

Microsoft Dynamics 365-XRM CRUD OPERATION

**CRUD
WITH**

 Microsoft | Dynamics 365

XRM TOOLING

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.

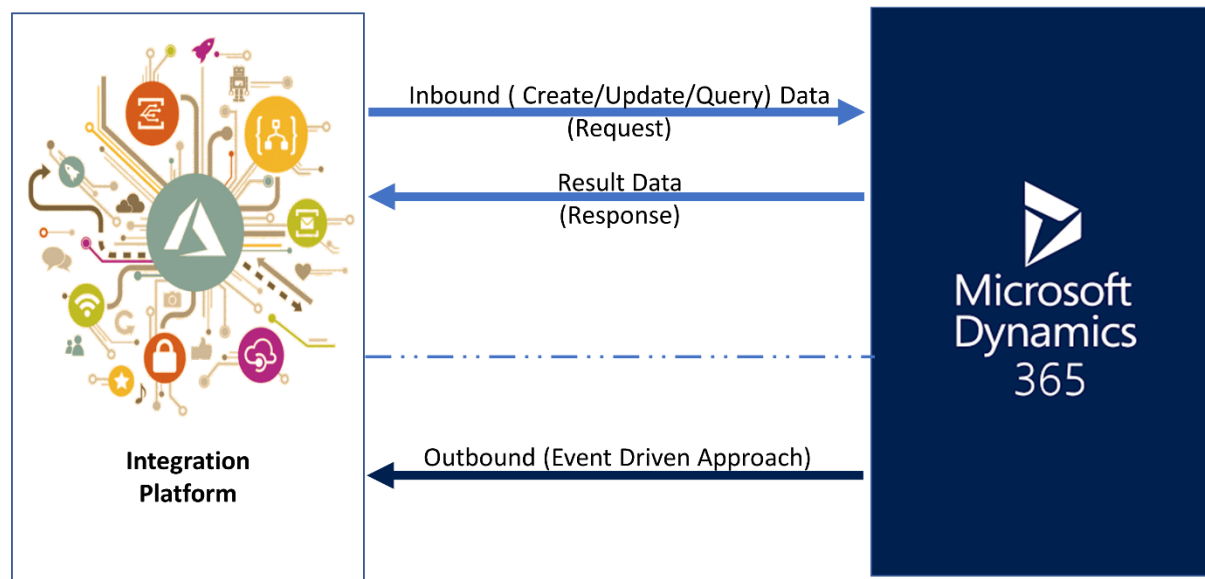
Contents

OBJECTIVE	3
CLIENT API XRM OBJECT MODEL	3
Xrm.Device	5
Xrm.Encoding	5
Xrm.Navigation	6
Xrm.Panel	6
Xrm.Utility	7
Xrm.WebAPI	7
ONLINE AND OFFLINE WEB API.....	8
Online Web API	8
Offline Web API	9
Dynamics 365 XRM Web API for CRM and ERP	9
XRM.WEBAPI.....	11
D365 Web API and Azure	11
Methods	12
CREATERECORD (CLIENT API REFERENCE)	12
Syntax	12
Parameters	13
Return Value	13
Sample Implementation	14
DELETERECORD (CLIENT API REFERENCE).....	14
Syntax	14
Parameters	15
Return Value	15
RETRIEVERECORD (CLIENT API REFERENCE)	16

Syntax	16
Parameters	16
Return Value	17
Sample Implementation	17
UPDATERECORD (CLIENT API REFERENCE)	18
Syntax	18
Parameters	18
Return Value	18
Sample Implementation	19
RETRIEVEMULTIPLERECORDS (CLIENT API REFERENCE)	19
Syntax	19
Parameters	20
Return Value	20
Sample Implementation	20
ISAVAILABLEOFFLINE (CLIENT API REFERENCE)	21
Syntax	21
Return Value	21
CONCLUSION	22

