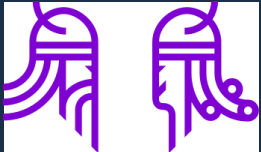


2020-01-28: DFDS A/S Tech Hub @ Harbour House II

# CrmWebApiUtil + LINQ Provider

F# dotnet + C# dotnet = Cloud Ready (Docker)



```
DFDS.CEM.CustomerDataIntegration.Services.CRM.v1 - FilterHelper(Source settings, Guid?[] accountids)
var accounts = new List<ModPAX.Account>();
try
{
    using var crm = new CrmClient(settings);
    var ctx = new WebAPI.Context<ModPAX.Account>(crm, Re1PAX.Lookups);
    var query =
        from a in ctx
        where
            accountids.Contains(a.AccountId)
        select
            new ModPAX.Account
            {
                AccountId = a.AccountId
                , AccountNumber = a.AccountNumber
                /*
                , Name = a.Name
                , CustomerTypeCode = a.CustomerTypeCode
                , NumberOfEmployees = a.NumberOfEmployees
                ,
                , OwningUser =
                new ModPAX.SystemUser
                {
                    SystemUserId = a.OwningUser.SystemUserId
                    , FullName = a.OwningUser.FullName
                }
                , OwningTeam =
                new ModPAX.Team
                {
                    TeamId = a.OwningTeam.TeamId
                    , Name = a.OwningTeam.Name
                }
                ,
                /*
                ,
                , StageId = a.StageId
                ,
                , PrimaryContactId =
                new ModPAX.Contact
                {
                    ContactId = a.PrimaryContactId.ContactId
                }
            };
    accounts.AddRange(query);
}
catch (Exception ex)
{
    throw new ErrorHandling.Services.CRM.Account
        ( ex.Message
        , ex.InnerException
        );
}

return await Tasks.Task.Run(() => accounts);
}
```



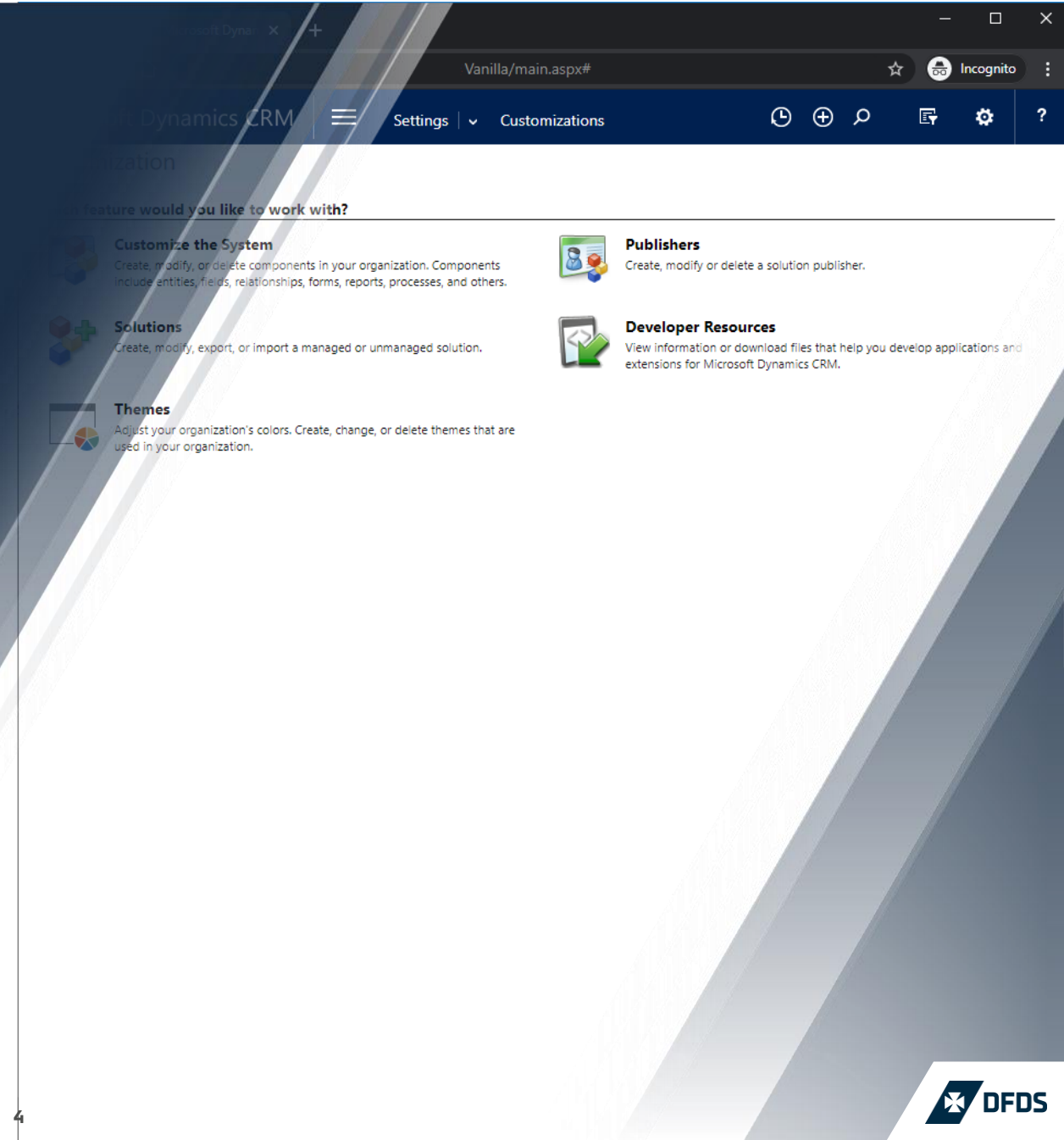


# Matching of expectations

- A short **introduction** to the platform: **Microsoft CRM** / Dynamics 365 Customer Engagement
- What **problem** are we **solving** and why is it a problem in the first place
- Hopefully we will be able to convince you that the **chosen approach gives sense** and you might find **inspiration** to do something similar with your own platform

**Note:** Please **save your questions** to the **QA** at the end of the talk unless I say something that is really **incorrect** 😊

# Background



The screenshot shows the Microsoft Dynamics CRM Customizations page in an Incognito browser window. The page title is "Microsoft Dynamics CRM" and the URL is "Vanilla/main.aspx#". The navigation bar includes "Settings" and "Customizations". The main content area is titled "Customization" and features a section "Which feature would you like to work with?". This section contains four cards: "Customize the System" (Create, modify, or delete components in your organization. Components include entities, fields, relationships, forms, reports, processes, and others.), "Solutions" (Create, modify, export, or import a managed or unmanaged solution.), "Themes" (Adjust your organization's colors. Create, change, or delete themes that are used in your organization.), "Publishers" (Create, modify or delete a solution publisher.), and "Developer Resources" (View information or download files that help you develop applications and extensions for Microsoft Dynamics CRM.).

Microsoft Dynamics CRM

Vanilla/main.aspx#

Settings | Customizations

Customization

Which feature would you like to work with?

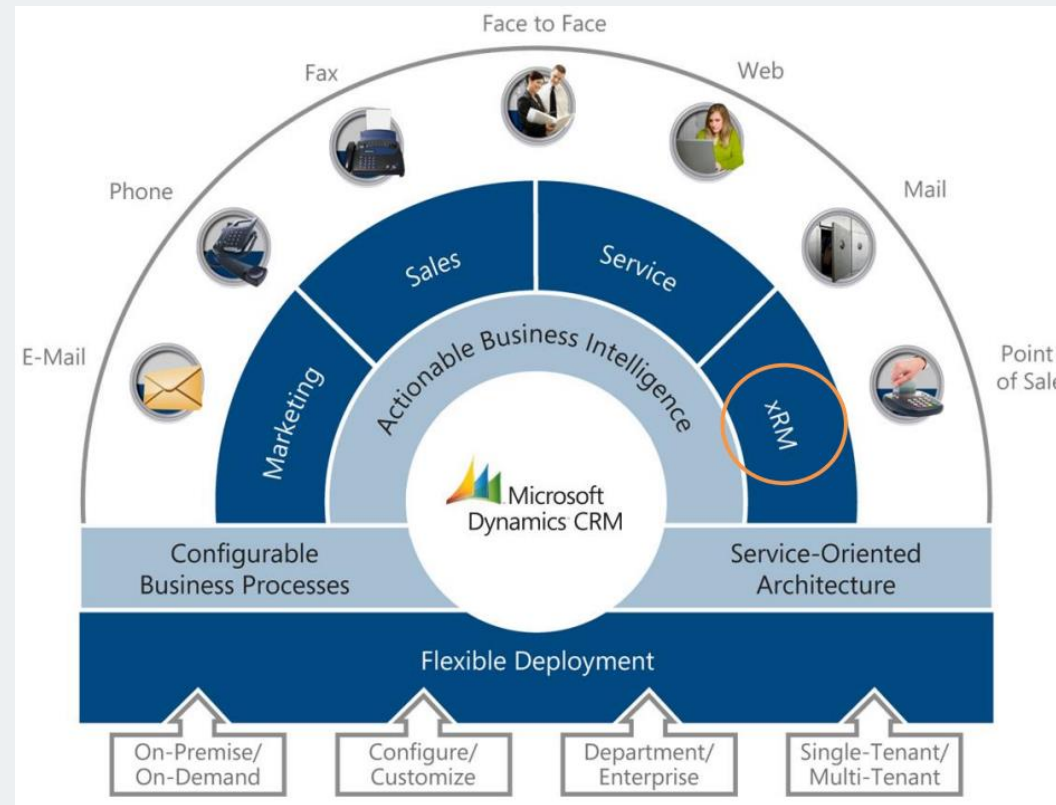
- Customize the System**  
Create, modify, or delete components in your organization. Components include entities, fields, relationships, forms, reports, processes, and others.
- Solutions**  
Create, modify, export, or import a managed or unmanaged solution.
- Themes**  
Adjust your organization's colors. Create, change, or delete themes that are used in your organization.
- Publishers**  
Create, modify or delete a solution publisher.
- Developer Resources**  
View information or download files that help you develop applications and extensions for Microsoft Dynamics CRM.

4

DFDS

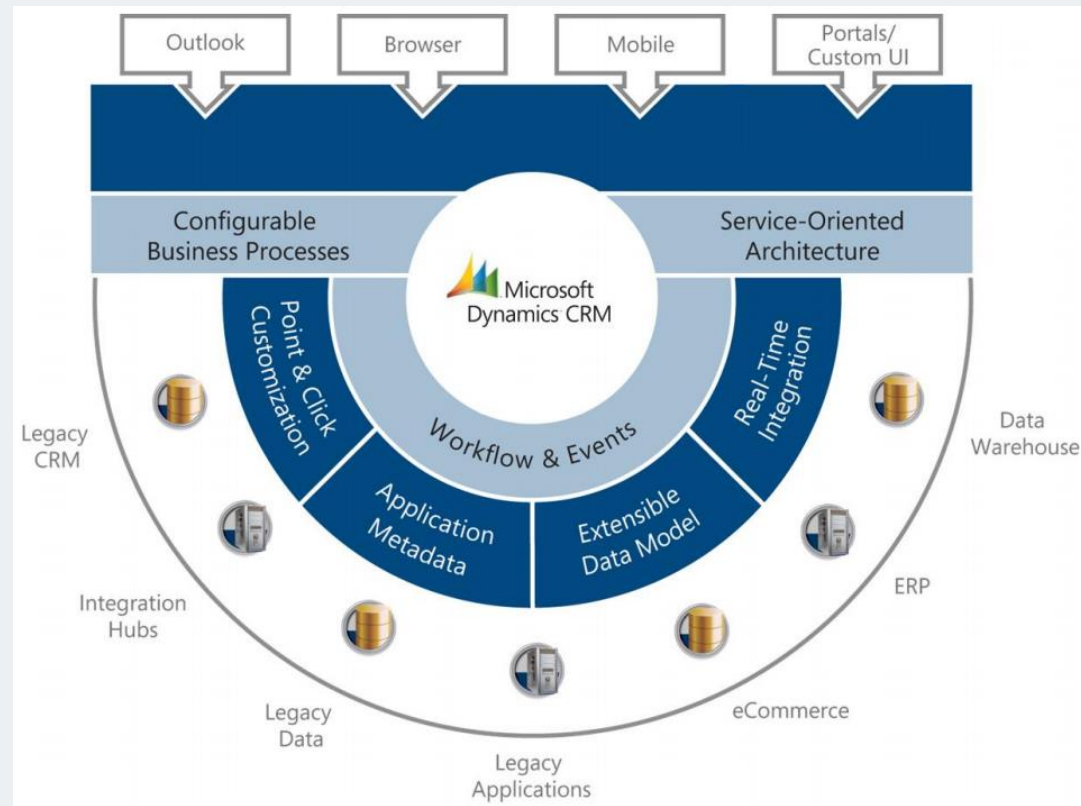
# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Overview > Customer Relationship Management



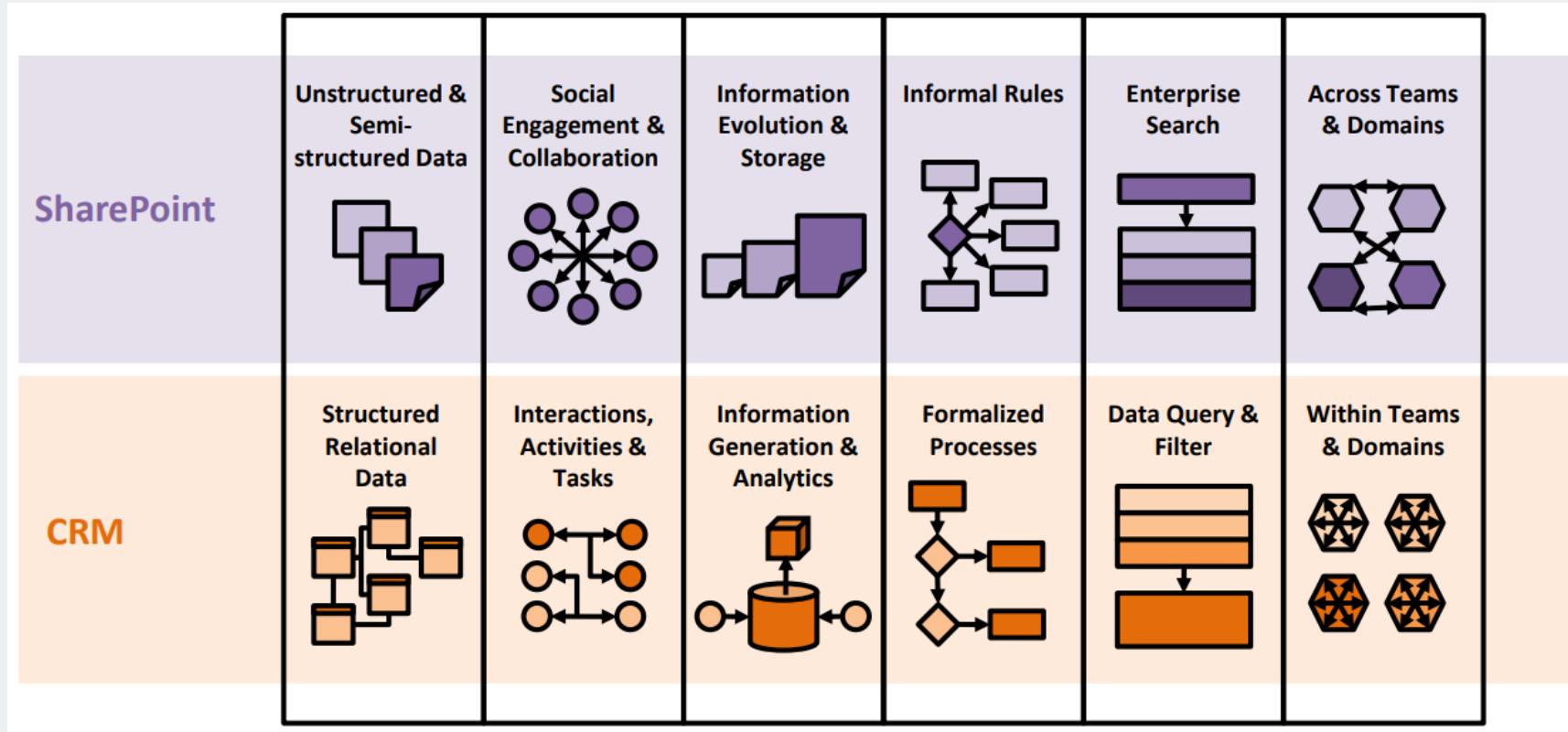
# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Overview > Integration Hub



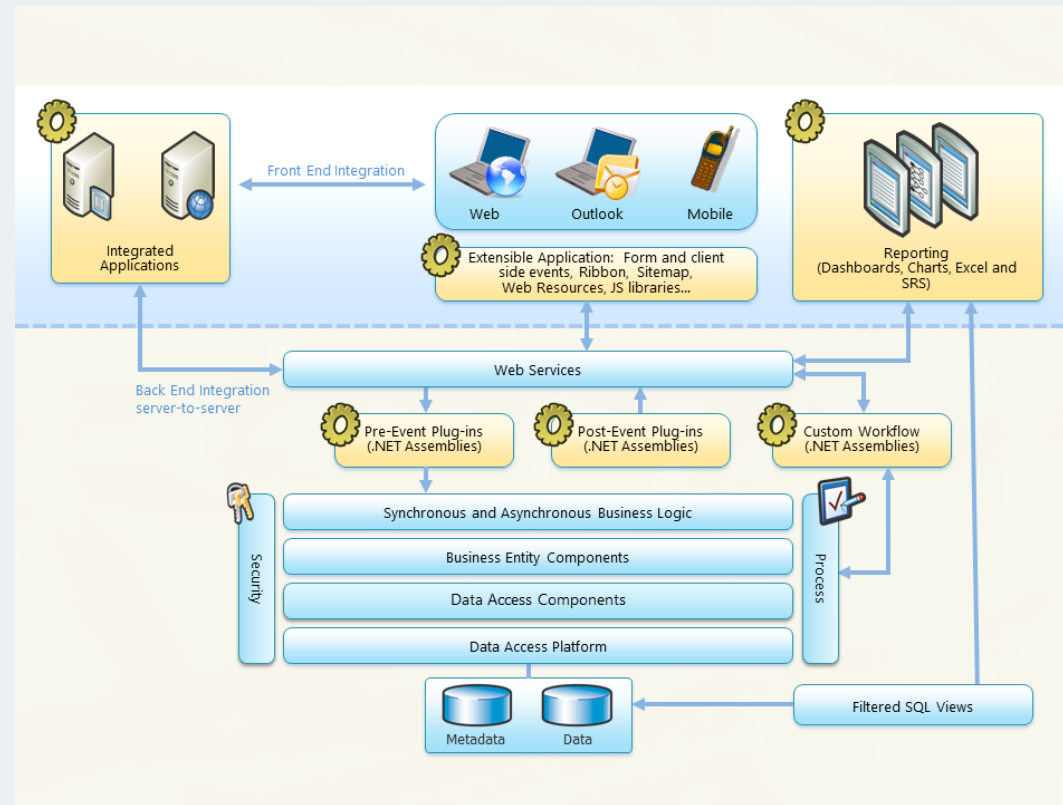
# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > Loose (SharePoint) vs Very-tight and formal (CRM)



# Microsoft CRM / Dynamics 365 Customer Engagement

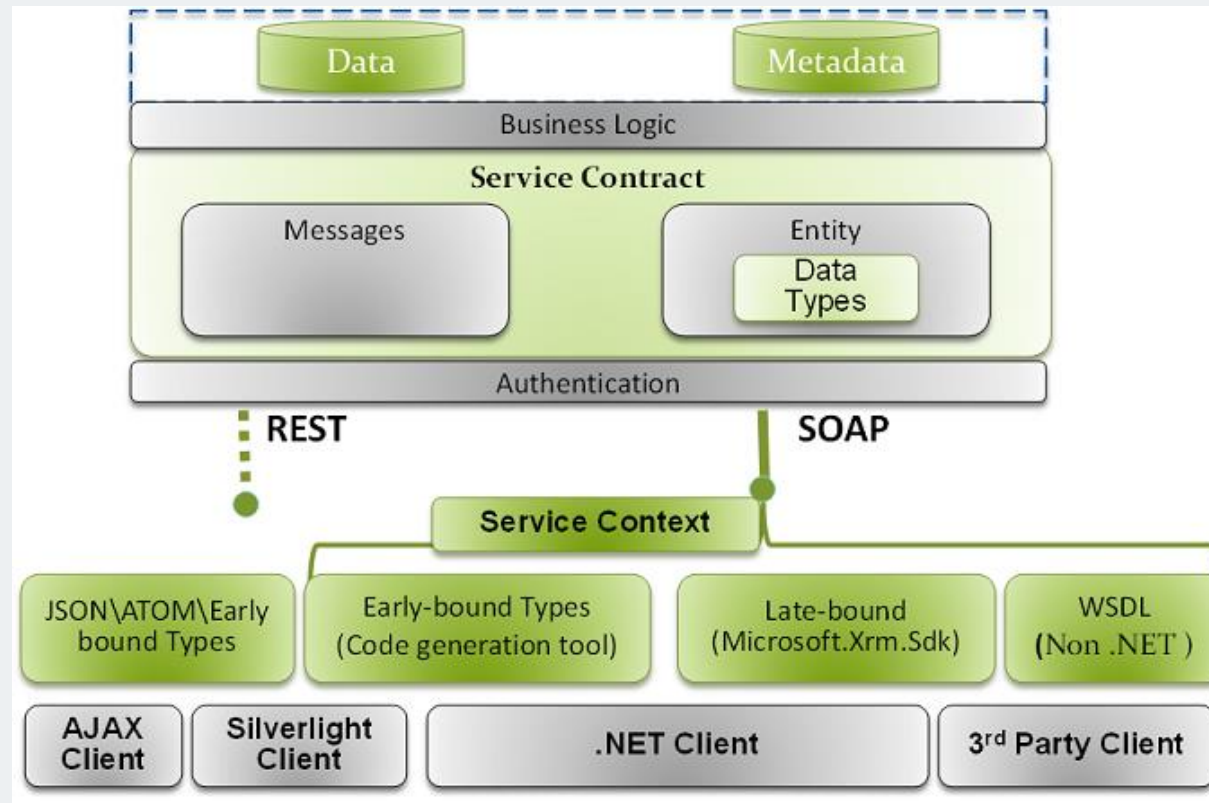
Background > Platform > Architecture > MVC pattern (state changes go through WS Controller and Views can read directly)





# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > SOAP (XML) and REST / WebAPI (JSON)



# Microsoft CRM / Dynamics 365 Customer Engagement

## Background > Platform > Architecture > SOAP (XML) > Early-bound (Code generation tool)

- Create early bound entity classes with the code generation tool (**CrmsvcUtil**):
  - <https://docs.microsoft.com/en-us/dynamics365/customerengagement/on-premises/developer/org-service/create-early-bound-entity-classes-code-generation-tool>
- The tool will create **type-safe C# classes** based on the data model from CRM application instances
- This will allow to write code with **LINQ** (subset of Entity Framework) in order to receive and send data to CRM
- Maintained by **Microsoft** and accessible through **NuGet**

# Microsoft CRM / Dynamics 365 Customer Engagement

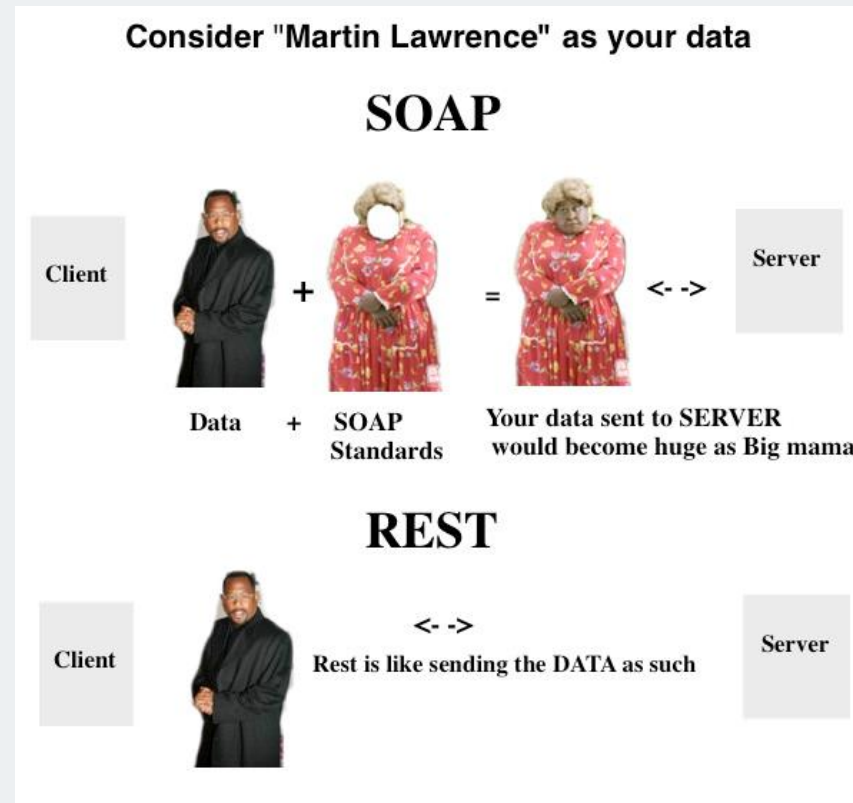
## Background > Platform > Architecture > SOAP (XML) > Early-bound (Code generation tool)

- Alternatively use **XrmContext**, as it's very similar to **CrmsvcUtil**, but it has several new features that help you code better and more reliably, and generates files that takes up less space:
  - <https://github.com/delegateas/XrmContext/wiki>
- Maintained by **Delegate A/S** and accessible through **NuGet** as well



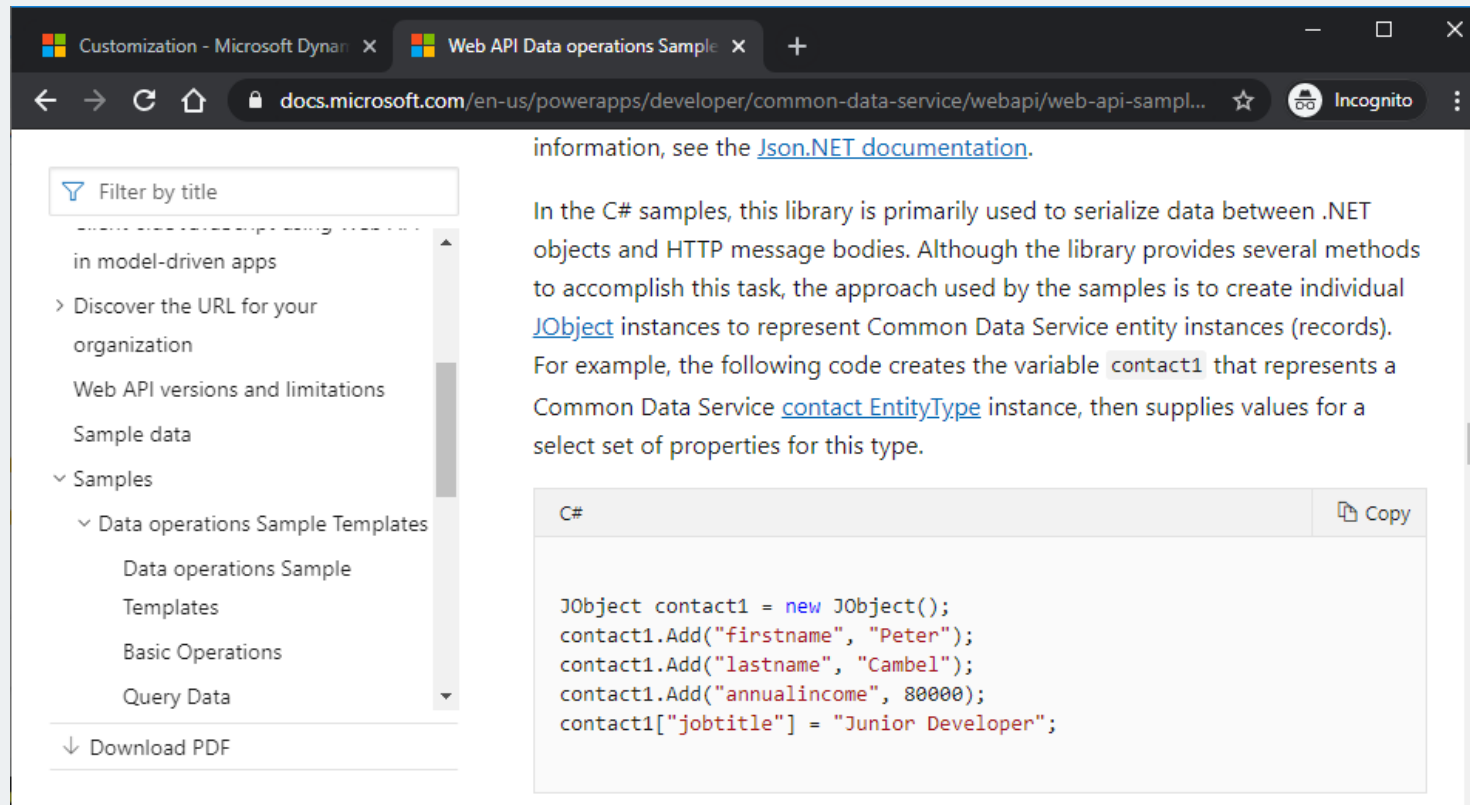
# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > SOAP (XML) vs REST / WebAPI (JSON)



# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Send > Untyped, well JSON (Newtonsoft) typed !



The screenshot shows a web browser window with two tabs: "Customization - Microsoft Dyna..." and "Web API Data operations Sample...". The address bar shows the URL: [docs.microsoft.com/en-us/powerapps/developer/common-data-service/webapi/web-api-sampl...](https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/webapi/web-api-sampl...). The page content includes a sidebar with a search filter and a list of navigation items. The main content area contains text explaining the use of the `JsonObject` library in C# samples for serializing data between .NET objects and HTTP message bodies. A code block shows the following C# code:

```
C# Copy
JsonObject contact1 = new JsonObject();
contact1.Add("firstname", "Peter");
contact1.Add("lastname", "Cambel");
contact1.Add("annualincome", 80000);
contact1["jobtitle"] = "Junior Developer";
```

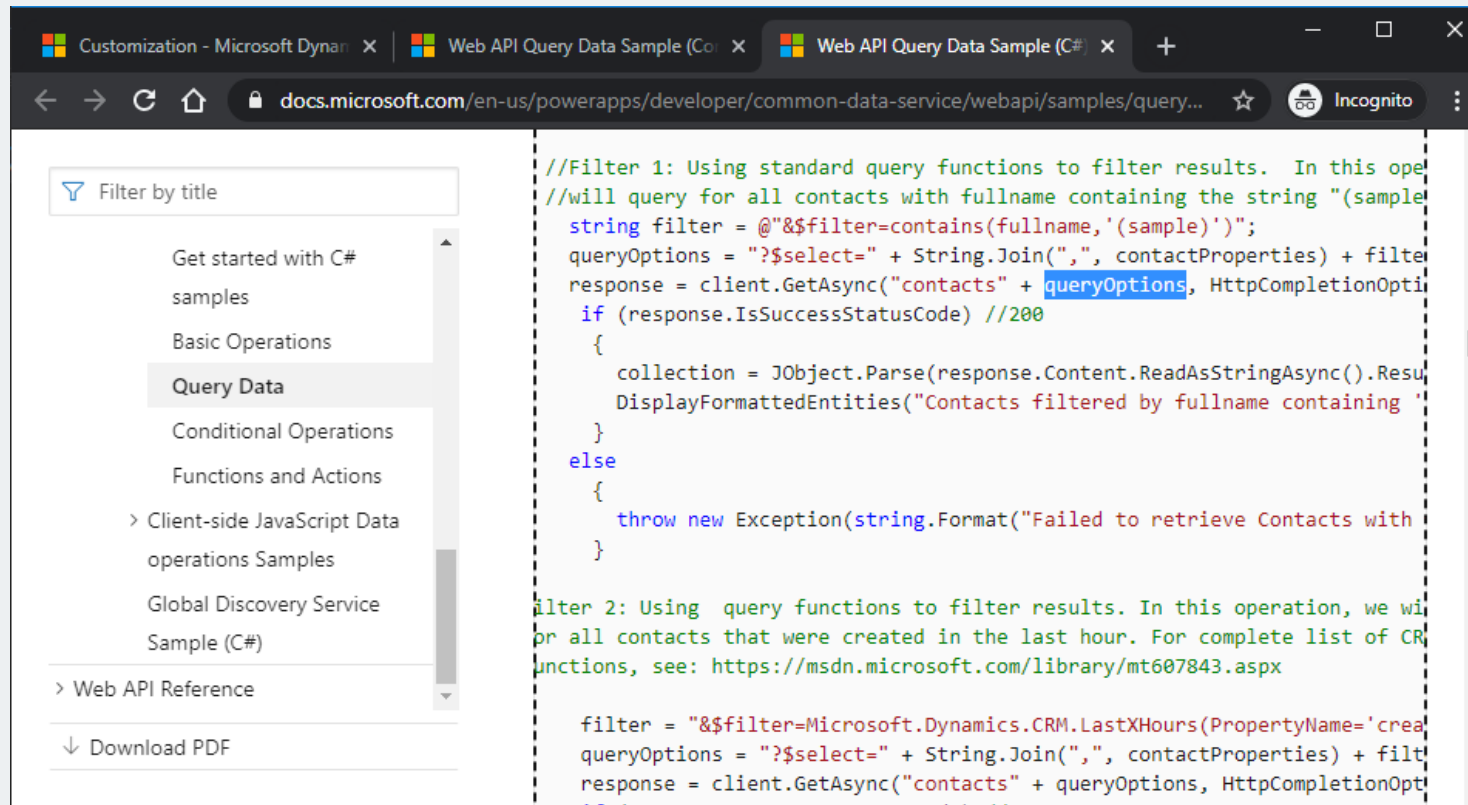
# Microsoft CRM / Dynamics 365 Customer Engagement

**Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Send > Untyped, well JSON (Newtonsoft) typed !**



# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Receive > Query String (not sanitized text)



The screenshot shows a web browser window with the URL `docs.microsoft.com/en-us/powerapps/developer/common-data-service/webapi/samples/query...`. The page content is divided into two main sections:

- Navigation Menu (Left):** A sidebar with a search box labeled "Filter by title" and a list of categories including "Get started with C# samples", "Basic Operations", "Query Data" (highlighted), "Conditional Operations", "Functions and Actions", "Client-side JavaScript Data operations Samples", "Global Discovery Service Sample (C#)", "Web API Reference", and "Download PDF".
- Code Block (Right):** A code editor showing C# code for a REST API query. The code includes comments and logic for filtering results based on a query string. The visible code is:

```
//Filter 1: Using standard query functions to filter results. In this operation, we will query for all contacts with fullname containing the string "(sample)".
string filter = @"&$filter=contains(fullname,'(sample)')";
queryOptions = "?$select=" + String.Join(",", contactProperties) + filter;
response = client.GetAsync("contacts" + queryOptions, HttpCompletionOptions.Default);
if (response.IsSuccessStatusCode) //200
{
    collection = JObject.Parse(response.Content.ReadAsStringAsync().Result);
    DisplayFormattedEntities("Contacts filtered by fullname containing '" + filter + "'");
}
else
{
    throw new Exception(string.Format("Failed to retrieve Contacts with filter: {0}", filter));
}

//Filter 2: Using query functions to filter results. In this operation, we will query for all contacts that were created in the last hour. For complete list of CRM query functions, see: https://msdn.microsoft.com/library/mt607843.aspx
filter = "&$filter=Microsoft.Dynamics.CRM.LastXHours(PropertyName='createdon',XHours=1);";
queryOptions = "?$select=" + String.Join(",", contactProperties) + filter;
response = client.GetAsync("contacts" + queryOptions, HttpCompletionOptions.Default);
```

# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Receive > Query String (not sanitized text)





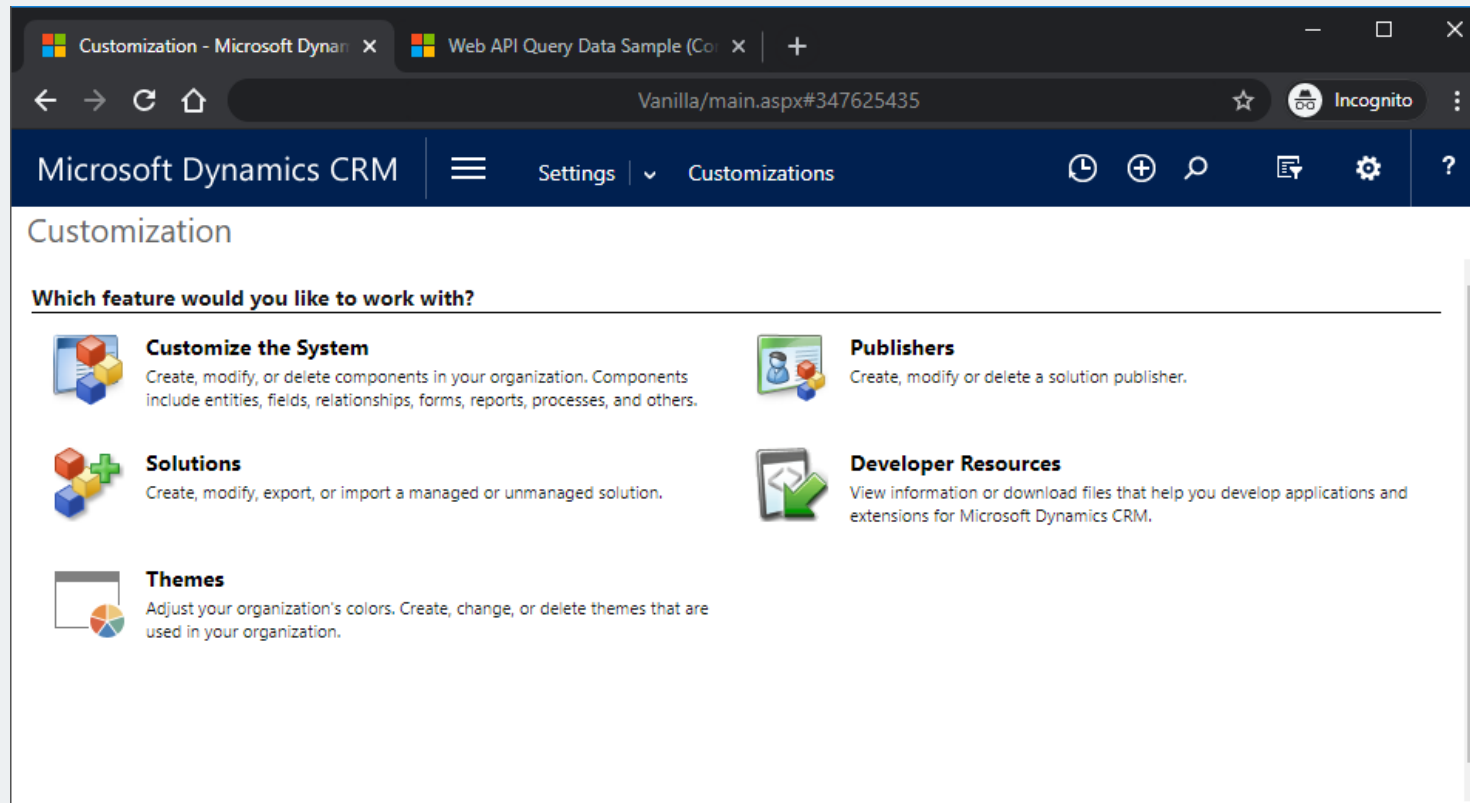
# Microsoft CRM / Dynamics 365 Customer Engagement

**Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Approach (Send + Receive)**

- Pros:
  - **Lightweight**
  - **No dependencies to any assembly**
  - **No dependencies to any platform.** Could easily run on a \*nix box (iff, Mono or .NET Core are ported)  
**Note:** Microsoft recommends to use Json.NET - Newtonsoft library
- Cons:
  - **Not really type safe**
  - **Not sanitized** commands nor queries (CQRS)
  - Very **tedious to work with ...**

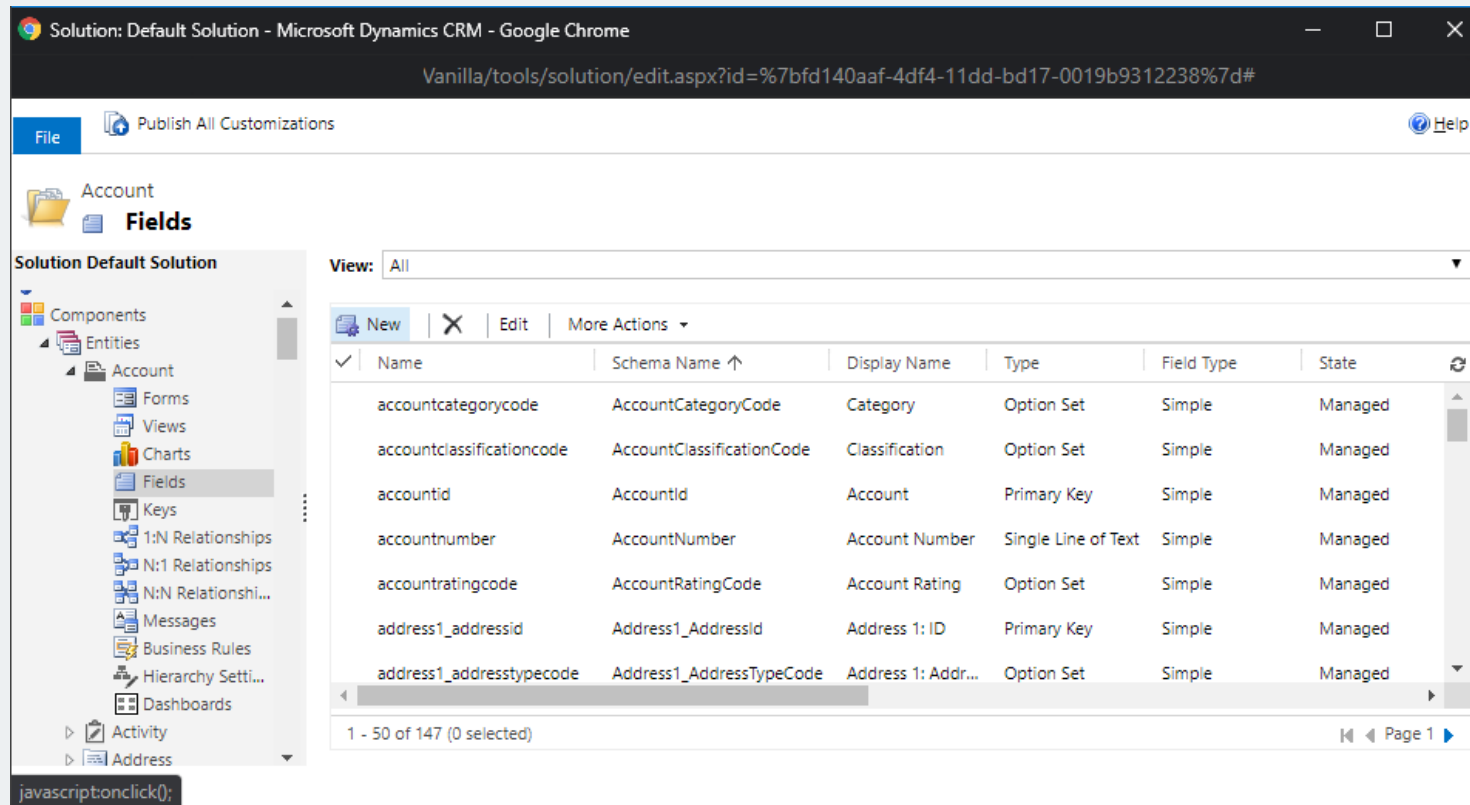
# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Approach > Tedious to work with > Add Lookup



# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Approach > Tedious to work with > Add Lookup



The screenshot shows the Microsoft Dynamics CRM interface in Google Chrome. The browser address bar displays the URL: `Vanilla/tools/solution/edit.aspx?id=%7bfd140aaf-4df4-11dd-bd17-0019b9312238%7d#`. The page title is "Solution: Default Solution - Microsoft Dynamics CRM - Google Chrome".

