To integration Dynamics 365 web API and any other product like Node.js, we need to create Azure App.

After creating azure app we can integrate Dynamics 365 with our other product.

There are two types of Azure Application in Azure.

1. Native Application
2. Web App /API Application.

We can use Both type of application in integration. But the major difference is:-

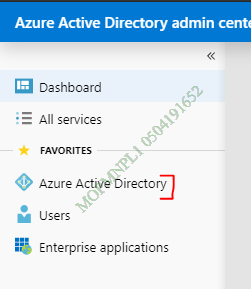
1. We need to use User name and Password while integrate via Native APP.
2. We need to use secret key and application user while integrate via Web App /API APP.

**Native APP Integration:-**

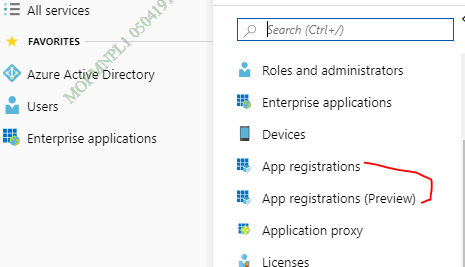
1. Login in Azure with admin permission

[https://aad.portal.azure.com](https://aad.portal.azure.com/)

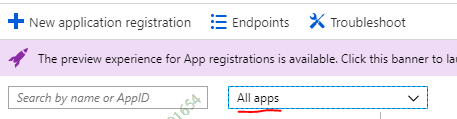
1. Click on “Azure Active Directory”



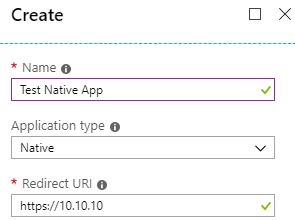
1. Click on App registration.



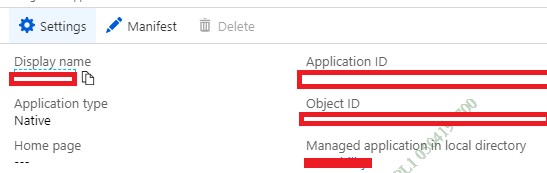
1. Here we will find “My Apps” and “All Apps”.



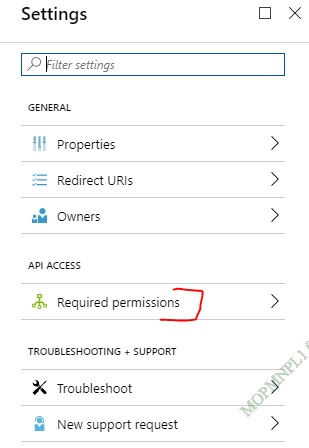
1. Click on “New application registration”.
2. Provide information and select Application Type as “Native”.



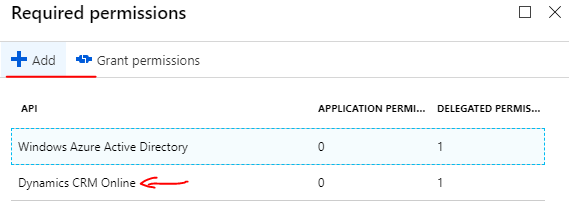
1. **Copy Application ID to use as Client Id while integration.**



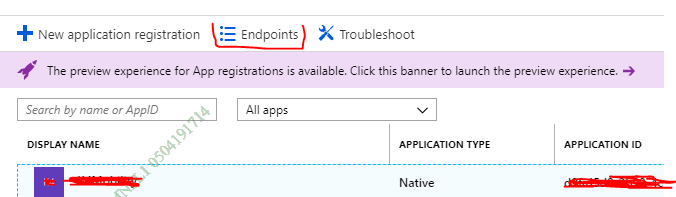
1. Click on settings and select “Required Permissions”.



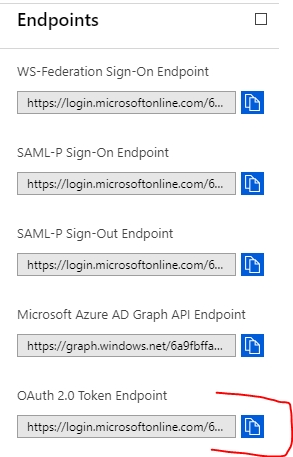
1. Add Dynamics CRM Online in permission.