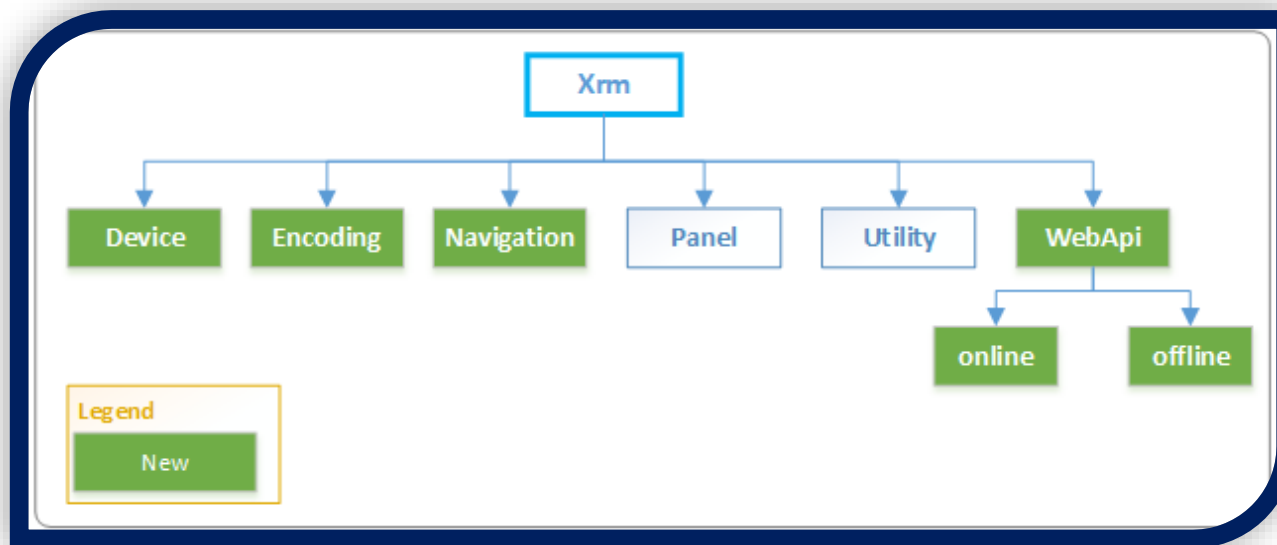
OBJECTIVE

Dynamics 365 XRM Web API provides properties and methods to use web API to create and manage dataverse records and execute Web API actions and functions in model-driven apps. It provides the functionality of CRUD operations along with a few more functionalities. The available functionalities will be discussed in this case study.



CLIENT API XRM OBJECT MODEL

The Xrm object is globally available to use in your code without having to use the execution context in Client API. It provides a global object for performing operations that do not directly impact the data and UI in forms, grids, subgrids, controls, or attributes. For example, navigate forms, create and manage records using Web API.



The following illustration displays the Xrm object model:

Xrm.Device			Xrm.Encoding			Xrm.WebApi		
Method	Parameters	Description	Method	Parameters	Description	Property	Parameters	Description
captureAudio	successCallback, errorCallback	Invokes the device microphone to radio.	htmlAttributeEncode	arg(string to be encoded)	Encodes the specified string so that it can be used in an HTML attribute.	online		Provides methods to use Web API to create and manage records and execute Web API actions and functions in model-driven apps when connected to the model-driven apps server(online mode).
captureImage	imageOptions, successCallback, errorCallback	Invokes the device camera to capture an image.	htmlDecode	arg(HTML-encoded string to be decoded)	Converts a string that has been HTML-encoded into decoded string.	offline		Provides methods to create and manage records in the model-driven apps mobile client while working in the offline mode.
captureVideo	successCallback, errorCallback	Invokes the device camera to record video.	htmlEncode	arg(string to be encoded)	Converts a string to an HTML-encoded string.	Methods	Parameters	Description
getBarcodeValue	successCallback, errorCallback	Invokes the device camera to scan the barcode information, such as product number.	xmlAttributeEncode	arg(string to be encoded)	Encodes the specified string so that it can be used in XML attribute.			
getCurrentPosition	successCallback, errorCallback	Returns the current location using the device geolocation capability.	xmlDecode	arg(string to be encoded)	Converts string to an XML-encoded string.	createRecord	entityLogicalName, data, successCallback, errorCallback	Creates an entity record.
pickFile	pickFileOptions, successCallback, errorCallback	Opens a dialog box to select files from computer (web client) or mobile device (mobile clients).				deleteRecord	entityLogicalName, id, successCallback, errorCallback	Deletes an entity record.
Xrm.Navigation			Xrm.Utility			retrieveRecord	entityLogicalName, id, options, successCallback, errorCallback	Retrieves an entity records.
Method	Parameters	Description	Method	Parameters	Description	retrieveRecords	entityLogicalName, options, mapAgg, successCallback, errorCallback	Retrieves a collection of entity records.
openAlertDialog	alertStrings, alertOptions, successCallback, errorCallback	Display an alert dialog containing a message and a button.	closeProgressIndicator		Closes a progress dialog box.	updateRecord	entityLogicalName, id, data, successCallback, errorCallback	Updates an entity record.
openConfirmDialog	confirmStrings, confirmOptions, successCallback, errorCallback	Displays a confirmation dialog box containing a message and a two button.	getAllowedStatusTransitions	entityName, stateCode, successCallback, errorCallback	Returns the valid state transitions for the specified entity type and state code.	isAvailableOffline	entityLogicalName	Returns a boolean value indicating whether an entity is present in user's profile and is currently available for use in offline mode.
openErrorDialog	errorOptions, successCallback, errorCallback	Displays an error dialog.	getEntityMetadata	entityName, attributes, successCallback, errorCallback	Returns the entity metadata for the specified entity.	execute	request, successCallback, errorCallback	Execute a single action, function or CRUD operation.
openFile	file, openFileOptions	Opens a file.	getGlobalContext		Gets the global context.	executeMultiple	requests, successCallback, errorCallback	Execute a collection of action, function or CRUD operation.
openForm	entityFormOptions, formParameters, successCallback, errorCallback	Opens an entity form or a quick create form.	getNamingWebAttribute		Returns the name of the DOM attribute by the Learning Path (guided help) Content Designer for identifying UI controls in the model-driven apps forms.			
openUrl	url, openUrlOptions	Opens a URL including file URLs.	getResourceString	webResourceName, key	Returns the localized string for a given key associated with the specified web resource.			
openWebResource	webResourceName, windowOptions, data	Opens an HTML web resource.	invokeProcessAction	name, parameters, successCallback, errorCallback	Invokes an action based on the specified parameters.			
Xrm.Panel			lookupObjects	allowMultiSelect, defaultEntityType, defaultViewId, disabledIds, entityType, filter, showBarcodeScanner, viewInfo, successCallback, errorCallback	Opens a lookup control to select a one or more items.			
Method	Parameters	Description	refreshParentGrid	entityType, id, name	Refreshes the parent grid containing the specified record.			
loadPanel	url, title	Displays the web page represented by a URL, in the static area in the side pane, which appears on all pages in the model-driven apps web client.	showProgressIndicator	message	Displays a progress dialog with the specified message.			

We help you to execute Web API actions and functions in model-driven apps

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

Xrm.Device

Provides methods to use native device capabilities of mobile devices. It includes `captureAudio`, `captureImage`, `captureVideo`, `getBarcodeValue`, `getCurrentPosition` and `pickFile` functionalities.

Method	Description
<code>captureAudio</code>	Invokes the device microphone to record audio.
<code>captureImage</code>	Invokes the device camera to capture an image.
<code>captureVideo</code>	Invokes the device camera to record video.
<code>getBarcodeValue</code>	Invokes the device camera to scan the barcode information, such as a product number.
<code>getCurrentPosition</code>	Returns the current location using the device geolocation capability.
<code>pickFile</code>	Opens a dialog box to select files for the upload.

Xrm.Encoding

Provides methods to encode strings. It includes `htmlAttributeEncode`, `htmlDecode`, `htmlEncode`, `xmlAttributeEncode` and `xmlEncode` functions.

Method	Description
<code>htmlAttributeEncode</code>	Encodes the specified string so that it can be used in an HTML.
<code>htmlDecode</code>	Converts a string that has been HTML-encoded into a decoded string.
<code>htmlEncode</code>	Converts a string to an HTML-encoded string.
<code>xmlAttributeEncode</code>	Encodes the specified string so that it can be used in an XML.
<code>xmlEncode</code>	Converts a string to an XML-encoded string.

Xrm.Navigation

Provides methods for navigating forms and items in model-driven apps.