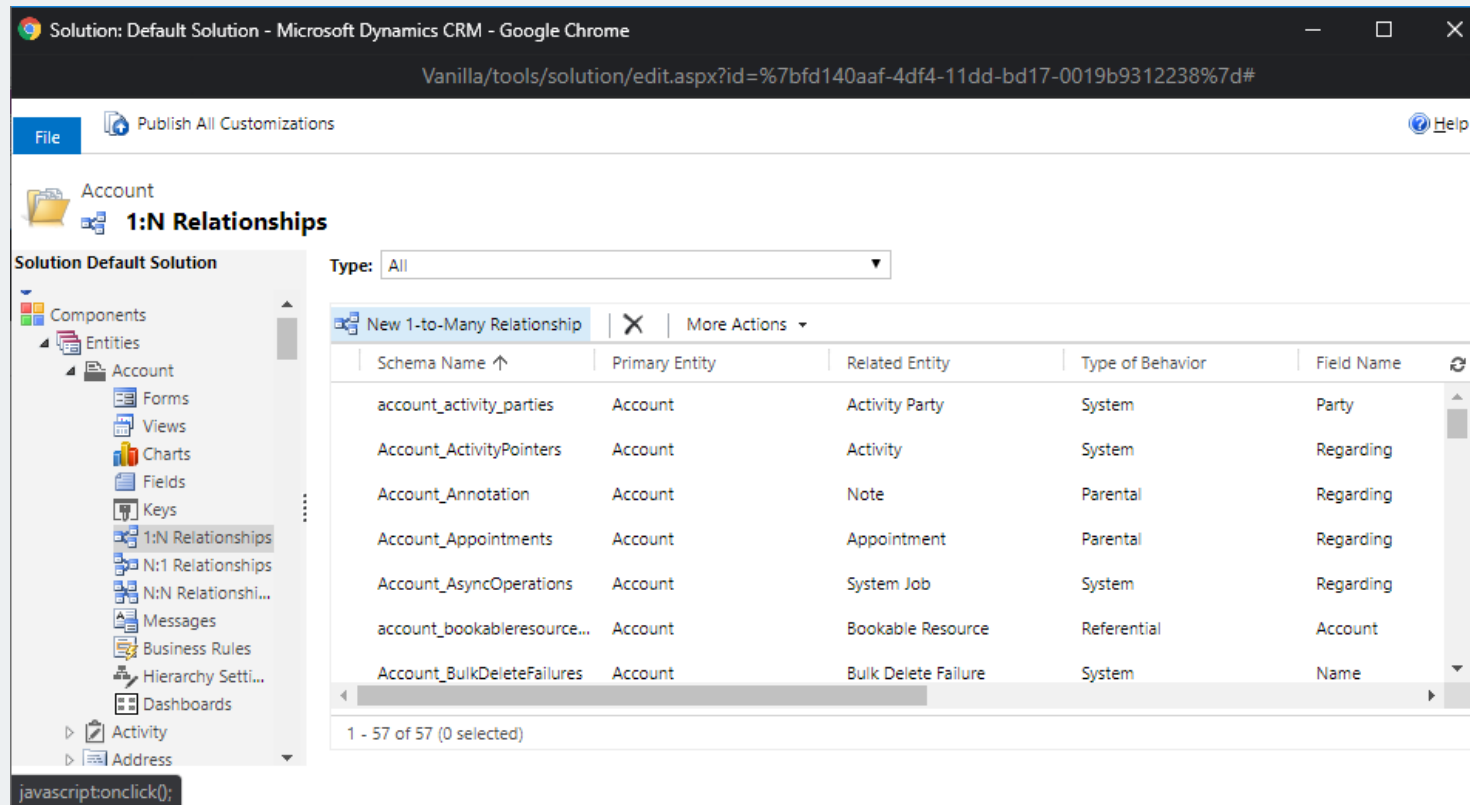
The interface includes a "File" menu with "Publish All Customizations" and a "Help" button. The left sidebar shows the "Account" entity selected under "Fields". The main area displays a table of fields for the "Account" entity.

| Name                      | Schema Name               | Display Name       | Type                | Field Type | State   |
|---------------------------|---------------------------|--------------------|---------------------|------------|---------|
| accountcategorycode       | AccountCategoryCode       | Category           | Option Set          | Simple     | Managed |
| accountclassificationcode | AccountClassificationCode | Classification     | Option Set          | Simple     | Managed |
| accountid                 | AccountId                 | Account            | Primary Key         | Simple     | Managed |
| accountnumber             | AccountNumber             | Account Number     | Single Line of Text | Simple     | Managed |
| accountratingcode         | AccountRatingCode         | Account Rating     | Option Set          | Simple     | Managed |
| address1_addressid        | Address1_AddressId        | Address 1: ID      | Primary Key         | Simple     | Managed |
| address1_addrstypcode     | Address1_AddressTypeCode  | Address 1: Addr... | Option Set          | Simple     | Managed |

At the bottom of the screenshot, there is a small text box containing the code: `javascript:onclick();`

# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Approach > Tedious to work with > Add Lookup



The screenshot shows the Microsoft Dynamics CRM interface in Google Chrome. The browser address bar displays the URL: `Vanilla/tools/solution/edit.aspx?id=%7bfd140aaf-4df4-11dd-bd17-0019b9312238%7d#`. The interface includes a menu bar with 'File' and 'Publish All Customizations', and a 'Help' button. The left sidebar shows the 'Solution Default Solution' tree with 'Account' selected under 'Entities'. The main area displays '1:N Relationships' for the 'Account' entity. A table lists various relationships with columns for Schema Name, Primary Entity, Related Entity, Type of Behavior, and Field Name. The table contains 7 rows of data. At the bottom of the table, it shows '1 - 57 of 57 (0 selected)'. A small code snippet `javascript:onclick();` is visible at the bottom left of the interface.

| Schema Name ↑               | Primary Entity | Related Entity      | Type of Behavior | Field Name |
|-----------------------------|----------------|---------------------|------------------|------------|
| account_activity_parties    | Account        | Activity Party      | System           | Party      |
| Account_ActivityPointers    | Account        | Activity            | System           | Regarding  |
| Account_Annotation          | Account        | Note                | Parental         | Regarding  |
| Account_Appointments        | Account        | Appointment         | Parental         | Regarding  |
| Account_AsyncOperations     | Account        | System Job          | System           | Regarding  |
| account_bookableresource... | Account        | Bookable Resource   | Referential      | Account    |
| Account_BulkDeleteFailures  | Account        | Bulk Delete Failure | System           | Name       |

# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Approach > Tedious to work with > Add Lookup

The screenshot shows the Microsoft Dynamics CRM interface for configuring N:1 Relationships. The browser address bar indicates the URL: `Vanilla/tools/solution/edit.aspx?id=%7bfd140aaf-4df4-11dd-bd17-0019b9312238%7d#`. The page title is "Solution: Default Solution - Microsoft Dynamics CRM - Google Chrome".

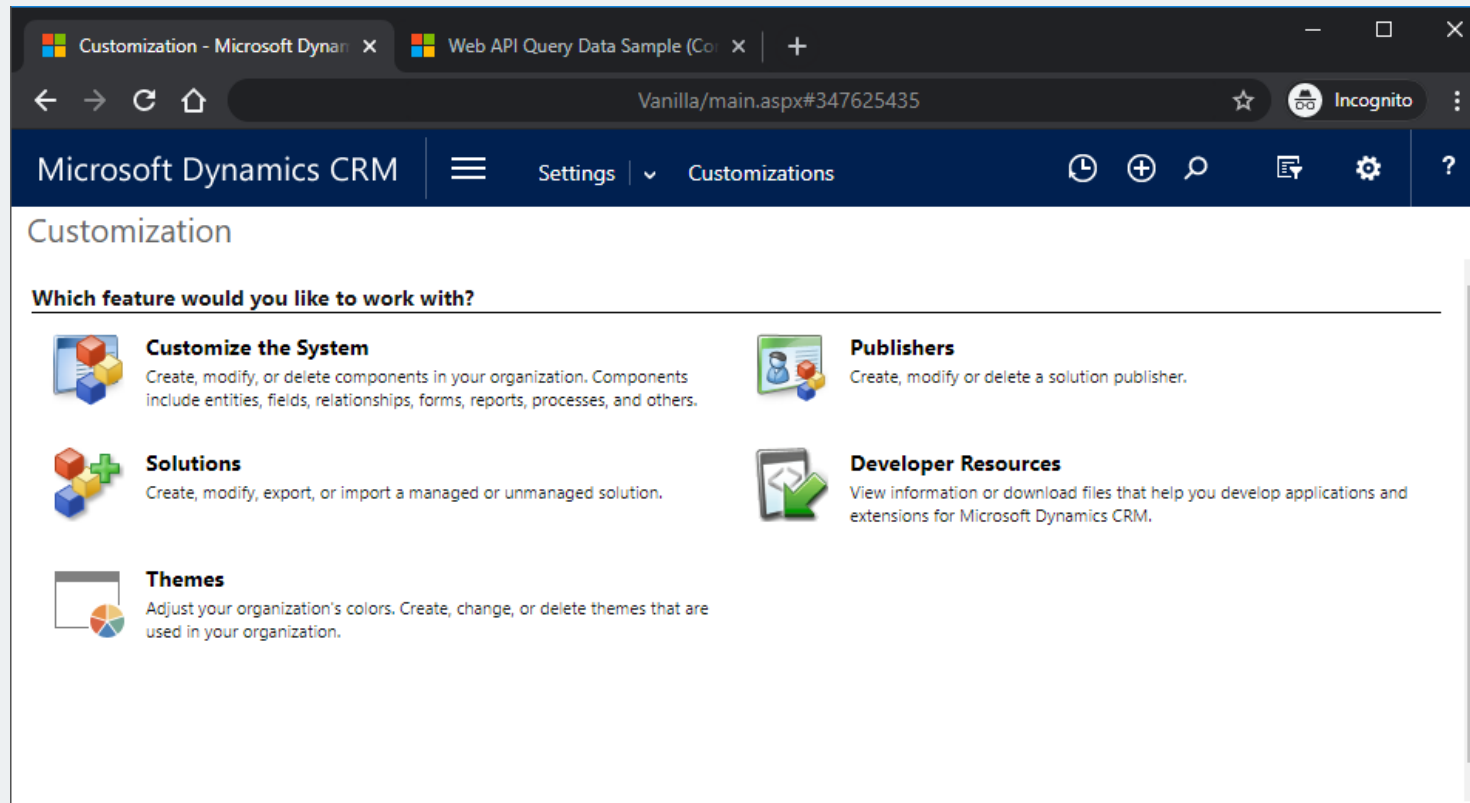
The interface includes a "File" menu with "Publish All Customizations" and a "Help" button. The left sidebar shows the "Solution Default Solution" tree with "Components" > "Entities" > "Account" > "N:1 Relationships" selected. The main content area displays a table of existing relationships.

| Schema Name ↑            | Primary Entity     | Related Entity | Type of Behavior | Field Name ↻       |
|--------------------------|--------------------|----------------|------------------|--------------------|
| account_master_account   | Account            | Account        | System           | Master ID          |
| account_originating_lead | Lead               | Account        | Referential      | Originating Lead   |
| account_parent_account   | Account            | Account        | Parental         | Parent Account     |
| account_primary_contact  | Contact            | Account        | Referential      | Primary Contact    |
| business_unit_accounts   | Business Unit      | Account        | System           | Owning Business    |
| equipment_accounts       | Facility/Equipment | Account        | Referential      | Preferred Facility |
| lk_account_entityimage   | Image Descriptor   | Account        | System           | Entity Image Id    |

At the bottom of the table, it shows "1 - 24 of 24 (0 selected)".

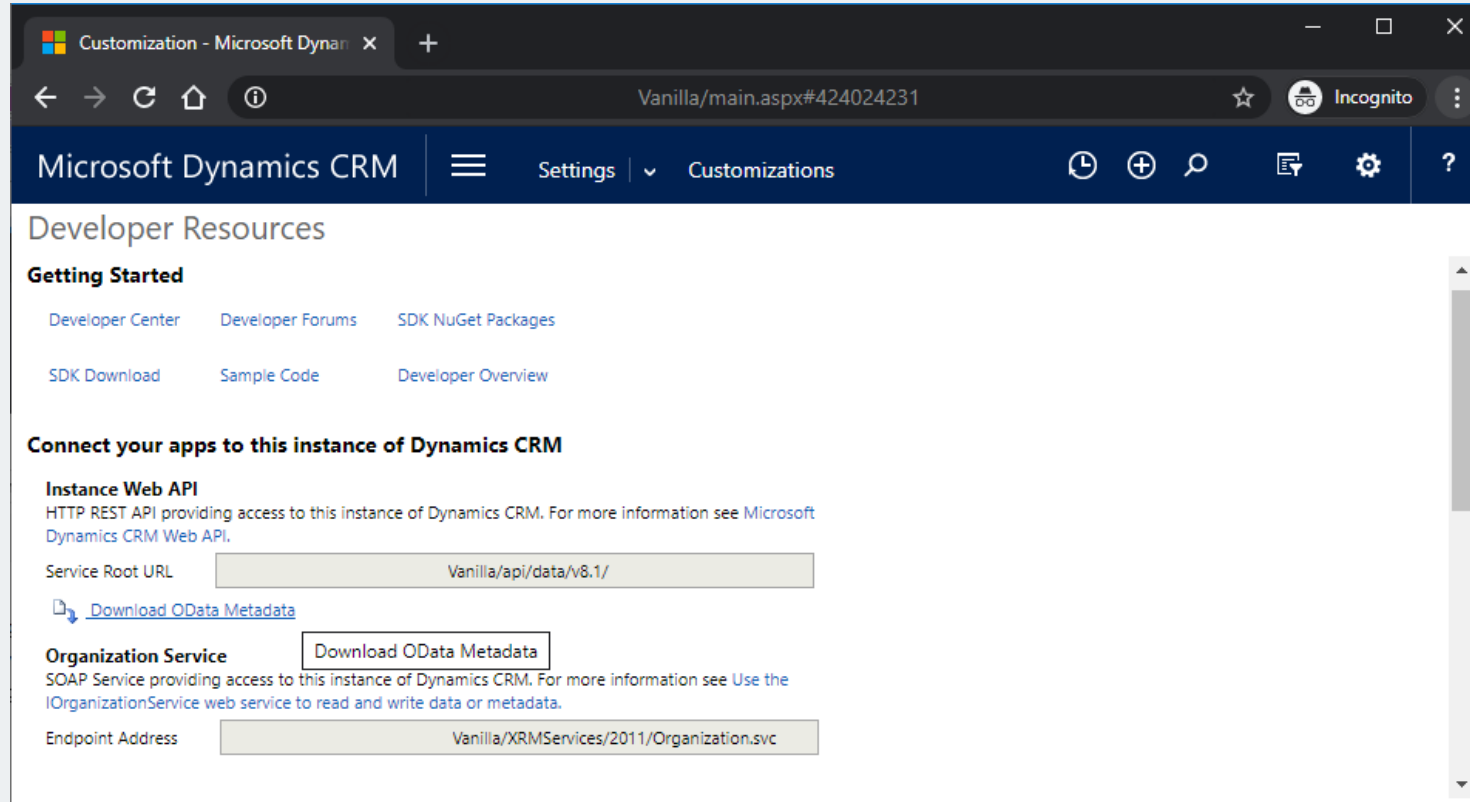
# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Approach > Tedious to work with > Add Lookup



# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Approach > Tedious to work with > Add Lookup



The screenshot shows a web browser window with the title "Customization - Microsoft Dynam" and the URL "Vanilla/main.aspx#424024231". The browser is in Incognito mode. The page content is as follows:

**Microsoft Dynamics CRM** | Settings | Customizations

## Developer Resources

### Getting Started

- [Developer Center](#)
- [Developer Forums](#)
- [SDK NuGet Packages](#)
- [SDK Download](#)
- [Sample Code](#)
- [Developer Overview](#)

### Connect your apps to this instance of Dynamics CRM

#### Instance Web API

HTTP REST API providing access to this instance of Dynamics CRM. For more information see [Microsoft Dynamics CRM Web API](#).

Service Root URL:

[Download OData Metadata](#)

#### Organization Service

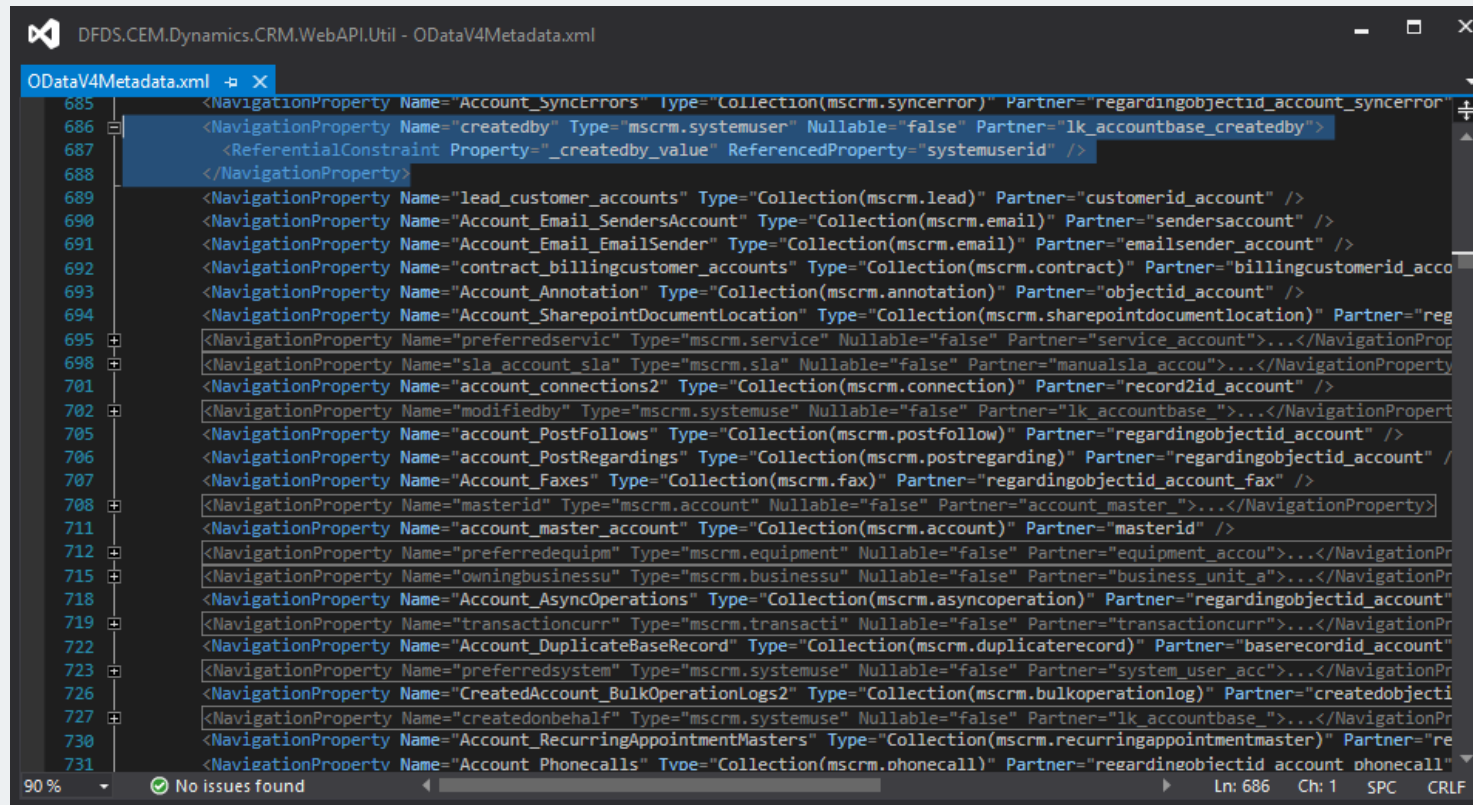
SOAP Service providing access to this instance of Dynamics CRM. For more information see [Use the IOrganizationService web service to read and write data or metadata](#).