1. Select your application and click on “Endpoints”.

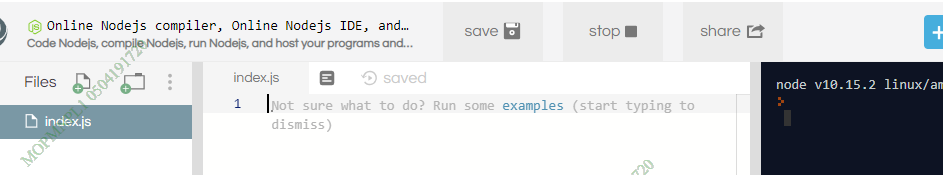


1. **Copy “OAuth 2.0 Token Endpoint” for further use.**



1. Now our App ready to integrate for other product like Node.Js
2. To test application Node.js online, open <https://repl.it/languages/javascript>

and paste below code. Update organization and application information in Code.



//Code started

'use strict';

var https = require('https');

//set these values to retrieve the oauth token

var crmorg = 'https://CRMORG.crm8.dynamics.com';

//Application ID

var clientid = 'XXXXX-XXXX-XXXX-XXXXX-XXXX';

var username = username@domain.onmicrosoft.com';

var userpassword = 'Password';

var tokenendpoint = 'Your copied token';

//set these values to query your crm data

var crmwebapihost = ‘domain.api.crm8.dynamics.com';

var fetchQuery = "%3Cfetch%20top%3D%2250%22%3E%0A%3Centity%20name%3D%22dynamicproperty%22%3E%0A%3Cattribute%20name%3D%22defaultvaluestring%22%2F%3E%0A%3Cattribute%20name%3D%22name%22%2F%3E%0A%3Cfilter%3E%0A%3Ccondition%20attribute%3D%22statecode%22%20operator%3D%22eq%22%20value%3D%220%22%2F%3E%0A%3Ccondition%20attribute%3D%22statuscode%22%20operator%3D%22eq%22%20value%3D%221%22%2F%3E%0A%3C%2Ffilter%3E%0A%3Clink-entity%20name%3D%22product%22%20from%3D%22parentproductid%22%20to%3D%22regardingobjectid%22%20alias%3D%22Product%22%3E%0A%3Cattribute%20name%3D%22name%22%2F%3E%0A%3Cattribute%20name%3D%22productid%22%2F%3E%0A%3Cfilter%3E%0A%3Ccondition%20attribute%3D%22statuscode%22%20operator%3D%22eq%22%20value%3D%221%22%2F%3E%0A%3Ccondition%20attribute%3D%22k4\_productsubtype%22%20operator%3D%22eq%22%20value%3D%22636130000%22%2F%3E%0A%3C%2Ffilter%3E%0A%3C%2Flink-entity%3E%0A%3C%2Fentity%3E%0A%3C%2Ffetch%3E";

var crmwebapipath = '/api/data/v9.1/dynamicproperties?fetchXml=' + fetchQuery;

//remove https from tokenendpoint url

tokenendpoint = tokenendpoint.toLowerCase().replace('https://', '');

//get the authorization endpoint host name

var authhost = tokenendpoint.split('/')[0];

//get the authorization endpoint path

var authpath = '/' + tokenendpoint.split('/').slice(1).join('/');

//build the authorization request

//if you want to learn more about how tokens work, see IETF RFC 6749 - https://tools.ietf.org/html/rfc6749

var reqstring = 'client\_id=' + clientid;

reqstring += '&resource=' + encodeURIComponent(crmorg);

reqstring += '&username=' + encodeURIComponent(username);

reqstring += '&password=' + encodeURIComponent(userpassword);

reqstring += '&grant\_type=password';

//set the token request parameters

var tokenrequestoptions = {

host: authhost,

path: authpath,

method: 'POST',

headers: {

'Content-Type': 'application/x-www-form-urlencoded',

'Content-Length': Buffer.byteLength(reqstring)

}

};

//make the token request

var tokenrequest = https.request(tokenrequestoptions, function (response) {

//make an array to hold the response parts if we get multiple parts

var responseparts = [];

response.setEncoding('utf8');

response.on('data', function (chunk) {

//add each response chunk to the responseparts array for later

responseparts.push(chunk);

});

response.on('end', function () {

//once we have all the response parts, concatenate the parts into a single string

var completeresponse = responseparts.join('');

//console.log('Response: ' + completeresponse);

console.log('Token response retrieved . . . ');

