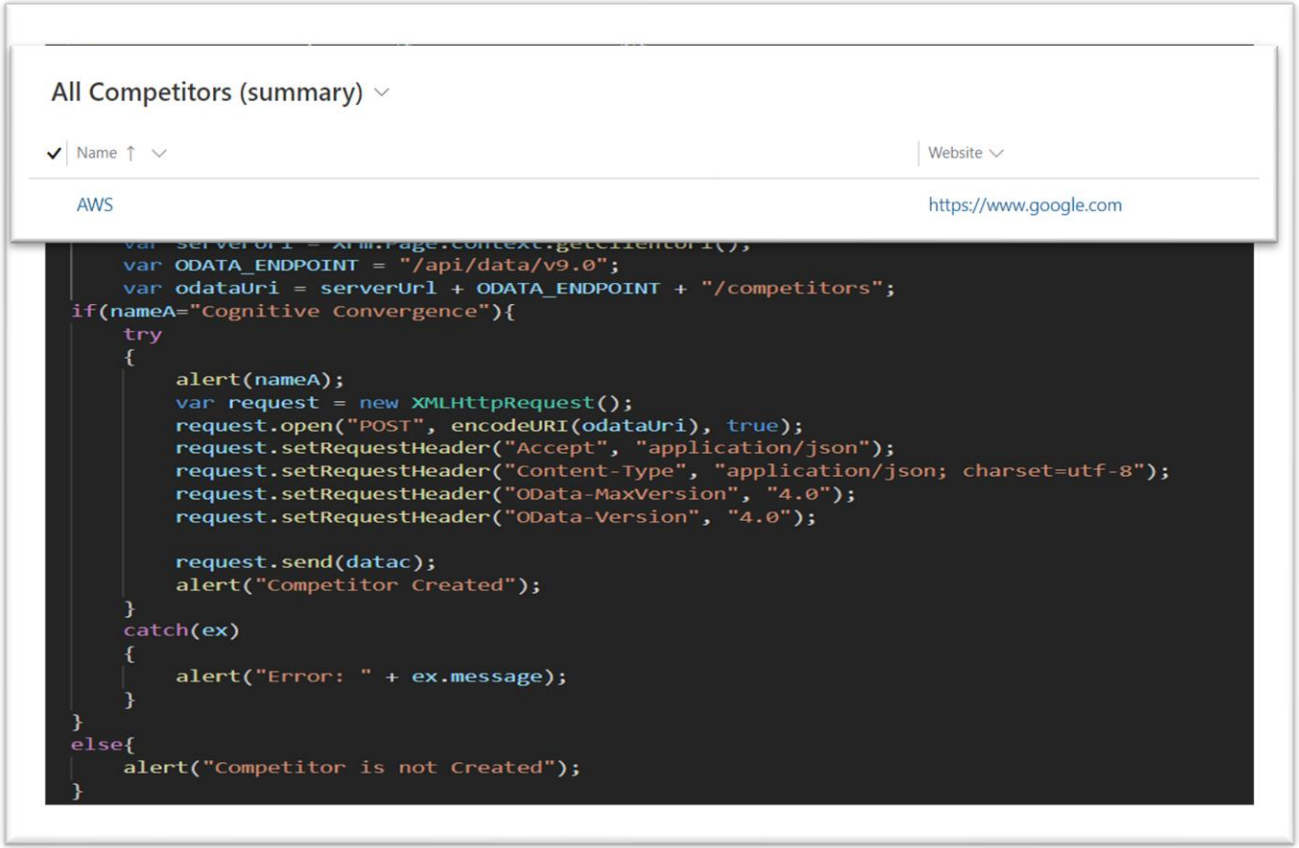
```
Content-Type: application/json
```

REST API CRUD OPERATIONS

Cognitive Convergence is a subject matter expert in Dynamics 365 REST API and its functions. A sample implementation of CRUD Operations is demonstrated.