Method	Description
navigateTo	Navigates to the specified table list, table record, or HTML web resource.
openAlertDialog	Displays an alert dialog containing a message and a button.
openConfirmDialog	Displays a confirmation dialog box containing a message and two buttons.
openErrorDialog	Displays an error dialog.
openFile	Opens a file.
openForm	Opens an entity form or a quick create form.
openUrl	Opens a URL, including file URLs.
openWebResource	Opens an HTML web resource in a new window.

Xrm.Panel

Provides a method to display a web page in the side pane of model-driven apps form. It provides the functionality of load Panel.

Method	Description
loadPanel	Displays the web page represented by a URL in the static area in the side pane, which appears on all pages in the model-driven apps web client.

We will assist you to use Web API to create
and manage records
Cognitive Convergence
<http://www.cognitiveconvergence.com>
+1 4242530744
shahzad@cognitiveconvergence.com

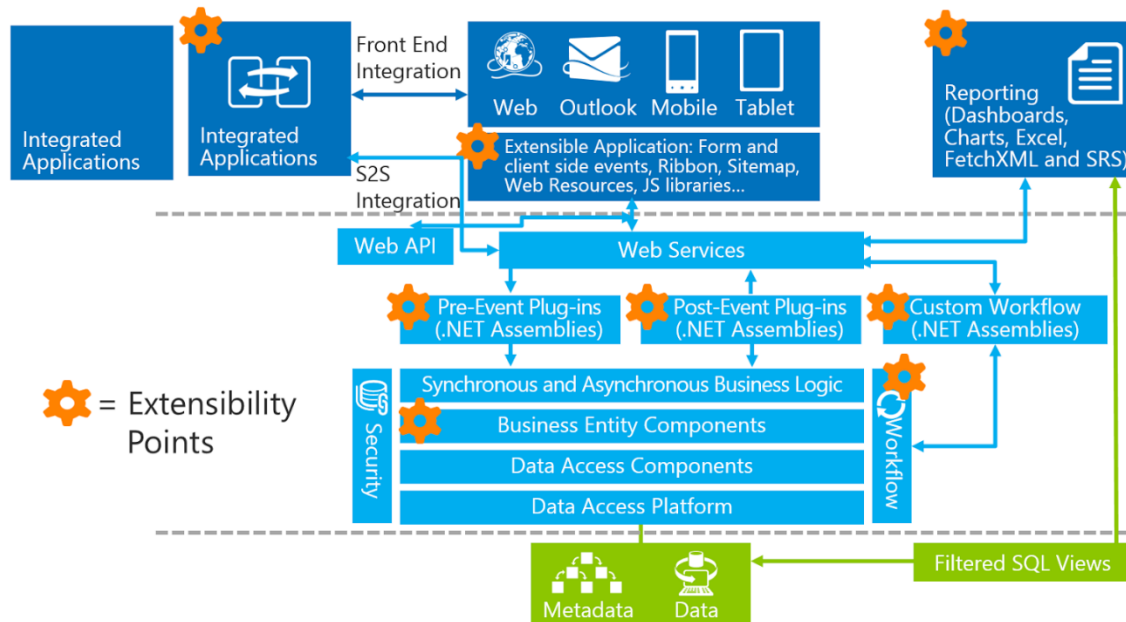
Xrm.Utility

Provides a container for useful methods. It provides a lot of functionalities including `closeProgressIndicator`, `getGlobalContext` and so on.

Method	Description
closeProgressIndicator	Closes a progress dialog box.
getAllowedStatusTransitions	Returns the valid state transitions for the specified table type and status code.
getEntityMetadata	Returns table definitions for the specified table.
getEntityMainFormDescriptor	Returns the default main form descriptor for the specified table.
getGlobalContext	Gets the global context.
getLearningPathAttributeName	Returns the name of the DOM attribute expected by the Learning Path (guided help) Content Designer for identifying UI controls in the model-driven apps forms.
getPageContext	Gets the page context as an object representing the page.
getResourceString	Returns the localized string for a given key associated with the specified web resource.
invokeProcessAction	Invokes an action based on the specified parameters.
lookupObjects	Opens a lookup control to select one or more items.
refreshParentGrid	Refreshes the parent grid containing the specified record.
showProgressIndicator	Displays a progress dialog with the specified message.