//parse the response JSON

var tokenresponse = JSON.parse(completeresponse);

//extract the token

var token = tokenresponse.access\_token;

//pass the token to our data retrieval function

getData(token);

});

});

tokenrequest.on('error', function (e) {

console.error(e);

});

//post the token request data

tokenrequest.write(reqstring);

//close the token request

tokenrequest.end();

function getData(token) {

//set the web api request headers

var requestheaders = {

'Authorization': 'Bearer ' + token,

'OData-MaxVersion': '4.0',

'OData-Version': '4.0',

'Accept': 'application/json',

'Content-Type': 'application/json; charset=utf-8',

'Prefer': 'odata.maxpagesize=500',

'Prefer': 'odata.include-annotations=OData.Community.Display.V1.FormattedValue'

};

//set the crm request parameters

var crmrequestoptions = {

host: crmwebapihost,

path: crmwebapipath,

method: 'GET',

headers: requestheaders

};

//make the web api request

var crmrequest = https.request(crmrequestoptions, function (response) {

//make an array to hold the response parts if we get multiple parts

var responseparts = [];

response.setEncoding('utf8');

response.on('data', function (chunk) {

//add each response chunk to the responseparts array for later

responseparts.push(chunk);

});

response.on('end', function () {

//once we have all the response parts, concatenate the parts into a single string

var completeresponse = responseparts.join('');

//parse the response JSON

var collection = JSON.parse(completeresponse).value;

//loop through the results and write out the fullname

collection.forEach(function (row, i) {

console.log(row['Product.name'] + ":" + row['Product.productid'] + ":" + row['name'] + ":" + row['defaultvaluestring']);

});

});

});

crmrequest.on('error', function (e) {

console.error(e);

});

//close the web api request

crmrequest.end();

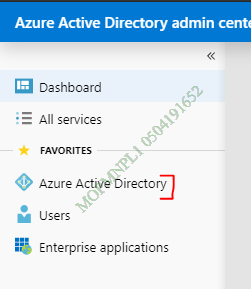
}

**Web App /API APP:-**

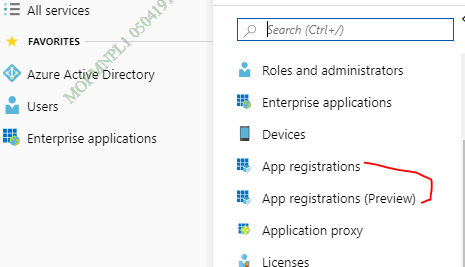
1. Login in Azure with admin permission

[https://aad.portal.azure.com](https://aad.portal.azure.com/)

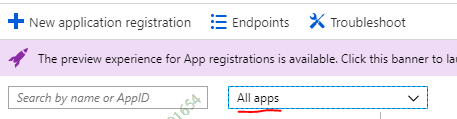
1. Click on “Azure Active Directory”



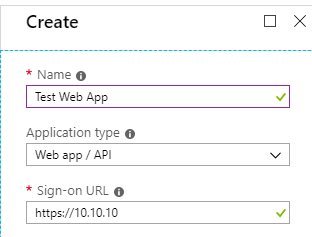
1. Click on App registration.



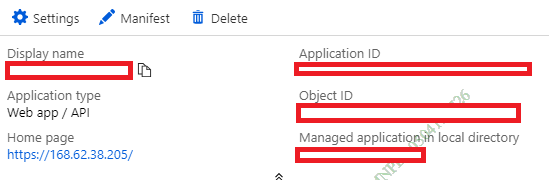
1. Here we will find “My Apps” and “All Apps”.



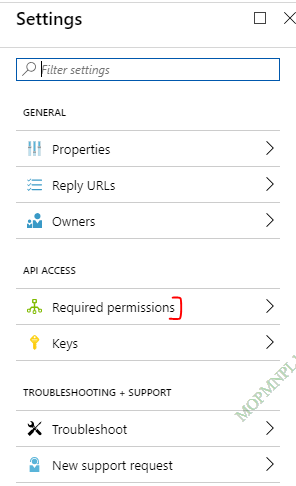
1. Click on “New application registration”.
2. Provide information and select Application Type “Web App/ API”.