Endpoint Address:

A "Download OData Metadata" button is highlighted with a red box in the original image.

# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Approach > Tedious to work with > Add Lookup

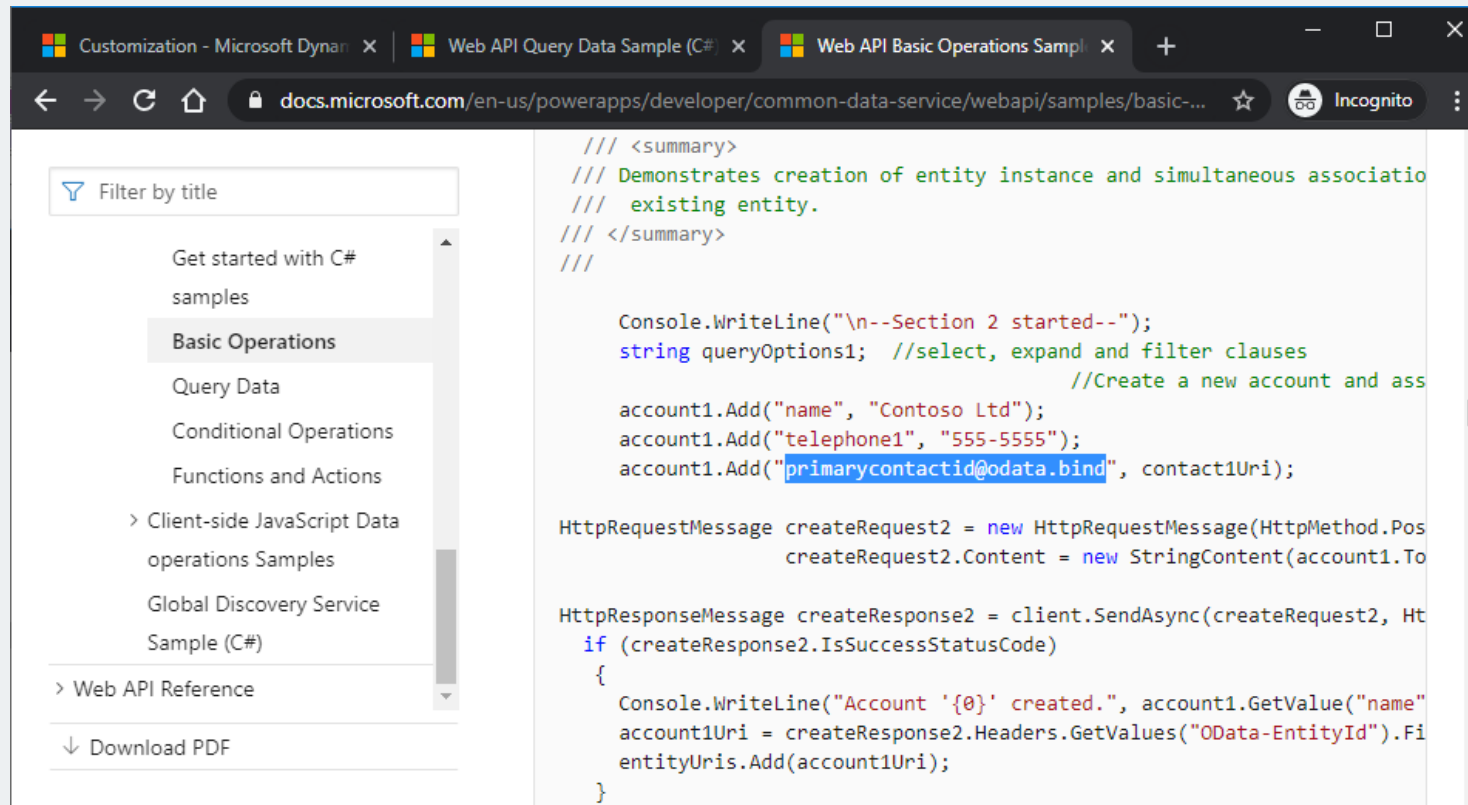


```
DFDS.CEM.Dynamics.CRM.WebAPI.Util - ODataV4Metadata.xml
ODataV4Metadata.xml
685 <NavigationProperty Name="Account_SyncErrors" Type="Collection(mscrm.syncerror)" Partner="regardingobjectid_account_syncerror" />
686 <NavigationProperty Name="createdby" Type="mscrm.systemuser" Nullable="false" Partner="lk_accountbase_createdby">
687   <ReferentialConstraint Property="_createdby_value" ReferencedProperty="systemuserid" />
688 </NavigationProperty>
689 <NavigationProperty Name="lead_customer_accounts" Type="Collection(mscrm.lead)" Partner="customerid_account" />
690 <NavigationProperty Name="Account_Email_SendersAccount" Type="Collection(mscrm.email)" Partner="sendersaccount" />
691 <NavigationProperty Name="Account_Email_EmailSender" Type="Collection(mscrm.email)" Partner="emailsender_account" />
692 <NavigationProperty Name="contract_billingcustomer_accounts" Type="Collection(mscrm.contract)" Partner="billingcustomerid_accou" />
693 <NavigationProperty Name="Account_Annotation" Type="Collection(mscrm.annotation)" Partner="objectid_account" />
694 <NavigationProperty Name="Account_SharepointDocumentLocation" Type="Collection(mscrm.sharepointdocumentlocation)" Partner="reg" />
695 <NavigationProperty Name="preferredservic" Type="mscrm.service" Nullable="false" Partner="service_account">...</NavigationProp
698 <NavigationProperty Name="sla_account_sla" Type="mscrm.sla" Nullable="false" Partner="manualsla_accou">...</NavigationProperty
701 <NavigationProperty Name="account_connections2" Type="Collection(mscrm.connection)" Partner="record2id_account" />
702 <NavigationProperty Name="modifiedby" Type="mscrm.systemuse" Nullable="false" Partner="lk_accountbase_">...</NavigationPropert
705 <NavigationProperty Name="account_PostFollows" Type="Collection(mscrm.postfollow)" Partner="regardingobjectid_account" />
706 <NavigationProperty Name="account_PostRegardings" Type="Collection(mscrm.postregarding)" Partner="regardingobjectid_account" /
707 <NavigationProperty Name="Account_Faxes" Type="Collection(mscrm.fax)" Partner="regardingobjectid_account_fax" />
708 <NavigationProperty Name="masterid" Type="mscrm.account" Nullable="false" Partner="account_master_">...</NavigationProperty>
711 <NavigationProperty Name="account_master_account" Type="Collection(mscrm.account)" Partner="masterid" />
712 <NavigationProperty Name="preferredequipm" Type="mscrm.equipment" Nullable="false" Partner="equipment_accou">...</NavigationPr
715 <NavigationProperty Name="owningbusinessu" Type="mscrm.businessu" Nullable="false" Partner="business_unit_a">...</NavigationPr
718 <NavigationProperty Name="Account_AsyncOperations" Type="Collection(mscrm.asyncoperation)" Partner="regardingobjectid_account" />
719 <NavigationProperty Name="transactioncurr" Type="mscrm.transacti" Nullable="false" Partner="transactioncurr">...</NavigationPr
722 <NavigationProperty Name="Account_DuplicateBaseRecord" Type="Collection(mscrm.duplicaterecord)" Partner="baserecordid_account" />
723 <NavigationProperty Name="preferredsystem" Type="mscrm.systemuse" Nullable="false" Partner="system_user_acc">...</NavigationPr
726 <NavigationProperty Name="CreatedAccount_BulkOperationLogs2" Type="Collection(mscrm.bulkoperationlog)" Partner="createdobjecti
727 <NavigationProperty Name="createdonbehalf" Type="mscrm.systemuse" Nullable="false" Partner="lk_accountbase_">...</NavigationPr
730 <NavigationProperty Name="Account_RecurringAppointmentMasters" Type="Collection(mscrm.recurringappointmentmaster)" Partner="re
731 <NavigationProerty Name="Account_Phonecalls" Tvoe="Collection(mscrm.phonecall)" Partner="regardineobjectid account_phonecall" />
```



# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Approach > Tedious to work with > Add Lookup



```
/// <summary>
/// Demonstrates creation of entity instance and simultaneous associatio
/// existing entity.
/// </summary>
///
    Console.WriteLine("\n--Section 2 started--");
    string queryOptions1; //select, expand and filter clauses
                                //Create a new account and ass
    account1.Add("name", "Contoso Ltd");
    account1.Add("telephone1", "555-5555");
    account1.Add("primarycontactid@odata.bind", contact1Uri);

    HttpRequestMessage createRequest2 = new HttpRequestMessage(HttpMethod.Pos
        createRequest2.Content = new StringContent(account1.To

    HttpResponseMessage createResponse2 = client.SendAsync(createRequest2, Ht
    if (createResponse2.IsSuccessStatusCode)
    {
        Console.WriteLine("Account '{0}' created.", account1.GetValue("name"
        account1Uri = createResponse2.Headers.GetValues("OData-EntityId").Fi
        entityUris.Add(account1Uri);
    }
```

# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > MSDN > Approach > Tedious to work with > Add Lookup

**ARE YOU FORKING KIDDING ME**



# Microsoft CRM / Dynamics 365 Customer Engagement

## Background > Platform > Architecture > REST / WebAPI (JSON) > Delegate A/S > XrmDefinitelyTyped

- **XrmDefinitelyTyped** generates **TypeScript** declaration files based on your Dynamics 365/CRM/xRM solution:
  - <https://github.com/delegateas/XrmDefinitelyTyped/wiki>
- It is the **TypeScript** equivalent of **CrmSvcUtil**, but instead of generating early-bound .NET classes for server-side code, it generates TypeScript interfaces for all your client-side coding.
- Maintained by **Delegate A/S** and accessible through **NuGet**



# Microsoft CRM / Dynamics 365 Customer Engagement

## Background > Platform > Architecture > REST / WebAPI (JSON) > Delegate A/S > XrmDefinitelyTyped

- Pros:
  - **Lightweight**, as generated declaration files will never be part of the final output
  - **Generation of types** for custom xRM data entities created in the CRM application instances
  - **Ideal** to use with **CRM front-end development** instead of JavaScript
- Cons:
  - **Slow**, as TypeScript is transpiled to JavaScript which means that we only have access to single thread:
    - Welcome back to 2001 (Power4 was the first mainstream computer processor with more than one core on a single die)
    - Or welcome to 2020, as the OCaml community would say 😊 (shots fired)
  - **Runtime** infrastructure: **Dependency** to **NPM** to execute outputted JavaScript
  - **Development** setup: **Dependency** to **MS CRM SDK** assemblies (not .NET Core friendly)
    - [https://github.com/delegateas/XrmDefinitelyTyped/blob/gh\\_master/paket.dependencies#L8-L10](https://github.com/delegateas/XrmDefinitelyTyped/blob/gh_master/paket.dependencies#L8-L10)
  - **Buggy** as the WebAPI is pretty complicated to understand and it might be an error to implement all of it in a naive way

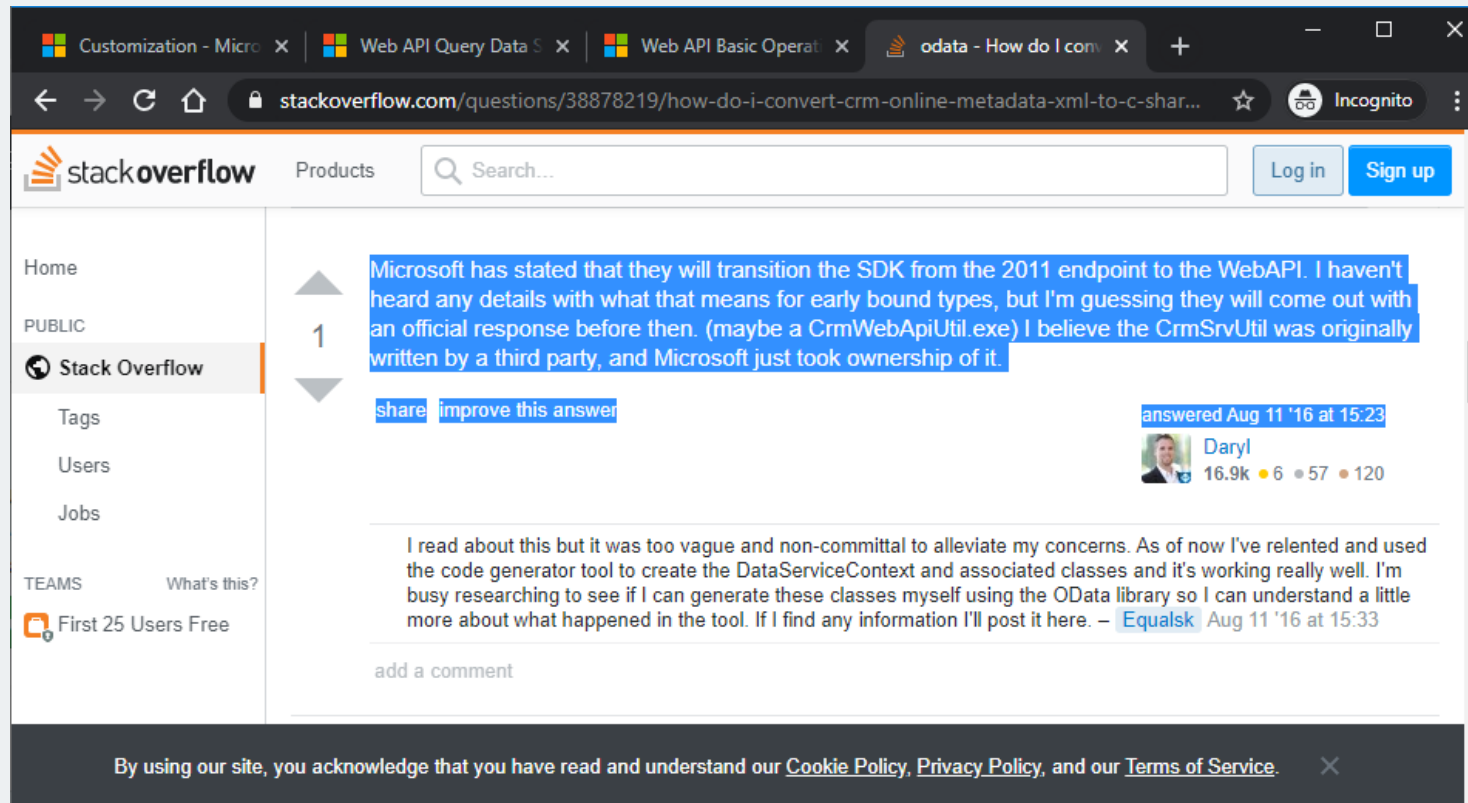
# Microsoft CRM / Dynamics 365 Customer Engagement

## Background > Platform > Architecture > REST / WebAPI (JSON) > Ideally what would we like to have?

- Generate **type-safe C# classes** for xRM data entities of **any CRM application instance**
- Support for **LINQ code**
- Data **transformation from and to JSON** should be done **automatically** (sanitization)
- **Lightweight** size of generated types
- **Concurrency / parallelism**
- Support **.NET Core v.3.1 (LTS)**
  - Runtime infrastructure
  - Development setup
- **Few dependencies**
  - At most, Json.NET - Newtonsoft library, as Microsoft recommends to use it
  - None to the xRM SDKs as they don't support .NET Core
- **No SDK 2011 endpoint**
  - Only **/api/data/v8.1/** and above must be used

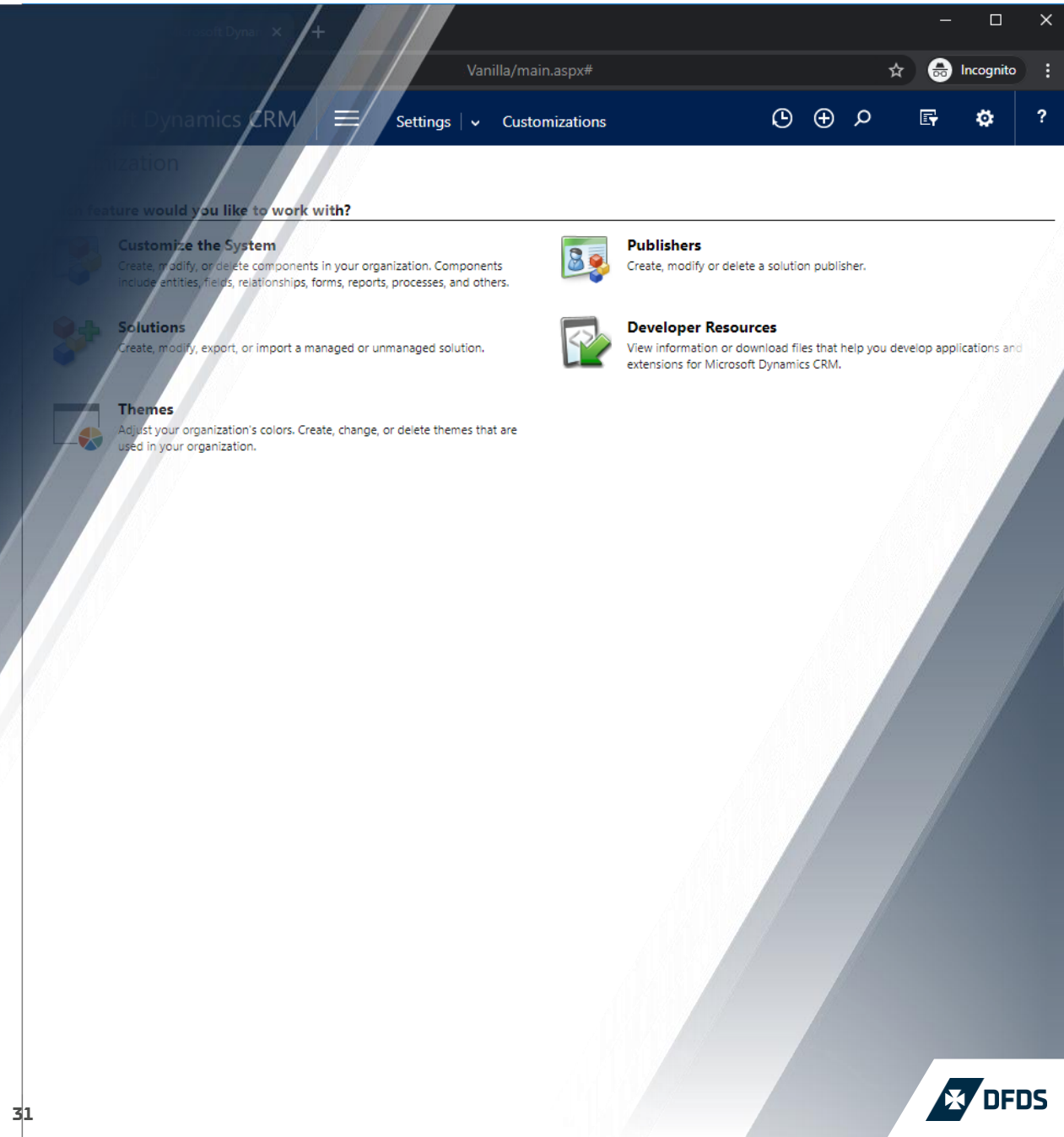
# Microsoft CRM / Dynamics 365 Customer Engagement

Background > Platform > Architecture > REST / WebAPI (JSON) > Ideally what would we like to have? (Missed since 2016)



The screenshot shows a web browser window displaying a Stack Overflow page. The browser's address bar shows the URL: `stackoverflow.com/questions/38878219/how-do-i-convert-crm-online-metadata-xml-to-c-shar...`. The page content includes a question and an answer. The question text is highlighted in blue and reads: "Microsoft has stated that they will transition the SDK from the 2011 endpoint to the WebAPI. I haven't heard any details with what that means for early bound types, but I'm guessing they will come out with an official response before then. (maybe a CrmWebApiUtil.exe) I believe the CrmSrvUtil was originally written by a third party, and Microsoft just took ownership of it." Below the question, there are buttons for "share" and "improve this answer". The answer is by a user named "Daryl", with a profile picture, a reputation of 16.9k, and 6 gold, 57 silver, and 120 bronze badges. The answer text reads: "I read about this but it was too vague and non-committal to alleviate my concerns. As of now I've relented and used the code generator tool to create the DataServiceContext and associated classes and it's working really well. I'm busy researching to see if I can generate these classes myself using the OData library so I can understand a little more about what happened in the tool. If I find any information I'll post it here. - Equalsk Aug 11 '16 at 15:33". At the bottom of the page, there is a dark banner with the text: "By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#)."

# CRM WebAPI Util



The screenshot shows the Microsoft Dynamics CRM interface. The browser address bar displays 'Vanilla/main.aspx#'. The navigation bar includes 'Dynamics CRM', 'Settings', and 'Customizations'. The main content area is titled 'Customization' and features a section 'Which feature would you like to work with?' with four options: 'Customize the System', 'Solutions', 'Themes', 'Publishers', and 'Developer Resources'. Each option includes a brief description of its function.