Xrm.WebAPI

Provides methods to use Web API to create and manage records and execute Web API actions and functions. It has 2 properties, `Xrm.WebApi.Offline` and `Xrm.WebApi.Online`.



ONLINE AND OFFLINE WEB API

Online Web API

Online Web API provides methods to create and manage records and execute Web API actions and functions in model-driven apps when connected to the model-driven apps server. The `onlineWebApi` object provides the following methods:

- `createRecord`
- `deleteRecord`
- `retrieveRecord`
- `retrieveMultipleRecords`
- `updateRecord`
- `execute`
- `executeMultiple`

We will enable you to connect to the model-driven apps server (online mode) and model-driven apps mobile clients (offline mode).

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

shahzad@cognitiveconvergence.com

+1 4242530744

Offline Web API

Provides methods to create and manage records in the model-driven apps mobile clients while working in the offline mode. When in the offline mode, the functions will work only for tables that are enabled for mobile offline synchronization and available in current user's mobile offline profile. The offlineWebApi object provides the following methods.

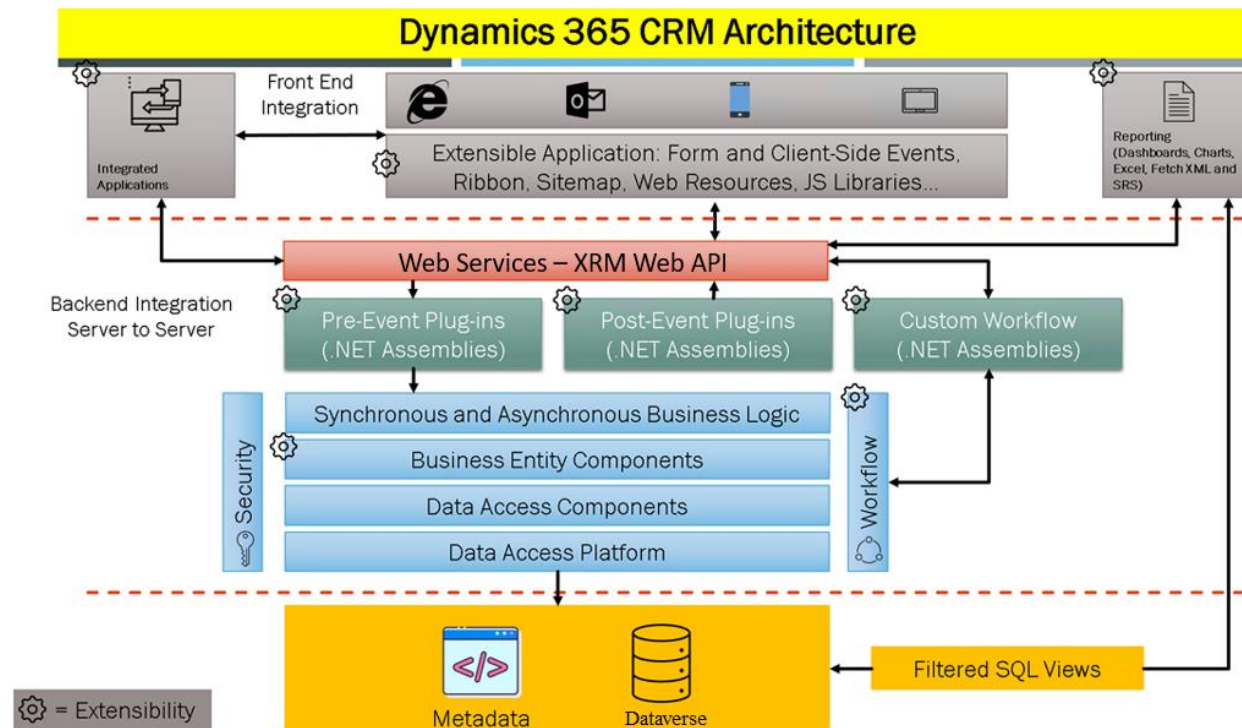
- createRecord
- deleteRecord
- isAvailableOffline
- retrieveRecord
- retrieveMultipleRecords
- updateRecord

Dynamics 365 XRM Web API for CRM and ERP

XRM Web API can be used for all the modules of Dynamics 365, including the Customer Engagement modules and Enterprise Resource Planning modules.