Create Record (POST request)

A use-case identified for creating a competitor against an account, in Sales module. Whenever an account named “Cognitive Convergence” is added, a competitor against it, in the competitor entity is created with alerts.



A competitor is created against Cognitive Convergence.

All Competitors (summary)

Name	Website
AWS	https://www.google.com

Update Record (PATCH request)

Telephone numbers have some prefixes for different cities.

```
function updateAddress(executionContext)
{
var formContext = executionContext.getFormContext();
var address1=formContext.getAttribute("address1_city").getValue();
var datac;
if(address1=="Lahore"){
        var jsonstr={
    "name": "Updated through rest API ",
    "telephone1" : "0423"
};
        datac=JSON.stringify(jsonstr);
}
else if(address1=="Karachi"){
            var datac=JSON.stringify({
    "name": "Updated through rest API ",
    "telephone1" : "0213"
});

}

else if(address1=="Faisalabad"){
            var datac=JSON.stringify({
    "name": "Updated through rest API ",
    "telephone1" : "0513"
});

}
```

Update through PATCH:
Updated record shows:

```
var serverUrl = Xrm.Page.context.getClientUrl();
    alert(serverUrl);
    var ODATA_ENDPOINT = "/api/data/v9.0";
    var odataUri = serverUrl + ODATA_ENDPOINT + "/accounts(dd7e5408-e763-ec11-8f8f-000d3a320d63)";
        alert(odataUri);
        alert(datac);

        try
        {
            alert(address1);
            var request = new XMLHttpRequest();
        request.open("PATCH", encodeURI(odataUri), true);
        request.setRequestHeader("Accept", "application/json");
        request.setRequestHeader("Content-Type", "application/json; charset=utf-8");
        request.setRequestHeader("OData-MaxVersion", "4.0");
        request.setRequestHeader("OData-Version", "4.0");
        //request.setRequestHeader("Prefer", "return=representation" );

        request.send(datac);

        var resp = JSON.parse(request.responseText);
        alert(resp);

            alert(address1);
        }
        catch(ex)
        {
            alert("Error: " + ex.message);
        }
```

Updated through rest API	0213	---

Delete Record (DELETE REQUEST)

If there is some change in the record named "Cognitive Convergence", its competitor is also removed.

```
function deleteRecordRest(executionContext) {
    var serverUrl = Xrm.Page.context.getClientUrl();
    var ODATA_ENDPOINT = "/api/data/v9.0";
    var odataUri = serverUrl + ODATA_ENDPOINT + "/competitors(f4f9136a-f063-ec11-8f8f-000d3a320d63)";

    var formContext = executionContext.getFormContext();

    formContext.getAttribute("name").addOnChange(function () {
        if (formContext.getAttribute("name").getValue() != "Cognitive Convergence") {
            try {
                alert(formContext.getAttribute("name").getValue());
                var request = new XMLHttpRequest();
                request.open("DELETE", encodeURI(odataUri), false);
                request.setRequestHeader("Accept", "application/json");
                request.setRequestHeader("Content-Type", "application/json; charset=utf-8");
                //request.setRequestHeader("Prefer", "return=representation" );

                request.send();
                alert("Competitor deleted");

            }
            catch (ex) {
                alert("Error: " + ex.message);
            }
        }
    });
}
```

Read Record (GET request)

```
function getAccountData(accountId)
{
    var serverUrl = Xrm.Page.context.getClientUrl();
    var ODATA_ENDPOINT = "/api/data/v9.0";
    var odataUri = serverUrl + ODATA_ENDPOINT + "/accounts?";
    if(accountId)
    {
        odataUri += "$filter=accountid eq '"+ accountId +"'";
    }

    var data;
    try
    {
        var request = new XMLHttpRequest();
        request.open("GET", encodeURI(odataUri), false);
        request.setRequestHeader("Accept", "application/json");
        request.setRequestHeader("Content-Type", "application/json; charset=utf-8");
        request.send(null);

        data = JSON.parse(request.responseText);
        console.log("The data is:");
        console.log(data);
    }
    catch(ex)
    {
        alert("Error: " + ex.message);
    }
}
```

MICROSOFT DYNAMICS 365 WEB API LIMITATIONS

Here is the list of limitations that comes with Microsoft Dynamics 365 Web API.