Microsoft Dynamics CRM

Vanilla/main.aspx#


Settings | Customizations

### Customization

Which feature would you like to work with?

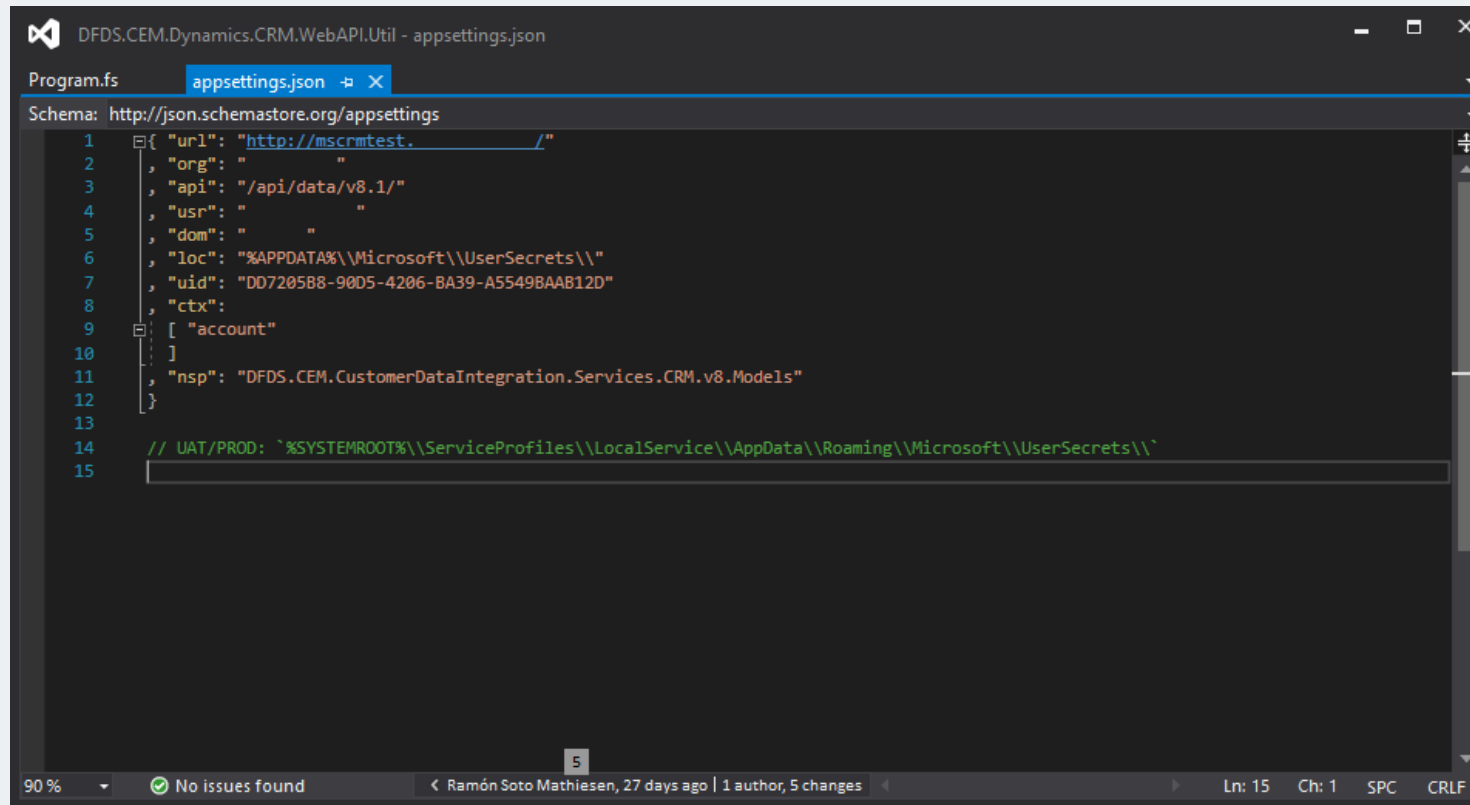
- Customize the System**  
Create, modify, or delete components in your organization. Components include entities, fields, relationships, forms, reports, processes, and others.
- Solutions**  
Create, modify, export, or import a managed or unmanaged solution.
- Themes**  
Adjust your organization's colors. Create, change, or delete themes that are used in your organization.
- Publishers**  
Create, modify or delete a solution publisher.
- Developer Resources**  
View information or download files that help you develop applications and extensions for Microsoft Dynamics CRM.

31



# CRM WebAPI Util > DFDS.CEM.Dynamics.CRM.WebAPI.Util (bin)

## CRM WebAPI Util > appsettings.json



```
DFDS.CEM.Dynamics.CRM.WebAPI.Util - appsettings.json
Program.fs  appsettings.json
Schema: http://json.schemastore.org/appsettings
1  { "url": "http://mscrmtest. /"
2    , "org": ""
3    , "api": "/api/data/v8.1/"
4    , "usr": ""
5    , "dom": ""
6    , "loc": "%APPDATA%\Microsoft\UserSecrets\"
7    , "uid": "DD720588-90D5-4206-BA39-A5549BAAB12D"
8    , "ctx": ""
9    , "account": [
10     ]
11    , "nsp": "DFDS.CEM.CustomerDataIntegration.Services.CRM.v8.Models"
12  }
13
14  // UAT/PROD: `%SYSTEMROOT%\ServiceProfiles\LocalService\AppData\Roaming\Microsoft\UserSecrets\`
15
```

90 % No issues found Ramón Soto Mathiesen, 27 days ago | 1 author, 5 changes Ln: 15 Ch: 1 SPC CRLF



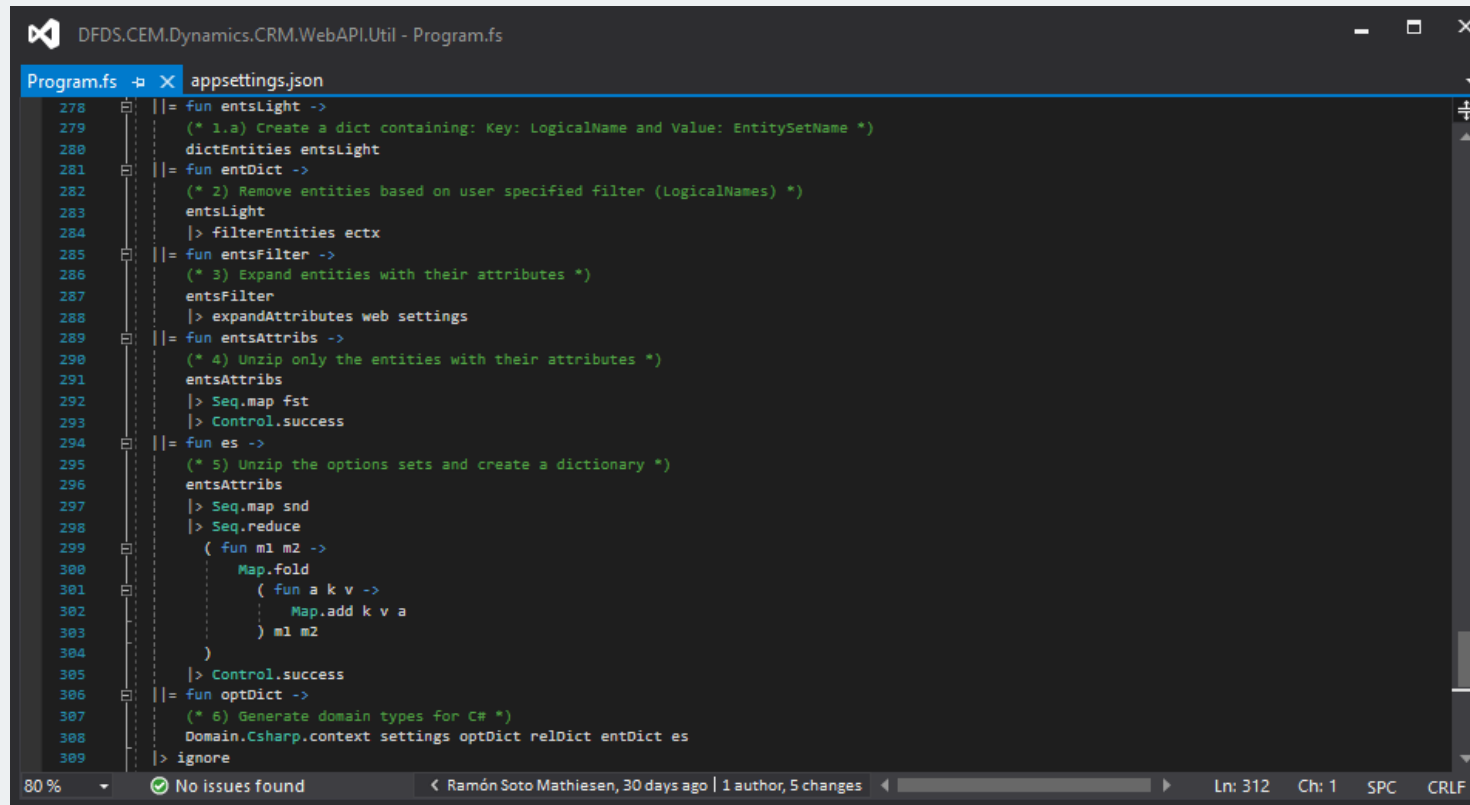
# CRM WebAPI Util > DFDS.CEM.Dynamics.CRM.WebAPI.Util (bin)

## CRM WebAPI Util > Main functionality (1/2)

```
DFDS.CEM.Dynamics.CRM.WebAPI.Util - Program.fs
Program.fs  appsettings.json
248 [EntryPoint]
249 let main argv =
250 (
251     if argv.Length > 0
252     then argv.[0] |> Some |> Control.success
253     else      None |> Control.success
254 )
255 ||= Settings.load
256 ||= fun settings ->
257     use web =
258         HTTP.OData.Client.web
259         ( settings.usr
260           , settings.pwd
261           , settings.dom
262         )
263     (* 0) Retrieve relationships and place them in a dictionary *)
264     settings.url + settings.org + settings.api + rdm
265     |> HTTP.OData.Client.get Error.exn2error web
266     |> Async.RunSynchronously
267     |> Parse.relationships
268 ||= fun relationships ->
269     addLevelToContext settings relationships
270 ||= fun ectx ->
271     dictRelationships ectx relationships
272 ||= fun relDict ->
273     (* 1) Retrieve light list of all (published) entities with: MetadataId, LogicalName, SchemaName and EntitySetName *)
274     settings.url + settings.org + settings.api + edm + edf
275     |> HTTP.OData.Client.get Error.exn2error web
276     |> Async.RunSynchronously
277     |> Parse.entitysets
278 ||= fun entsLight ->
279     (* 1.a) Create a dict containing: Key: LogicalName and Value: EntitySetName *)
```

# CRM WebAPI Util > DFDS.CEM.Dynamics.CRM.WebAPI.Util (bin)

## CRM WebAPI Util > Main functionality (2/2)



```
DFDS.CEM.Dynamics.CRM.WebAPI.Util - Program.fs
Program.fs x appsettings.json
278 ||= fun entsLight ->
279     (* 1.a) Create a dict containing: Key: LogicalName and Value: EntitySetName *)
280     dictEntities entsLight
281 ||= fun entDict ->
282     (* 2) Remove entities based on user specified filter (LogicalNames *)
283     entsLight
284     |> filterEntities ectx
285 ||= fun entsFilter ->
286     (* 3) Expand entities with their attributes *)
287     entsFilter
288     |> expandAttributes web settings
289 ||= fun entsAttribs ->
290     (* 4) Unzip only the entities with their attributes *)
291     entsAttribs
292     |> Seq.map fst
293     |> Control.success
294 ||= fun es ->
295     (* 5) Unzip the options sets and create a dictionary *)
296     entsAttribs
297     |> Seq.map snd
298     |> Seq.reduce
299     ( fun m1 m2 ->
300         Map.fold
301         ( fun a k v ->
302             Map.add k v a
303         ) m1 m2
304     )
305     |> Control.success
306 ||= fun optDict ->
307     (* 6) Generate domain types for C# *)
308     Domain.Csharp.context settings optDict relDict entDict es
309     |> ignore
```

80 % No issues found < Ramón Soto Mathiesen, 30 days ago | 1 author, 5 changes Ln: 312 Ch: 1 SPC CRLF

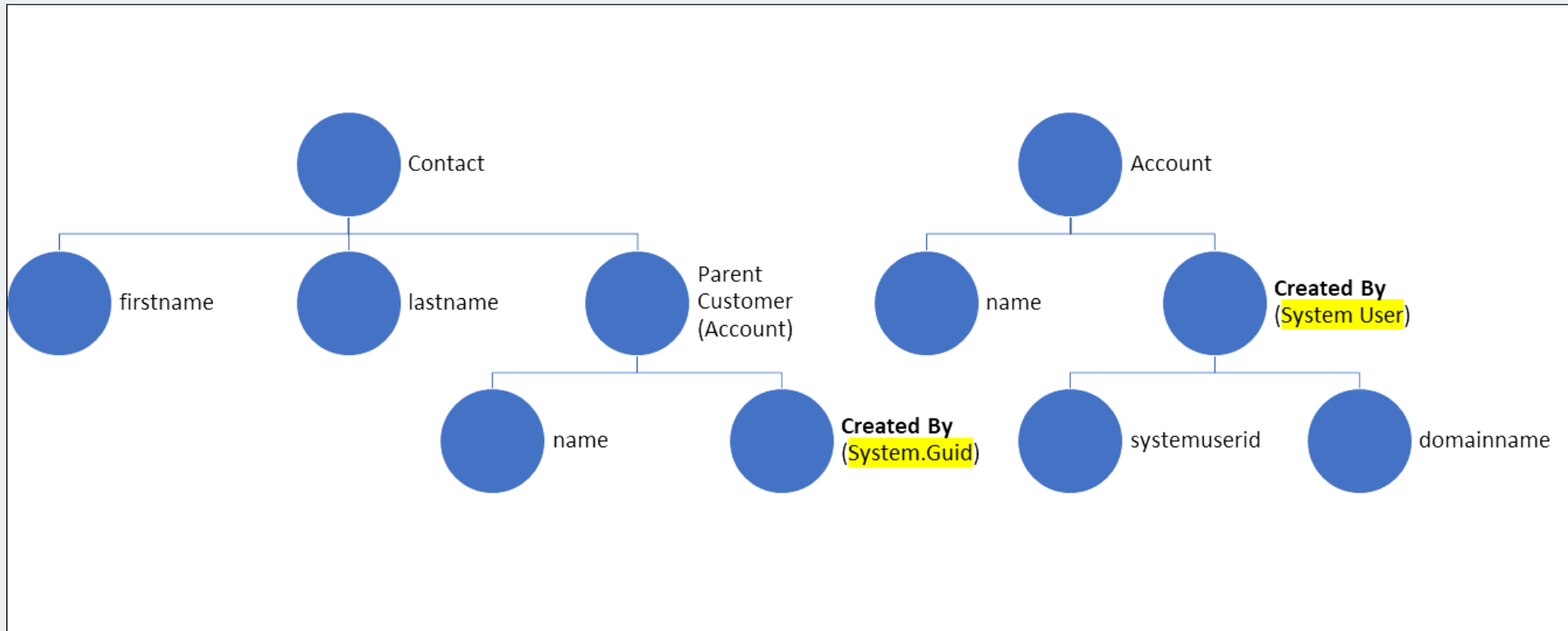
# CRM WebAPI Util > DFDS.CEM.Dynamics.CRM.WebAPI.Util (bin)

## CRM WebAPI Util > Metadata filters

```
DFDS.CEM.Dynamics.CRM.WebAPI.Util - Program.fs
Program.fs  appsettings.json
3 #nowarn "62"
4
5 open System.Net.Http
6
7 open DFDS.CEM
8 open DFDS.CEM.Dynamics.CRM.WebAPI
9
10 open Control.Operator
11
12 let rdm = "RelationshipDefinitions/Microsoft.Dynamics.CRM.OneToManyRelationshipMetadata"
13 let edm = "EntityDefinitions"
14 let edf = "?$select=MetadataId,LogicalName,SchemaName,EntitySetName"
15 let adm = "/Attributes"
16 let osm =
17     (* BooleanAttributeMetadata
18      - https://docs.microsoft.com/en-us/dynamics365/customer-engagement/web-api/booleanattributemetadata
19      *)
20     [| "/Microsoft.Dynamics.CRM.BooleanAttributeMetadata"
21     (* Inherit from EnumAttributeMetadata
22      - https://docs.microsoft.com/en-us/dynamics365/customer-engagement/web-api/enumattributemetadata
23      *)
24     //, "/Microsoft.Dynamics.CRM.EntityNameAttributeMetadata"
25     //, "/Microsoft.Dynamics.CRM.MultiSelectPicklistAttributeMetadata"
26     ; "/Microsoft.Dynamics.CRM.PicklistAttributeMetadata"
27     ; "/Microsoft.Dynamics.CRM.StateAttributeMetadata"
28     ; "/Microsoft.Dynamics.CRM.StatusAttributeMetadata"
29     |]
30 let ose = "?$expand=GlobalOptionSet,OptionSet"
31
```

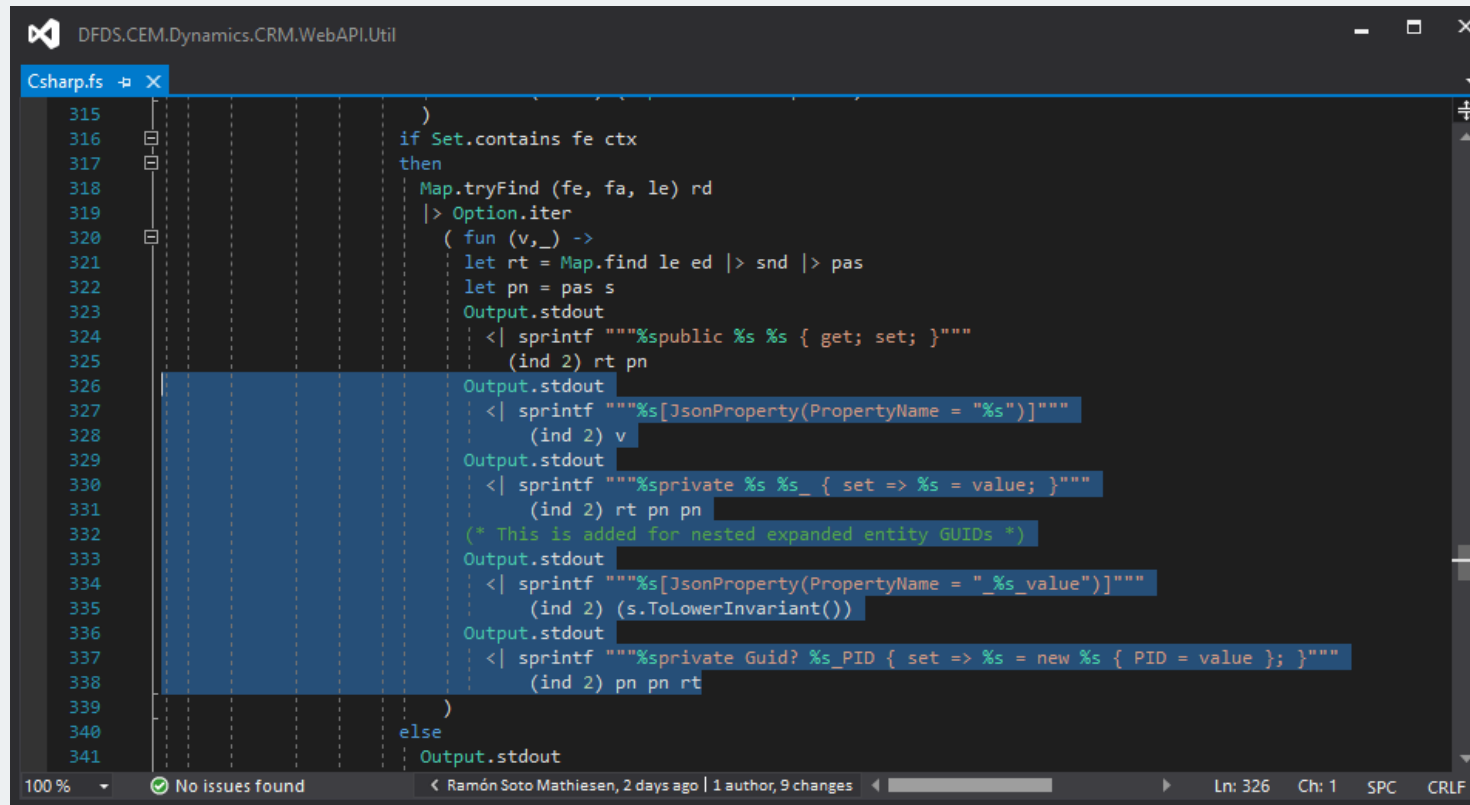
# CRM WebAPI Util > DFDS.CEM.Dynamics.CRM.WebAPI.Util (bin)