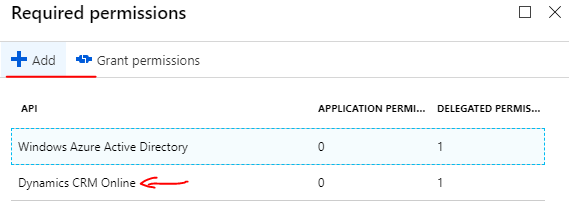
1. **Copy Application ID to use as Client Id while integration.**



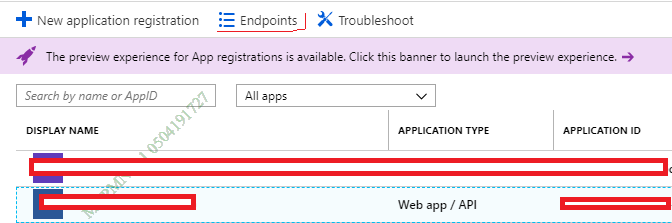
1. Click on settings and select “Required Permissions”.



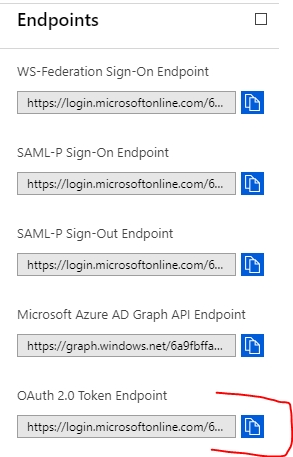
1. Add Dynamics CRM Online in permission.



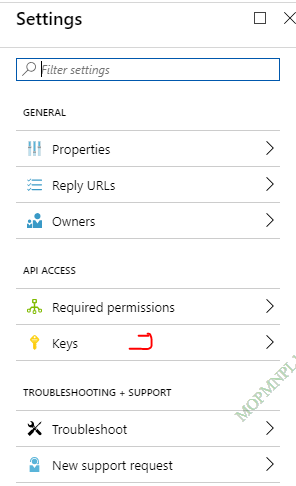
1. Select your application and click on “Endpoints”.



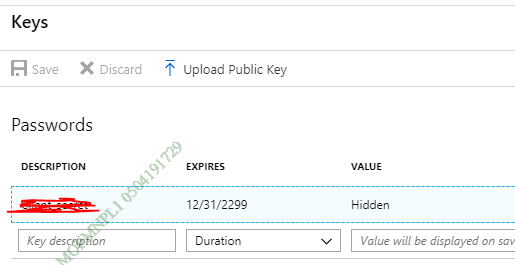
1. **Copy “OAuth 2.0 Token Endpoint” for further use.**



1. Click on settings and select “Key”

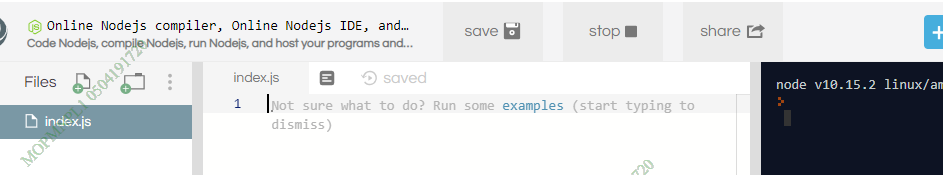


1. Create a Key and copy its value. It is important to copy key otherwise you will not find this value later.



1. Now our App ready to integrate for other product like Node.Js
2. To test application Node.js online, open <https://repl.it/languages/javascript>

and paste below code. Update organization and application information in Code.



//Code started

'use strict';

var https = require('https');

//set these values to retrieve the oauth token

var crmorg = 'https://crmOrg.crm8.dynamics.com';

//Application ID

var clientid = XXXXX-XXX-45XXX-XXXX-XXXXXXXX;

var Client\_secret = 'KqzDdfdsforqLD0uiLffb0tMYam3994vlas=';

var tokenendpoint = 'Copied Oath URL';