1. Some custom actions not available in Web API
2. Missing functions and actions for some organization service messages
    a. GrantAccessRequest

   b. ModifyAccessRequest
   c. RetrieveEntityKeyRequest
   d. RemoveMemberListRequest
   e. RemoveItemCampaignRequest
   f. RemoveItemCampaignActivityrequest
3. Can't query the date values.
4. When calling actions with entity collection parameters as well as other parameters, a collection parameter must be passed as the last parameter in the body.
5. Can't create a customer lookup attribute.
6. Can't retrieve unpublished metadata.
7. Web API not enabled for Microsoft Dynamics 365 for Outlook with Offline Access while the user is offline.
8. When calling actions with entity collection parameters as well as other parameters, a collection parameter must be passed as the last parameter in the body.
9. Can't filter queries based on the value of a single-valued navigation property.
10. Error when querying self-referential many-to-many relationships.
11. $select on some $expand expressions may be ignored.
12. Single-valued navigation properties may not be returned from a $expand query if they have a null value.
13. FetchXML queries linked to activitypointer may not include linked fields.
14. Null-valued properties may not be returned in expanded navigation property results.
15. Error when querying self-referential many-to-many relationships.

PERFORM OPERATIONS USING THE WEB API

The Web API is new for Microsoft Dynamics 365 (online & on-premises). It provides a modern, RESTful web service you can use to interact with data in Microsoft Dynamics 365 using a wide variety of platforms, programming languages, and devices. Here is the list of operations that can be performed using Web API:

1. Compose HTTP requests and handle errors
2. Query Data using the Web API
3. Retrieve and execute predefined queries
4. Create an entity using the Web API
5. Retrieve an entity using the Web API
6. Update and delete entities using the Web API
7. Associate and disassociate entities using the Web API
8. Use Web API functions
9. Use Web API actions
10. Execute batch operations using the Web API
11. Impersonate another user using the Web API
12. Perform conditional operations using the Web API

REST OPERATIONS: KEYS

Operation	Description
Generate Key	Generates tenant protection (encryption) key.
Import Key	Imports tenant protection (encryption) key.
Retrieve Available Tenant Protection Keys	Retrieves available tenant protection (encryption) keys.
Retrieve Tenant Protection Key	Retrieves the current tenant protection (encryption) key.
Set Tenant Protection Key	Sets the tenant protection (encryption) key.

REST OPERATIONS: NOTIFICATION

Operation	Description
Get All Notifications	Retrieves all notifications from the Notification service.
Post User Notification	Post user notifications to the Notification service.

REST OPERATIONS: TENANT APPLICATION IDENTITY

Operation	Description
Create Tenant Application Identity	Register a tenant application identity.
Delete Tenant Application Identity	Deletes a tenant application identity.
Disable Tenant Application Identity	Disables a tenant application identity.
Enable Tenant Application Identity	Enables a tenant application identity.
Get Tenant Application Identities	Retrieves all tenant application identities.
Get Tenant Application Identity	Retrieves a tenant application identity.

CONCLUSION

In this case study, a brief introduction was discussed about the Dynamics 365 Rest API, terminologies that can be used during development, Dynamics 365 Web Services, OData Protocols, HTTP requests along with the limitations of Dynamics 365 Web API. CRUD implementation using JavaScript through REST is also discussed.

Our Microsoft Dynamics 365 Consulting, Development, Customization, Integration services and solutions, can help companies maximize business performance, overcoming market challenges, achieving profitability, and providing the best customer care service.

Contact Us

Cognitive Convergence
http://www.cognitiveconvergence.com
+1 4242530744
shahzad@cognitiveconvergence.com