## CRM WebAPI Util > WebAPI Sum types (1/3)



# CRM WebAPI Util > DFDS.CEM.Dynamics.CRM.WebAPI.Util (bin)

## CRM WebAPI Util > WebAPI Sum types (2/3) (Json.NET - Newtonsoft and OOPD encapsulation to the rescue)

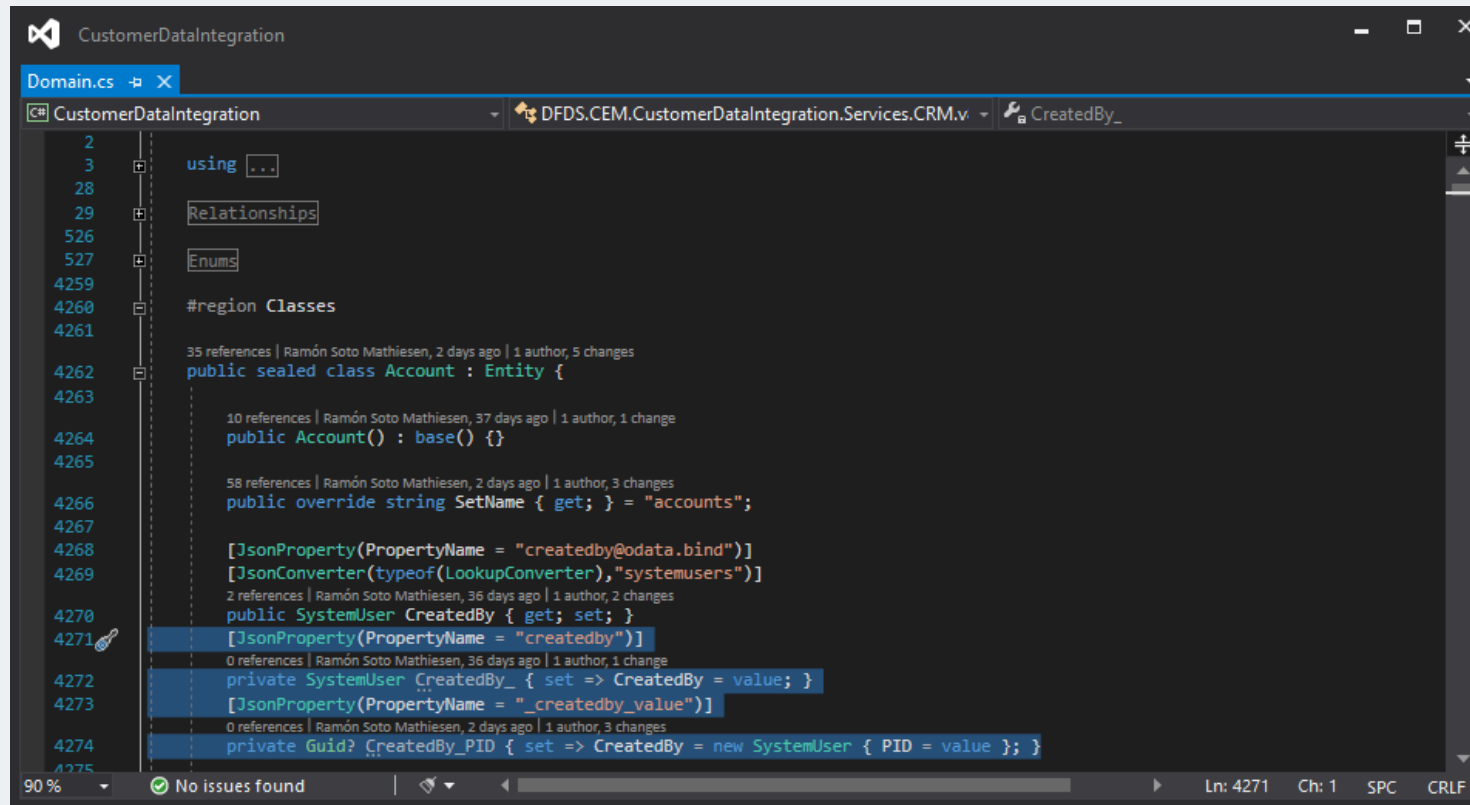


```
DFDS.CEM.Dynamics.CRM.WebAPI.Util
Csharp.fs
315 )
316 if Set.contains fe ctx
317 then
318     Map.tryFind (fe, fa, le) rd
319     |> Option.iter
320     ( fun (v,_) ->
321         let rt = Map.find le ed |> snd |> pas
322         let pn = pas s
323         Output.stdout
324             <| sprintf ""public %s %s { get; set; }""
325             (ind 2) rt pn
326         Output.stdout
327             <| sprintf ""%s[JsonProperty(PropertyName = "%s")]""
328             (ind 2) v
329         Output.stdout
330             <| sprintf ""private %s %s_ { set => %s = value; }""
331             (ind 2) rt pn pn
332         (* This is added for nested expanded entity GUIDs *)
333         Output.stdout
334             <| sprintf ""%s[JsonProperty(PropertyName = "%s_value")]""
335             (ind 2) (s.ToLowerInvariant())
336         Output.stdout
337             <| sprintf ""private Guid? %s_PID { set => %s = new %s { PID = value; }""
338             (ind 2) pn pn rt
339     )
340 else
341     Output.stdout
```

100 % No issues found < Ramón Soto Mathiesen, 2 days ago | 1 author, 9 changes > Ln: 326 Ch: 1 SPC CRLF

# CRM WebAPI Util > DFDS.CEM.Dynamics.CRM.WebAPI.Util (bin)

## CRM WebAPI Util > WebAPI Sum types (3/3) (Json.NET - Newtonsoft and OOPD encapsulation to the rescue)

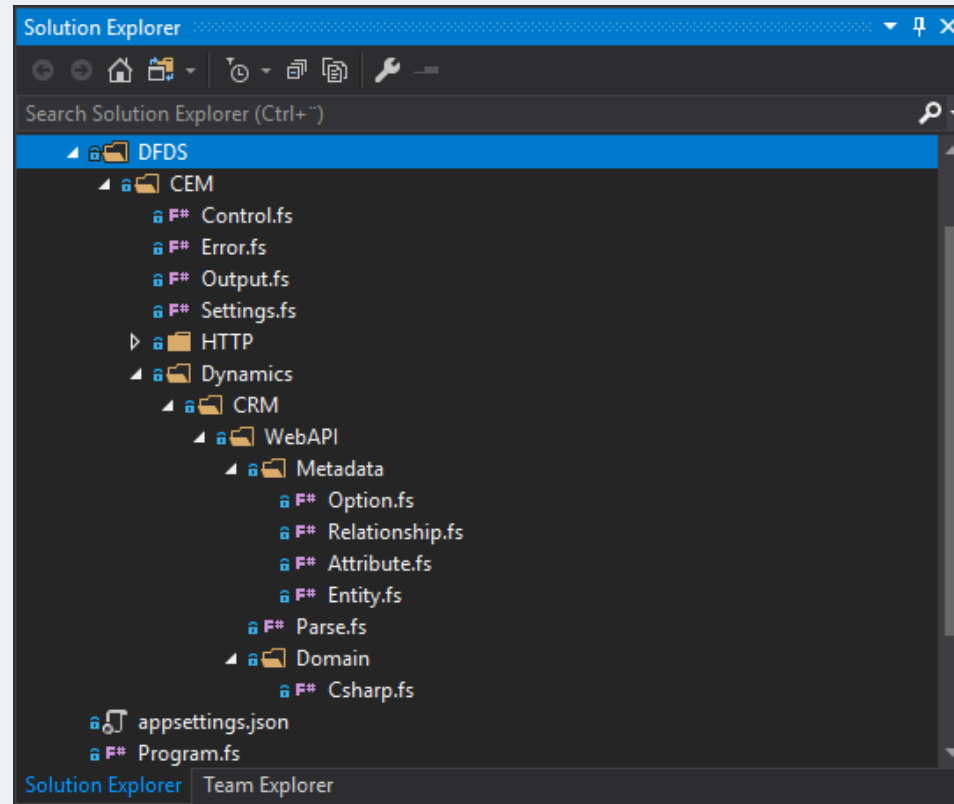


```
CustomerDataIntegration
Domain.cs
CustomerDataIntegration
DFDS.CEM.CustomerDataIntegration.Services.CRM.v
CreatedBy_

2
3
28
29 Relationships
526
527 Enums
4259
4260 #region Classes
4261
35 references | Ramón Soto Mathiesen, 2 days ago | 1 author, 5 changes
4262 public sealed class Account : Entity {
4263
10 references | Ramón Soto Mathiesen, 37 days ago | 1 author, 1 change
4264 public Account() : base() {}
4265
58 references | Ramón Soto Mathiesen, 2 days ago | 1 author, 3 changes
4266 public override string SetName { get; } = "accounts";
4267
[JsonProperty(PropertyName = "createdby@odata.bind")]
[JsonConverter(typeof(LookupConverter), "systemusers")]
2 references | Ramón Soto Mathiesen, 36 days ago | 1 author, 2 changes
4268 public SystemUser CreatedBy { get; set; }
4269 [JsonProperty(PropertyName = "createdby")]
4270 private SystemUser CreatedBy_ { set => CreatedBy = value; }
4271 [JsonProperty(PropertyName = "_createdby_value")]
0 references | Ramón Soto Mathiesen, 2 days ago | 1 author, 3 changes
4272 private Guid? CreatedBy_PID { set => CreatedBy = new SystemUser { PID = value; } }
4273
4274
4275
90% No issues found Ln: 4271 Ch: 1 SPC CRLF
```

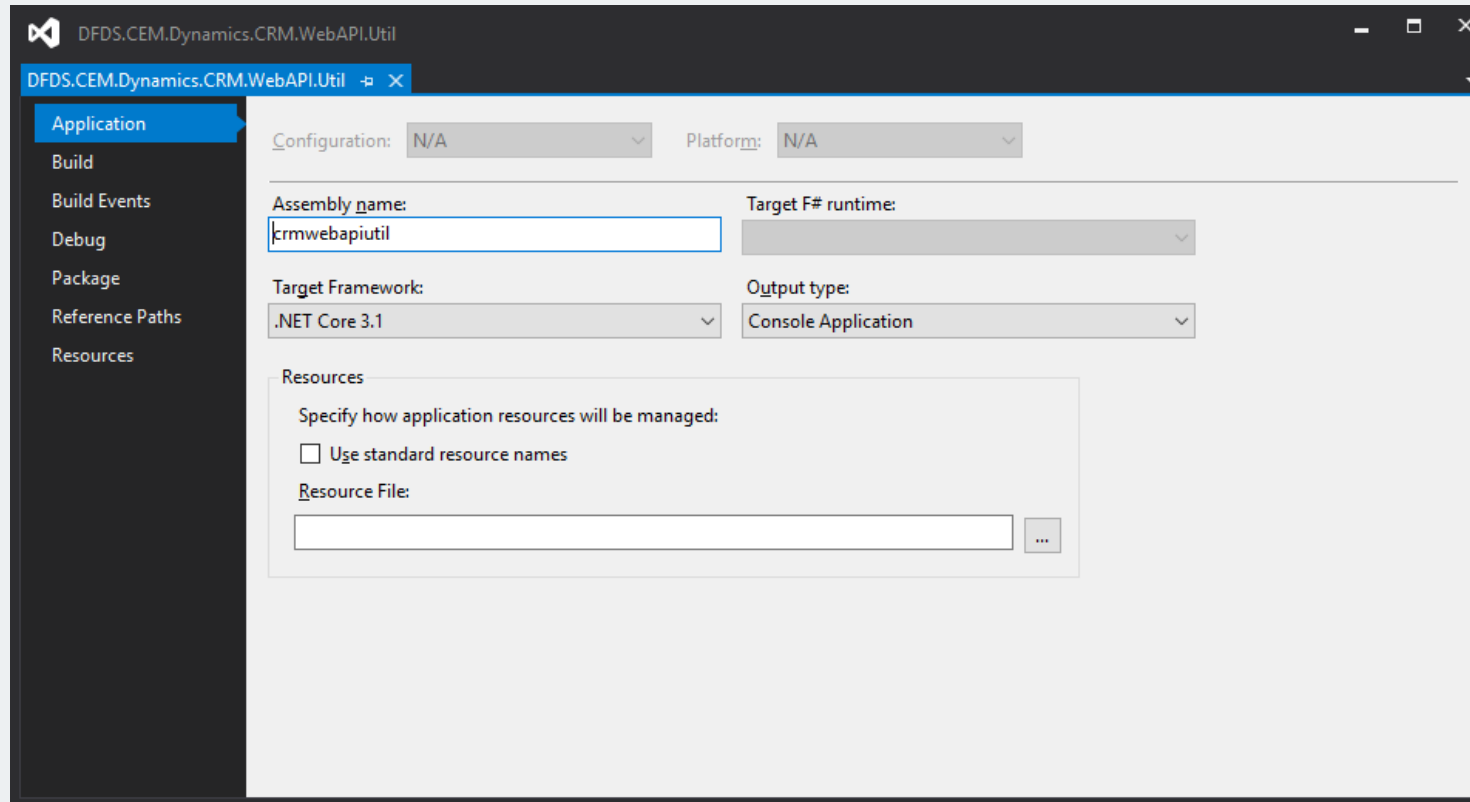
# CRM WebAPI Util > DFDS.CEM.Dynamics.CRM.WebAPI.Util (bin)

## CRM WebAPI Util > (simple) Visual Studio (F#) Project



# CRM WebAPI Util > DFDS.CEM.Dynamics.CRM.WebAPI.Util (bin)

## CRM WebAPI Util > (simple) Visual Studio (F#) Project > .NET Core v.3.1 (LTS)



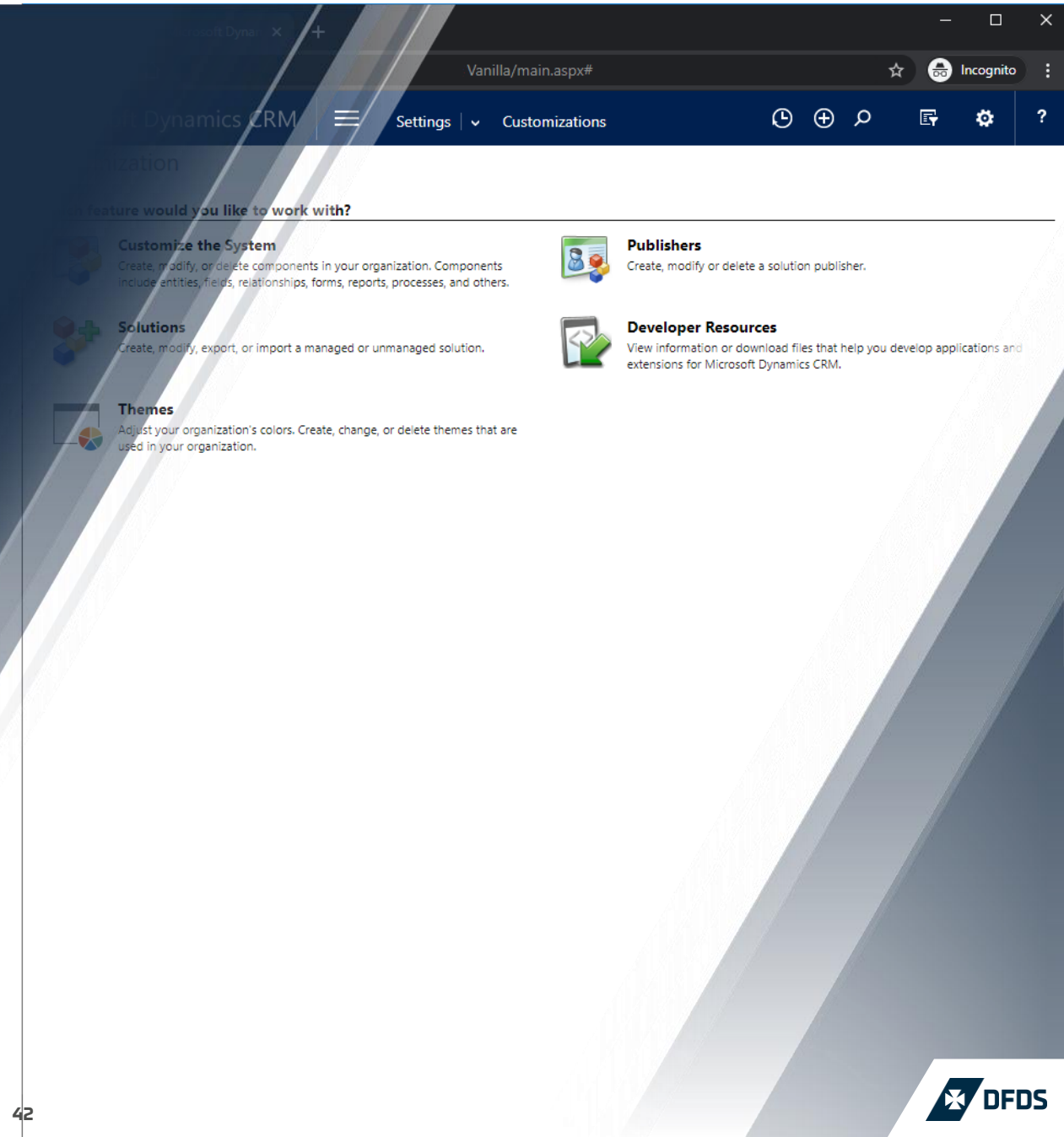


# CRM WebAPI Util > DFDS.CEM.Dynamics.CRM.WebAPI.Util (bin)

## Background > Platform > Architecture > REST / WebAPI (JSON) > Ideally what would we like to have?

- ✓ Generate **type-safe C# classes** for xRM data entities of **any CRM application instance**
  - Support for **LINQ code**
- ✓ Data **transformation from and to JSON** should be done **automatically** (sanitization)
- ✓ **Lightweight** size of generated types
  - **Concurrency / parallelism**
- ✓ Support **.NET Core v.3.1 (LTS)**
  - Runtime infrastructure
  - Development setup
- ✓ **Few dependencies**
  - At most, Json.NET - Newtonsoft library, as Microsoft recommends to use it
  - None to the xRM SDKs as they don't support .NET Core
- ✓ **No SDK 2011 endpoint**
  - Only **/api/data/v8.1/** and above must be used

# LINQ Provider



The screenshot shows the Microsoft Dynamics CRM interface in an Incognito browser window. The address bar shows 'Vanilla/main.aspx#'. The navigation bar includes 'Microsoft Dynamics CRM', 'Settings', and 'Customizations'. The main content area is titled 'Customization' and features a section 'Which feature would you like to work with?' with four options: 'Customize the System', 'Solutions', 'Themes', 'Publishers', and 'Developer Resources'. Each option includes a brief description of its function.

Microsoft Dynamics CRM

Vanilla/main.aspx#


Settings | Customizations

### Customization

Which feature would you like to work with?

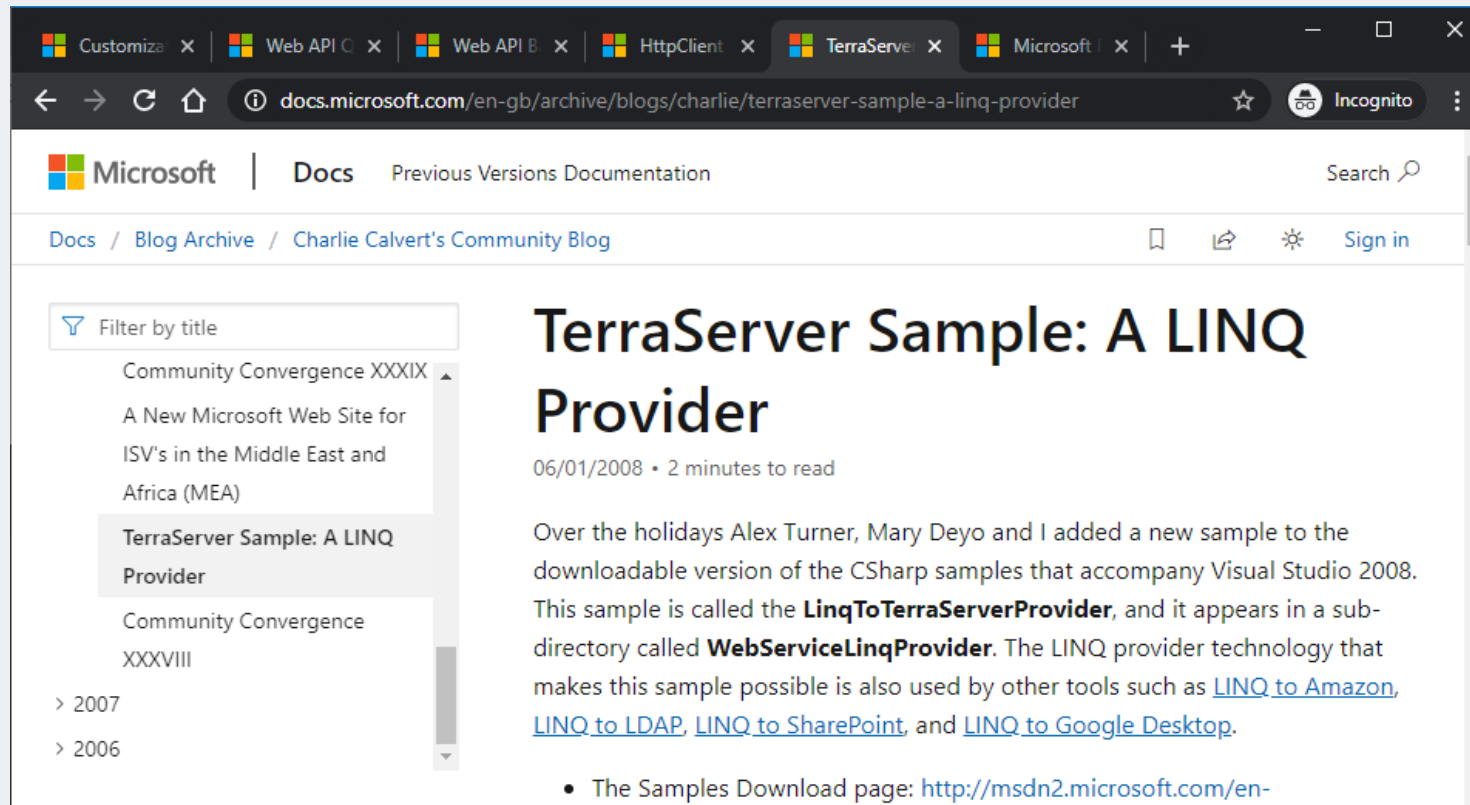
- Customize the System**  
Create, modify, or delete components in your organization. Components include entities, fields, relationships, forms, reports, processes, and others.
- Solutions**  
Create, modify, export, or import a managed or unmanaged solution.
- Themes**  
Adjust your organization's colors. Create, change, or delete themes that are used in your organization.
- Publishers**  
Create, modify or delete a solution publisher.
- Developer Resources**  
View information or download files that help you develop applications and extensions for Microsoft Dynamics CRM.