The integration can be done with Power Platform.



We create a development experience that is used across a wide variety of programming languages, platforms, and devices

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

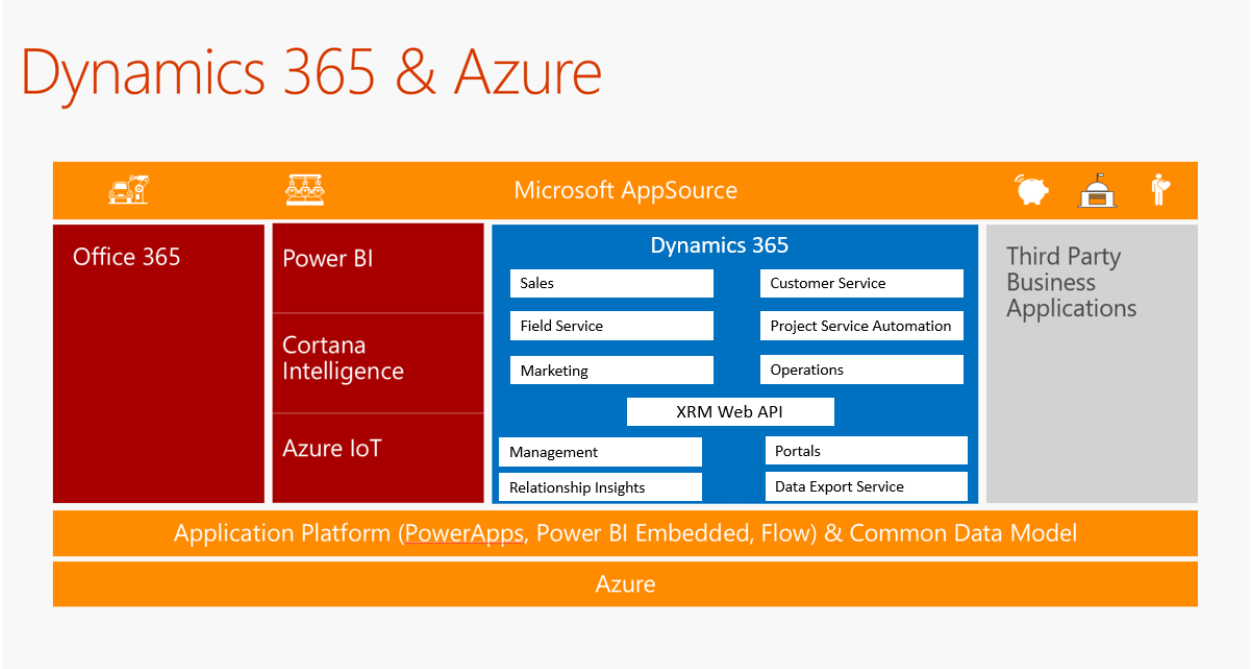
XRM.WEBAPI

The Web API is one of two web services that are to work with data, and table and column definitions in Dataverse. The Dataverse Web API provides a development experience that can be used across a wide variety of programming languages, platforms, and devices.

Xrm.WebApi provides properties and methods to use Web API to create and manage records and execute Web API actions and functions in model-driven apps.

D365 Web API and Azure

Dynamics 365 Web API is extensively used with Azure API management for unification of SaaS systems and exposing APIs to the customers. Using Azure functions and Web API, we can perform CRUD operations in different modules of Dynamics 365.



Methods

The methods that are used in XRM Web API includes:

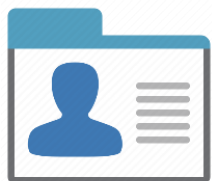
Method	Description
createRecord	Creates an entity record.
deleteRecord	Deletes an entity record.
retrieveRecord	Retrieves an entity record.
retrieveMultipleRecords	Retrieves a collection of entity records.
updateRecord	Updates an entity record.
isAvailableOffline	Returns a Boolean value indicating whether an entity is present in user's profile and is currently available for use in offline mode.
execute	Execute a single action, function, or CRUD operation.
executeMultiple	Execute a collection of action, function, or CRUD operations.