//set these values to query your crm data

var crmwebapihost = 'domain.api.crm8.dynamics.com';

var fetchQuery = "%3Cfetch%20top%3D%2250%22%3E%0A%3Centity%20name%3D%22dynamicproperty%22%3E%0A%3Cattribute%20name%3D%22defaultvaluestring%22%2F%3E%0A%3Cattribute%20name%3D%22name%22%2F%3E%0A%3Cfilter%3E%0A%3Ccondition%20attribute%3D%22statecode%22%20operator%3D%22eq%22%20value%3D%220%22%2F%3E%0A%3Ccondition%20attribute%3D%22statuscode%22%20operator%3D%22eq%22%20value%3D%221%22%2F%3E%0A%3C%2Ffilter%3E%0A%3Clink-entity%20name%3D%22product%22%20from%3D%22parentproductid%22%20to%3D%22regardingobjectid%22%20alias%3D%22Product%22%3E%0A%3Cattribute%20name%3D%22name%22%2F%3E%0A%3Cattribute%20name%3D%22productid%22%2F%3E%0A%3Cfilter%3E%0A%3Ccondition%20attribute%3D%22statuscode%22%20operator%3D%22eq%22%20value%3D%221%22%2F%3E%0A%3Ccondition%20attribute%3D%22k4\_productsubtype%22%20operator%3D%22eq%22%20value%3D%22636130000%22%2F%3E%0A%3C%2Ffilter%3E%0A%3C%2Flink-entity%3E%0A%3C%2Fentity%3E%0A%3C%2Ffetch%3E";

var crmwebapipath = '/api/data/v9.1/dynamicproperties?fetchXml=' + fetchQuery;

//remove https from tokenendpoint url

tokenendpoint = tokenendpoint.toLowerCase().replace('https://', '');

//get the authorization endpoint host name

var authhost = tokenendpoint.split('/')[0];

//get the authorization endpoint path

var authpath = '/' + tokenendpoint.split('/').slice(1).join('/');

//build the authorization request

//if you want to learn more about how tokens work, see IETF RFC 6749 - https://tools.ietf.org/html/rfc6749

var reqstring = 'client\_id=' + clientid;

reqstring += '&resource=' + encodeURIComponent(crmorg);

reqstring += '&client\_secret=' + Client\_secret;

reqstring += '&grant\_type=client\_credentials';

//set the token request parameters

var tokenrequestoptions = {

host: authhost,

path: authpath,

method: 'POST',

headers: {

'Content-Type': 'application/x-www-form-urlencoded',

'Content-Length': Buffer.byteLength(reqstring)

}

};

//make the token request

var tokenrequest = https.request(tokenrequestoptions, function (response) {

//make an array to hold the response parts if we get multiple parts

var responseparts = [];

response.setEncoding('utf8');

response.on('data', function (chunk) {

//add each response chunk to the responseparts array for later

responseparts.push(chunk);

});

response.on('end', function () {

//once we have all the response parts, concatenate the parts into a single string

var completeresponse = responseparts.join('');

//console.log('Response: ' + completeresponse);

console.log('Token response retrieved . . . ');

//parse the response JSON

var tokenresponse = JSON.parse(completeresponse);

//extract the token

var token = tokenresponse.access\_token;

//console.log(token);

//pass the token to our data retrieval function

getData(token);

});

});

tokenrequest.on('error', function (e) {

console.error(e);

});

//post the token request data

tokenrequest.write(reqstring);

//close the token request

tokenrequest.end();

function getData(token) {

//set the web api request headers

var requestheaders = {

'Authorization': 'Bearer ' + token,

'OData-MaxVersion': '4.0',

'OData-Version': '4.0',

'Accept': 'application/json',

'Content-Type': 'application/json; charset=utf-8',

'Prefer': 'odata.maxpagesize=500',

'Prefer': 'odata.include-annotations=OData.Community.Display.V1.FormattedValue'

};

//set the crm request parameters

var crmrequestoptions = {

host: crmwebapihost,

path: crmwebapipath,

method: 'GET',

headers: requestheaders

};

//make the web api request

var crmrequest = https.request(crmrequestoptions, function (response) {

//make an array to hold the response parts if we get multiple parts

var responseparts = [];

response.setEncoding('utf8');

response.on('data', function (chunk) {

//add each response chunk to the responseparts array for later

responseparts.push(chunk);

});

response.on('end', function () {

//once we have all the response parts, concatenate the parts into a single string

var completeresponse = responseparts.join('');

//parse the response JSON

if (completeresponse != "") {

var collection = JSON.parse(completeresponse).value;

//loop through the results and write out the fullname

collection.forEach(function (row, i) {

console.log(row['Product.name'] + ":" + row['Product.productid'] + ":" + row['name'] + ":" + row['defaultvaluestring']);

});

}

});

});

crmrequest.on('error', function (e) {

console.error(e);

});

//crmrequest.write(crmrequestoptions);

//close the web api request

crmrequest.end();

}