CREATERECORD (CLIENT API REFERENCE)

Creates a table record.

Syntax

```
Xrm.WebApi.createRecord(entityLogicalName, data).then(successCallback, errorCallback);
```



We develop new Xrm.WebApi
JavaScript methods to call the
Dynamics 365 Web APIs

Cognitive Convergence
<http://www.cognitiveconvergence.com>
 +1 4242530744

Parameters

Name	Type	Required	Description
entityLogicalName	String	Yes	Logical name of the entity you want to create. For example: "account".
data	Object	Yes	A JSON object defining the attributes and values for the new entity record. See examples later in this topic to see how you can define the data object for various create scenarios.
successCallback	Function	No	A function to call when a record is created. An object with the following properties will be passed to identify the new record: • entityType: String. The entity logical name of the new record. • id: String. GUID of the new record.
errorCallback	Function	No	A function to call when the operation fails. An object with the following properties will be passed: • errorCode: Number. The error code. • message: String. An error message describing the issue.

Return Value

On success, returns a promise object containing the attributes specified earlier in the description of the **successCallback** parameter.

Our consulting for Dynamics 365 XRM
WebAPI supports querying using
fetchXML

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

Sample Implementation

```
// define the data to create new account
var data =
{
    "name": "Sample Account",
    "creditonhold": false,
    "address1_latitude": 47.639583,
    "description": "This is the description of the sample account",
    "revenue": 5000000,
    "accountcategorycode": 1
}

// create account record
Xrm.WebApi.createRecord("account", data).then(
    function success(result) {
        console.log("Account created with ID: " + result.id);
        // perform operations on record creation
    },
    function (error) {
        console.log(error.message);
        // handle error conditions
    }
);
```

DELETERECORD (CLIENT API REFERENCE)

Deletes an entity record.

Syntax

Xrm.WebApi.deleteRecord(entityLogicalName, id).then(successCallback, errorCallback);



Parameters

Name	Type	Required	Description
entityLogicalName	String	Yes	The entity logical name of the record you want to delete. For example: "account".
id	String	Yes	GUID of the entity record you want to delete.
successCallback	Function	No	A function to call when a record is deleted. An object with the following properties will be passed to identify the deleted record: • entityType: String. The entity type of the record. • id: String. GUID of the record. • name: String. Name of the record.
errorCallback	Function	No	A function to call when the operation fails.

Return Value

On success, returns a promise object containing the attributes specified earlier in the description of the **successCallback** parameter.

Sample Implementation

It deletes a record with a specific account id.

```
Xrm.WebApi.deleteRecord("account", "5531d753-95af-e711-a94e-000d3a11e605").then(  
  function success(result) {  
    console.log("Account deleted");  
    // perform operations on record deletion  
  },  
  function (error) {  
    console.log(error.message);  
    // handle error conditions  
  }  
);
```


RETRIEVERECORD (CLIENT API REFERENCE)

Retrieves an entity record.

Syntax

```
Xrm.WebApi.retrieveRecord(entityLogicalName, id, options).then(successCallback, errorCallback);
```



Parameters

Name	Type	Required	Description
entityLogicalName	String	Yes	The entity logical name of the record you want to retrieve. For example: "account".
id	String	Yes	GUID of the entity record you want to retrieve.
options	String	No	OData system query options, \$select and \$expand , to retrieve your data. • \$select system query option to limit the properties returned by including a comma-separated list of property names. • \$expand system query option to control what data from related entities is returned.
successCallback	Function	No	A function to call when a record is retrieved. A JSON object with the retrieved properties and values will be passed to the function.
errorCallback	Function	No	A function to call when the operation fails.

Allow us to perform CRUD operations using
Xrm.WebApi

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

Return Value

On success, returns a promise containing a JSON object with the retrieved attributes and their values.

Sample Implementation

Retrieves the account with the given account number and filtered only on the bases of "name" and revenue" fields.

```
xrm.WebApi.retrieveRecord("account", "a8a19cdd-88df-e311-b8e5-6c3be5a8b200", "?$select=name,revenue").then(  
  function success(result) {  
    console.log("Retrieved values: Name: " + result.name + ", Revenue: " + result.revenue);  
    // perform operations on record retrieval  
  },  
  function (error) {  
    console.log(error.message);  
    // handle error conditions  
  }  
);
```

UPDATERECORD (CLIENT API REFERENCE)

Updates an entity record.

Syntax

```
Xrm.WebApi.updateRecord(entityLogicalName, id, data).then(successCallback, errorCallback);
```



Parameters

Name	Type	Required	Description
entityLogicalName	String	Yes	The entity logical name of the record you want to update. For example: "account".
id	String	Yes	GUID of the entity record you want to update.
data	Object	Yes	A JSON object containing key: value pairs
successCallback	Function	No	A function to call when a record is updated. An object with the following properties will be passed to identify the updated record: • entityType: String. The entity type of the updated record. • id: String. GUID of the updated record.
errorCallback	Function	No	A function to call when the operation fails. An object with the following properties will be passed: • errorCode: Number. The error code. • message: String. An error message describing the issue.

Return Value

On success, returns a promise object containing the attributes specified earlier in the description of the **successCallback** parameter.

Sample Implementation

Updates an existing account record with record ID = 5531d753-95af-e711-a94e-000d3a11e605.

```
// define the data to update a record
var data =
{
    "name": "Updated Sample Account ",
    "creditonhold": true,
    "address1_latitude": 47.639583,
    "description": "This is the updated description of the sample account",
    "revenue": 6000000,
    "accountcategorycode": 2
}
// update the record
Xrm.WebApi.updateRecord("account", "5531d753-95af-e711-a94e-000d3a11e605", data).then(
    function success(result) {
        console.log("Account updated");
        // perform operations on record update
    },
    function (error) {
        console.log(error.message);
        // handle error conditions
    }
);
```

RETRIEVEMULTIPLERECORDS (CLIENT API REFERENCE)

Retrieves a collection of table records.

Syntax

Xrm.WebApi.retrieveMultipleRecords(entityLogicalName, options, maxPageSize).then(successCallback, errorCallback);



Parameters

Name	Type	Required	Description
entityLogicalName	String	Yes	The table logical name of the records you want to retrieve. For example: "account".
options	String	No	OData system query options or FetchXML query to retrieve your data.
maxPageSize	Number	No	A positive number that indicates the number of table records to be returned per page
successCallback	Function	No	A function to call when table records are retrieved. An object with the following values is passed to the function: • entities • nextLink • fetchXmlPagingCookie
errorCallback	Function	No	A function to call when the operation fails.

Return Value

For a successful OData query retrieveMultipleRecords operation, returns a promise that contains an array of JSON objects (**entities**) containing the retrieved table records and the **nextLink** attribute (optional) with the URL pointing to next page of records in case paging (maxPageSize) is specified in the request, and the record count returned exceeds the paging value.

For a successful FetchXML-based retrieveMultipleRecords operations the promise response will contain a **fetchXmlPagingCookie** (optional) attribute when the operation returns more records than the paging value. This attribute will contain the paging cookie string that can be included in a subsequent fetchXml request to fetch the next page of records.

Sample Implementation

Most of the scenarios/examples mentioned in Query Data using the Web API can be achieved using the **retrieveMultipleRecords** method.

The following example queries the accounts table set and uses the \$select and \$top system query options to return the name property for the first three accounts:

ISAVAILABLEOFFLINE (CLIENT API REFERENCE)

Returns a boolean value indicating whether an entity is present in user's profile and is currently available for use in offline mode.

Syntax



```
Xrm.WebApi.offline.isAvailableOffline(entityLogicalName);
```

```
Xrm.WebApi.retrieveMultipleRecords("account", "?$select=name&$top=3").then(
    function success(result) {
        for (var i = 0; i < result.entities.length; i++) {
            console.log(result.entities[i]);
        }
        // perform additional operations on retrieved records
    },
    function (error) {
        console.log(error.message);
        // handle error conditions
    }
);
```

ParametersName	Type	Required	Description
entityLogicalName	String	Yes	Logical name of the table. For example: "account".

Return Value

Type: Boolean.

Description: true if the table is present in user's profile and is currently available for use in offline mode; otherwise, false.

CONCLUSION

In this paper, a brief introduction is discussed about XRM object & one of its namespace Xrm.WebApi is discussed thoroughly with respect to its for methods.

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 services.