42



# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

## LINQ Provider > Based on Charlie Calvert's TerraServer Sample



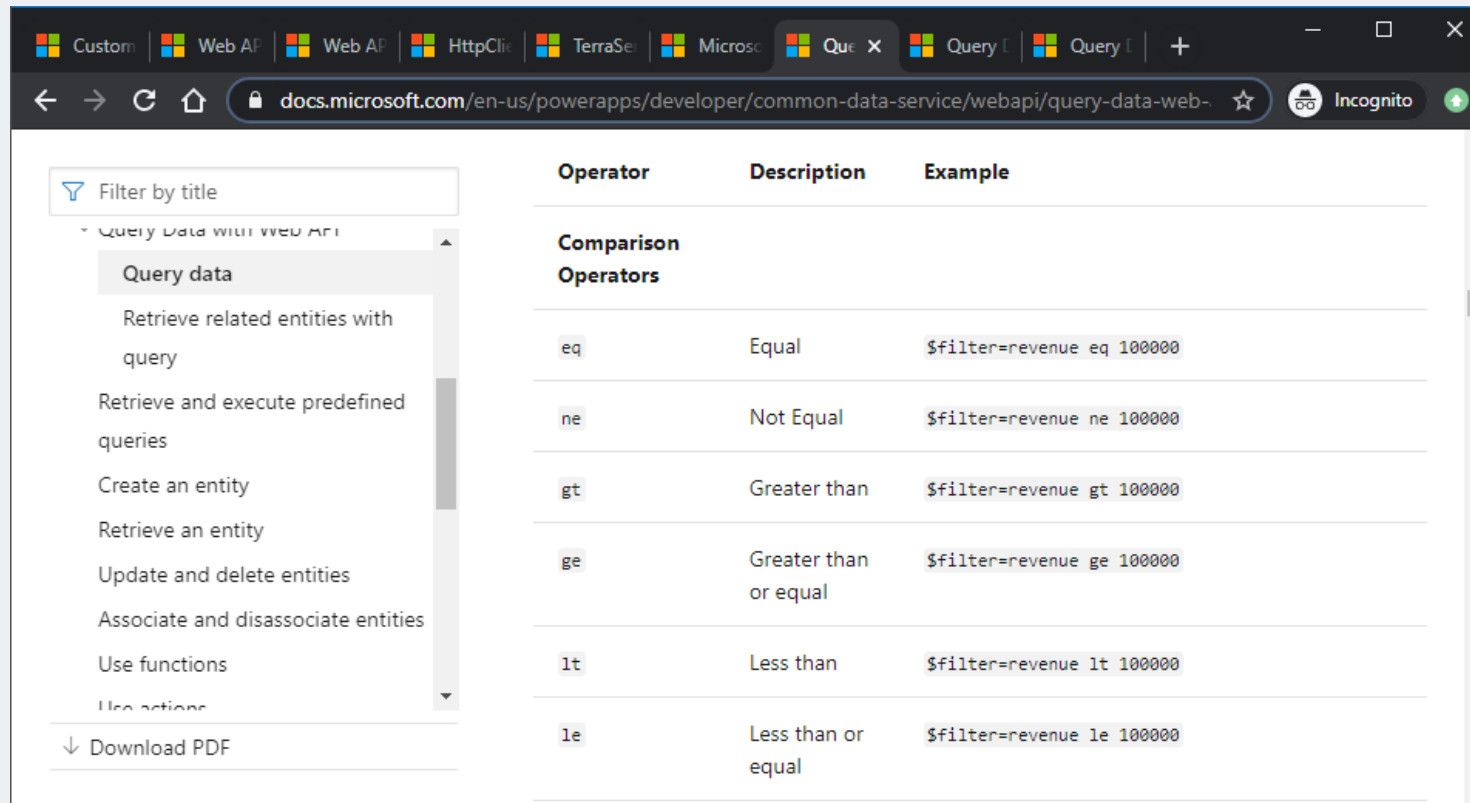
# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

## LINQ Provider > Based on Charlie Calvert's TerraServer Sample (Open Source)



# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

## LINQ Provider > Queries > Supported operations



The screenshot shows a web browser window displaying a page from docs.microsoft.com. The page title is "Query Data with Web API". A navigation menu on the left is open, showing "Query data" selected. The main content area contains a table titled "Comparison Operators" with the following data:

| Operator | Description           | Example                                 |
|----------|-----------------------|---|
| eq       | Equal                 | <code>\$filter=revenue eq 100000</code> |
| ne       | Not Equal             | <code>\$filter=revenue ne 100000</code> |
| gt       | Greater than          | <code>\$filter=revenue gt 100000</code> |
| ge       | Greater than or equal | <code>\$filter=revenue ge 100000</code> |
| lt       | Less than             | <code>\$filter=revenue lt 100000</code> |
| le       | Less than or equal    | <code>\$filter=revenue le 100000</code> |

# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

## LINQ Provider > Queries > Supported operations

Filter by title

- Query Data with Web API
  - Query data
  - Retrieve related entities with query
  - Retrieve and execute predefined queries
  - Create an entity
  - Retrieve an entity
  - Update and delete entities
  - Associate and disassociate entities
  - Use functions
  - Use actions
- Download PDF

### Logical Operators

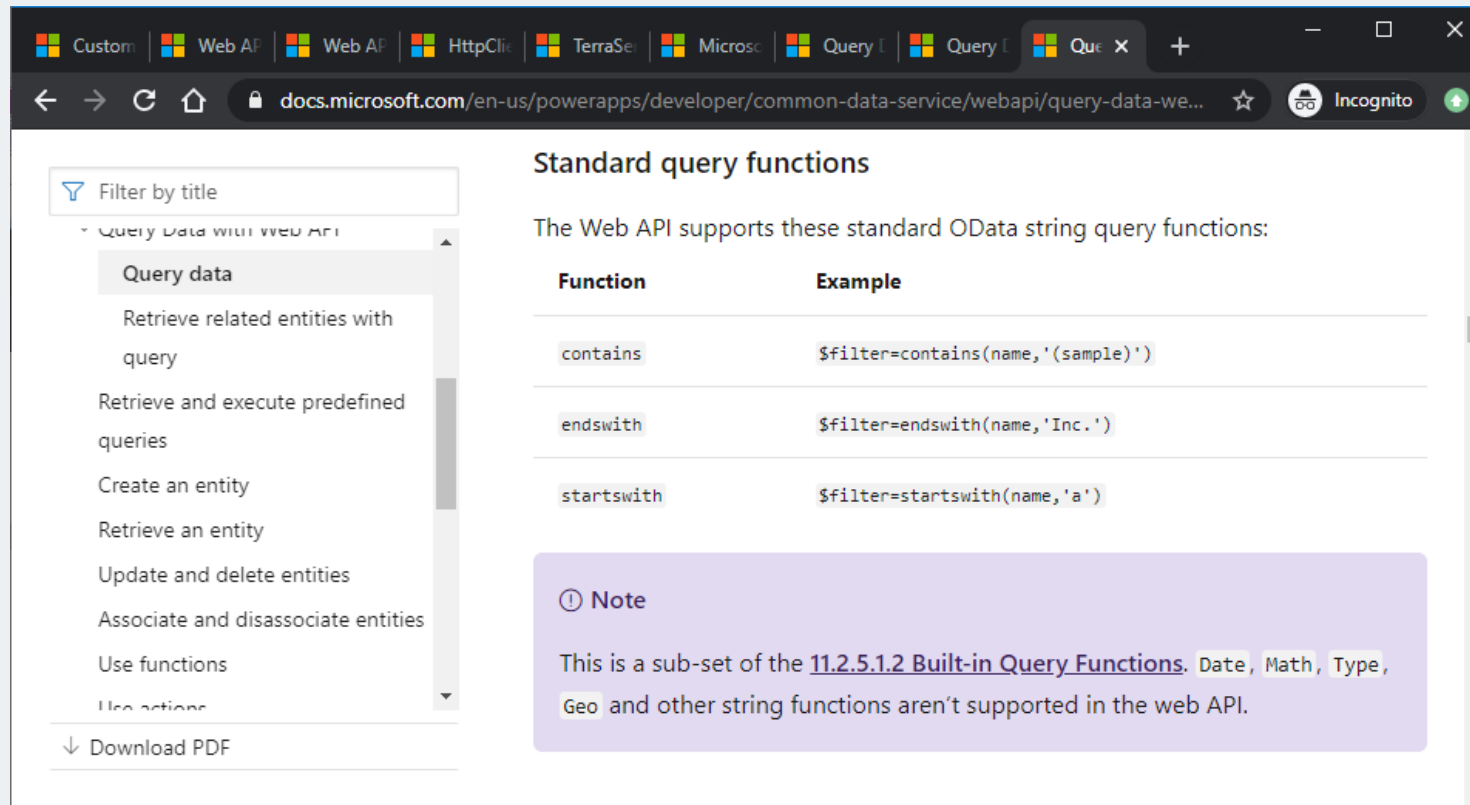
|     |                  |  |
|-----|------------------|--|
| and | Logical and      | <code>\$filter=revenue lt 100000 and revenue gt 2000</code>              |
| or  | Logical or       | <code>\$filter=contains(name,'(sample)') or contains(name,'test')</code> |
| not | Logical negation | <code>\$filter=not contains(name,'sample')</code>                        |

### Grouping Operators

|     |                     |   |
|-----|---------------------|---|
| ( ) | Precedence grouping | <code>(contains(name,'sample') or contains(name,'test')) and revenue gt 5000</code> |
|-----|---------------------|---|

# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

## LINQ Provider > Queries > Supported operations



Standard query functions

The Web API supports these standard OData string query functions:

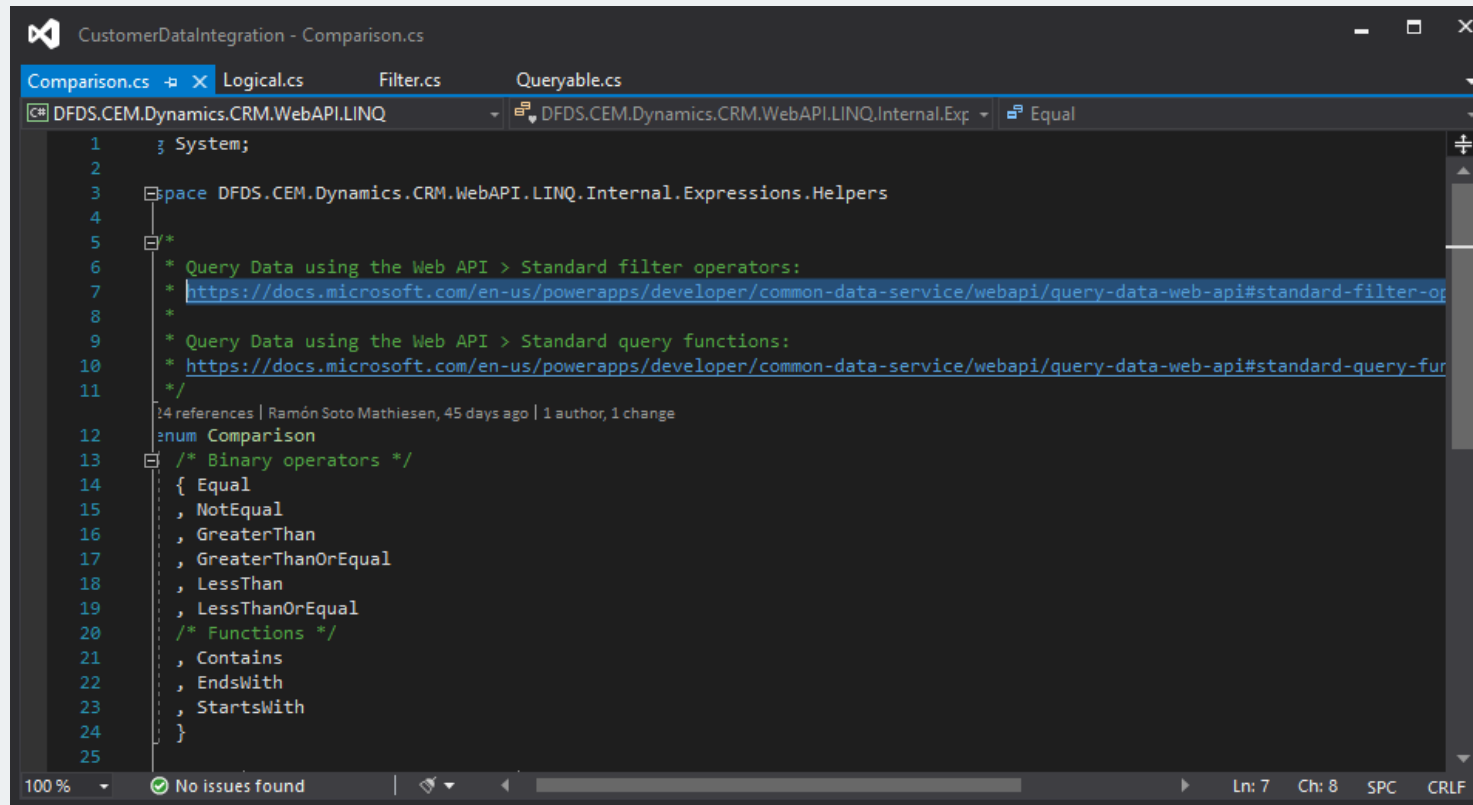
| Function                | Example   |
|-------------------------|---|
| <code>contains</code>   | <code>\$filter=contains(name,'(sample)')</code> |
| <code>endswith</code>   | <code>\$filter=endswith(name,'Inc.')</code>     |
| <code>startswith</code> | <code>\$filter=startswith(name,'a')</code>      |

**Note**

This is a sub-set of the [11.2.5.1.2 Built-in Query Functions](#). `Date`, `Math`, `Type`, `Geo` and other string functions aren't supported in the web API.

# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

## LINQ Provider > Queries > Supported operations

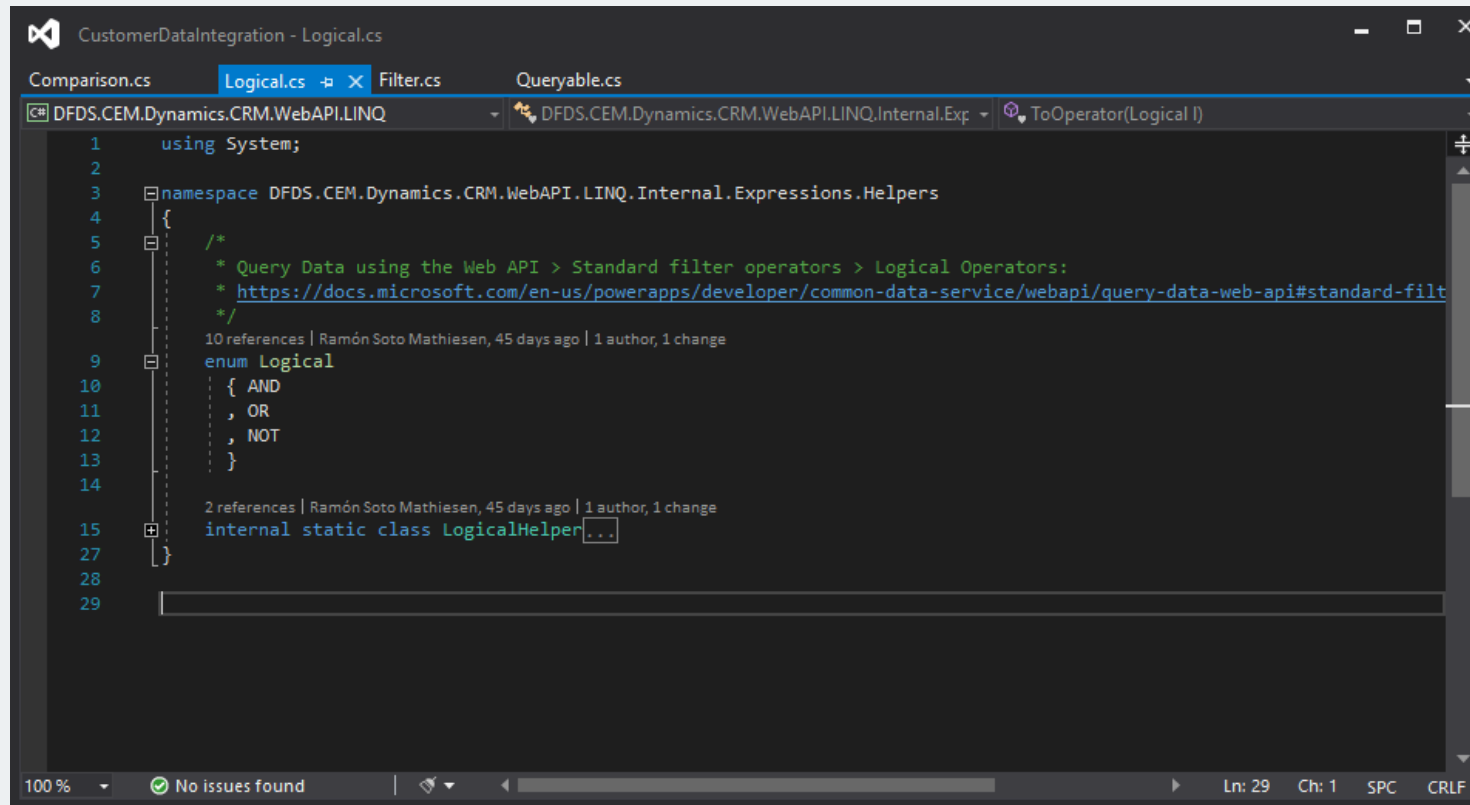


```
CustomerDataIntegration - Comparison.cs
Comparison.cs Logical.cs Filter.cs Queryable.cs
DFDS.CEM.Dynamics.CRM.WebAPI.LINQ DFDS.CEM.Dynamics.CRM.WebAPI.LINQ.Internal.Exp Equal
1  System;
2
3  namespace DFDS.CEM.Dynamics.CRM.WebAPI.LINQ.Internal.Expressions.Helpers
4
5  /*
6  * Query Data using the Web API > Standard filter operators:
7  * https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/webapi/query-data-web-api#standard-filter-of
8  *
9  * Query Data using the Web API > Standard query functions:
10 * https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/webapi/query-data-web-api#standard-query-fur
11 */
12 24 references | Ramón Soto Mathiesen, 45 days ago | 1 author, 1 change
13  enum Comparison
14  /* Binary operators */
15  {
16  , Equal
17  , NotEqual
18  , GreaterThan
19  , GreaterThanOrEqual
20  , LessThan
21  , LessThanOrEqual
22  /* Functions */
23  , Contains
24  , EndsWith
25  , StartsWith
26  }
100 % No issues found Ln: 7 Ch: 8 SPC CRLF
```



# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

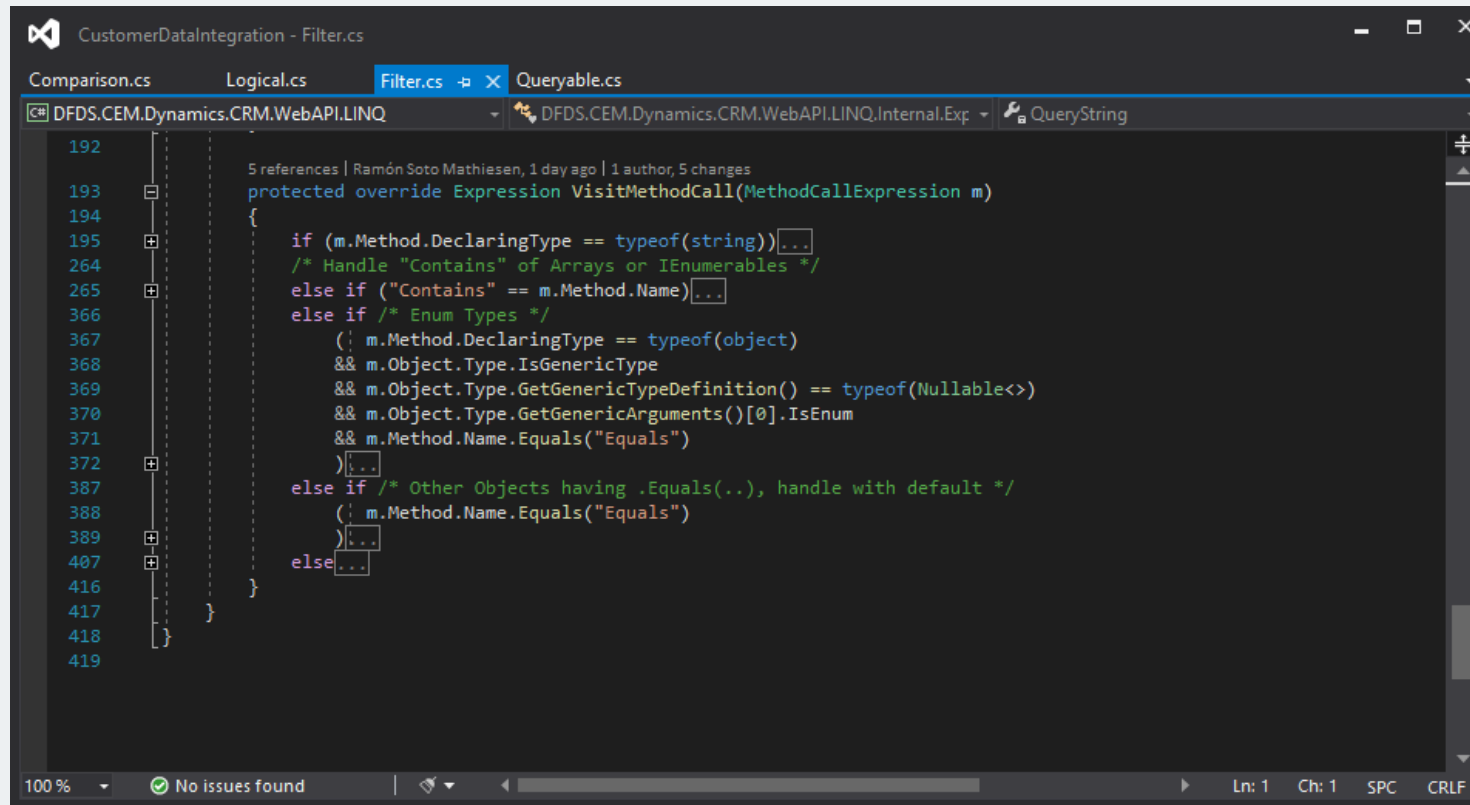
## LINQ Provider > Queries > Supported operations



```
CustomerDataIntegration - Logical.cs
Comparison.cs Logical.cs Filter.cs Queryable.cs
DFDS.CEM.Dynamics.CRM.WebAPI.LINQ DFDS.CEM.Dynamics.CRM.WebAPI.LINQ.Internal.Exp ToOperator(Logical I)
1 using System;
2
3 namespace DFDS.CEM.Dynamics.CRM.WebAPI.LINQ.Internal.Expressions.Helpers
4 {
5     /*
6     * Query Data using the Web API > Standard filter operators > Logical Operators:
7     * https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/webapi/query-data-web-api#standard-filt
8     */
9     10 references | Ramón Soto Mathiesen, 45 days ago | 1 author, 1 change
10     enum Logical
11     {
12         AND
13         , OR
14         , NOT
15     }
16     2 references | Ramón Soto Mathiesen, 45 days ago | 1 author, 1 change
17     internal static class LogicalHelper...
18
19
20
21
22
23
24
25
26
27
28
29
```

# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

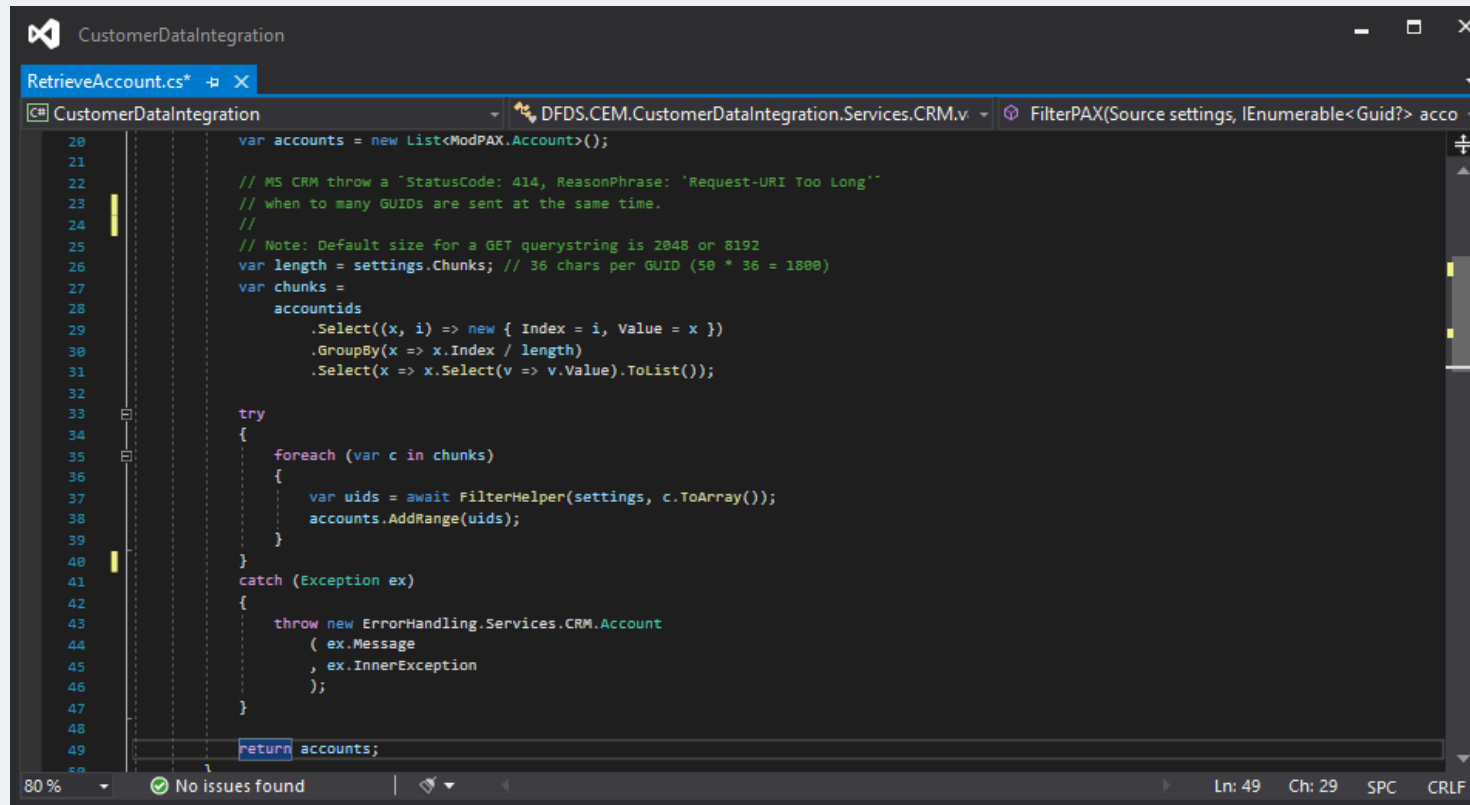
## LINQ Provider > Queries > Supported operations (+ Contains for containers data types)



```
CustomerDataIntegration - Filter.cs
Comparison.cs Logical.cs Filter.cs Queryable.cs
DFDS.CEM.Dynamics.CRM.WebAPI.LINQ DFDS.CEM.Dynamics.CRM.WebAPI.LINQ.Internal.Exp QueryString
5 references | Ramón Soto Mathiasen, 1 day ago | 1 author, 5 changes
192 protected override Expression VisitMethodCall(MethodCallExpression m)
193 {
194     {
195         if (m.Method.DeclaringType == typeof(string))...
264         /* Handle "Contains" of Arrays or IEnumerableables */
265         else if ("Contains" == m.Method.Name)...
366         else if /* Enum Types */
367             (: m.Method.DeclaringType == typeof(object)
368             && m.Object.Type.IsGenericType
369             && m.Object.Type.GetGenericTypeDefinition() == typeof(Nullable<>)
370             && m.Object.Type.GetGenericArguments()[0].IsEnum
371             && m.Method.Name.Equals("Equals")
372             )...
387         else if /* Other Objects having .Equals(..), handle with default */
388             (: m.Method.Name.Equals("Equals")
389             )...
407         else...
416     }
417 }
418 }
419 }
```

# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

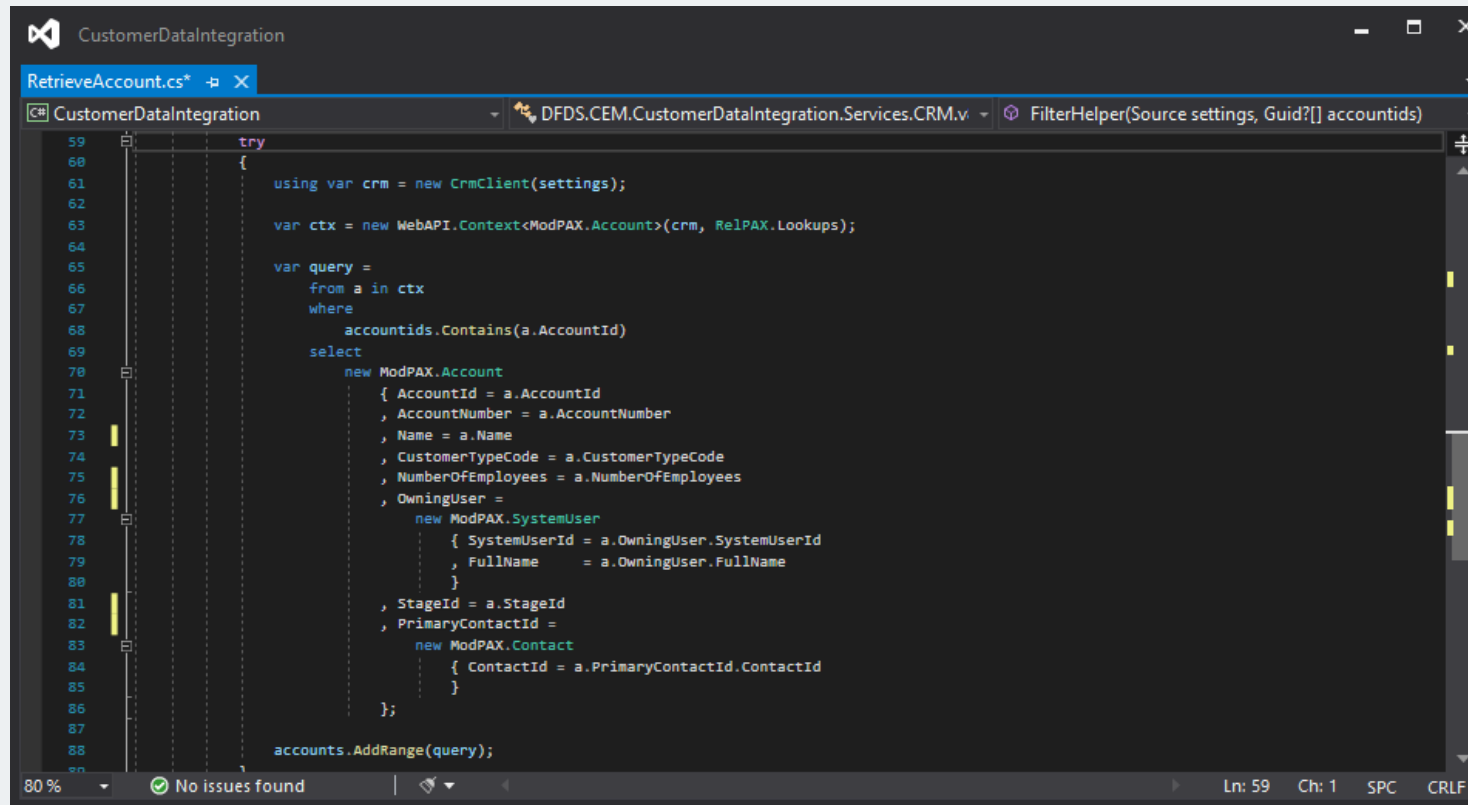
LINQ Provider > Queries > Chunks as translated output will be a Query String (limitation WebAPI)



```
CustomerDataIntegration
RetrieveAccount.cs*
CustomerDataIntegration DFDS.CEM.CustomerDataIntegration.Services.CRM.v FilterPAX(Source settings, IEnumerable<Guid?> acco
20 var accounts = new List<ModPAX.Account>();
21
22 // MS CRM throw a "StatusCode: 414, ReasonPhrase: 'Request-URI Too Long'"
23 // when to many GUIDs are sent at the same time.
24 //
25 // Note: Default size for a GET querystring is 2048 or 8192
26 var length = settings.Chunks; // 36 chars per GUID (50 * 36 = 1800)
27 var chunks =
28     accounts
29     .Select((x, i) => new { Index = i, Value = x })
30     .GroupBy(x => x.Index / length)
31     .Select(x => x.Select(v => v.Value).ToList());
32
33 try
34 {
35     foreach (var c in chunks)
36     {
37         var uids = await FilterHelper(settings, c.ToArray());
38         accounts.AddRange(uids);
39     }
40 }
41 catch (Exception ex)
42 {
43     throw new ErrorHandling.Services.CRM.Account
44         ( ex.Message
45         , ex.InnerException
46         );
47 }
48
49 return accounts;
```

# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

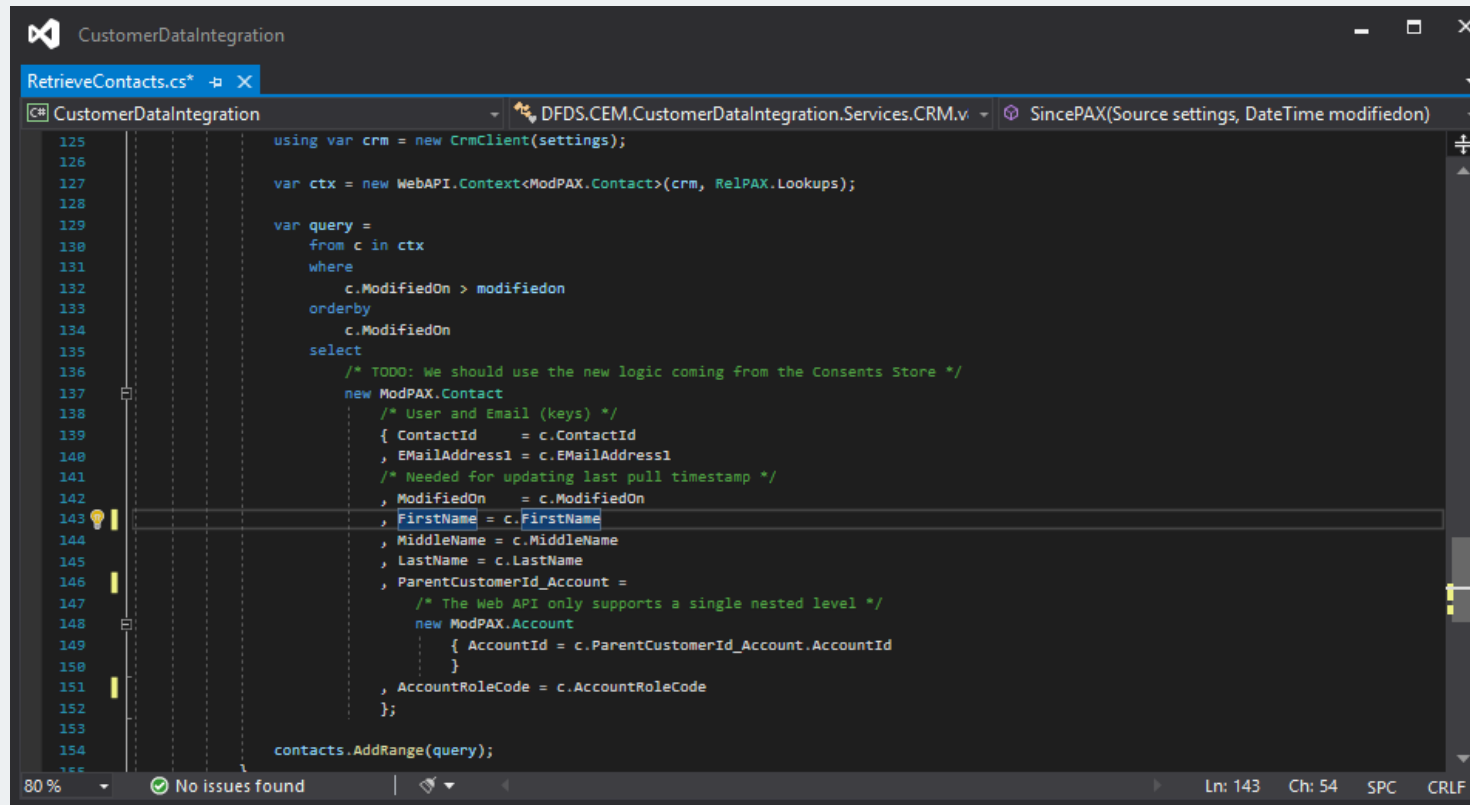
LINQ Provider > Queries > Chunks as translated output will be a Query String (limitation WebAPI)



```
59     try
60     {
61         using var crm = new CrmClient(settings);
62
63         var ctx = new WebAPI.Context<ModPAX.Account>(crm, RelPAX.Lookups);
64
65         var query =
66         from a in ctx
67         where
68             accountids.Contains(a.AccountId)
69         select
70             new ModPAX.Account
71             {
72                 AccountId = a.AccountId
73                 , AccountNumber = a.AccountNumber
74                 , Name = a.Name
75                 , CustomerTypeCode = a.CustomerTypeCode
76                 , NumberOfEmployees = a.NumberOfEmployees
77                 , OwningUser =
78                     new ModPAX.SystemUser
79                     {
80                         SystemUserId = a.OwningUser.SystemUserId
81                         , FullName = a.OwningUser.FullName
82                     }
83                 , StageId = a.StageId
84                 , PrimaryContactId =
85                     new ModPAX.Contact
86                     {
87                         ContactId = a.PrimaryContactId.ContactId
88                     }
89             };
90
91         accounts.AddRange(query);
92     }
```

# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

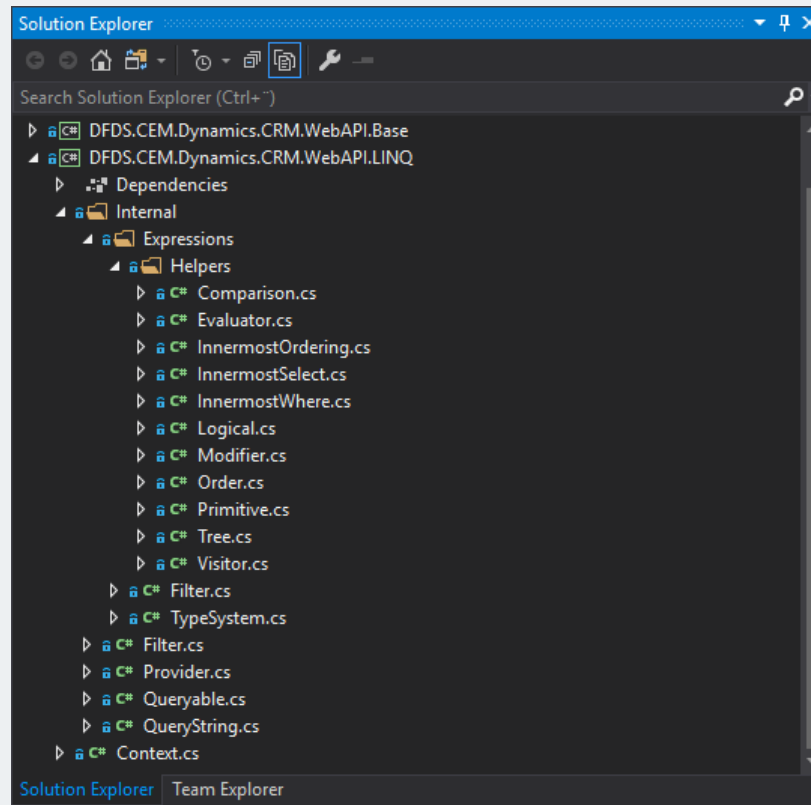
LINQ Provider > Queries > Only one level of nested entities is permitted (sadly we can only catch this runtime)



```
125 using var crm = new CrmClient(settings);
126
127 var ctx = new WebAPI.Context<ModPAX.Contact>(crm, RelPAX.Lookups);
128
129 var query =
130     from c in ctx
131     where
132         c.ModifiedOn > modifiedon
133     orderby
134         c.ModifiedOn
135     select
136         /* TODO: We should use the new logic coming from the Consents Store */
137         new ModPAX.Contact
138         /* User and Email (keys) */
139         { ContactId = c.ContactId
140           , EmailAddress1 = c.EmailAddress1
141           /* Needed for updating last pull timestamp */
142           , ModifiedOn = c.ModifiedOn
143           , FirstName = c.FirstName
144           , MiddleName = c.MiddleName
145           , LastName = c.LastName
146           , ParentCustomerId_Account =
147             /* The Web API only supports a single nested level */
148             new ModPAX.Account
149             { AccountId = c.ParentCustomerId_Account.AccountId
150             }
151           , AccountRoleCode = c.AccountRoleCode
152         };
153
154 contacts.AddRange(query);
```

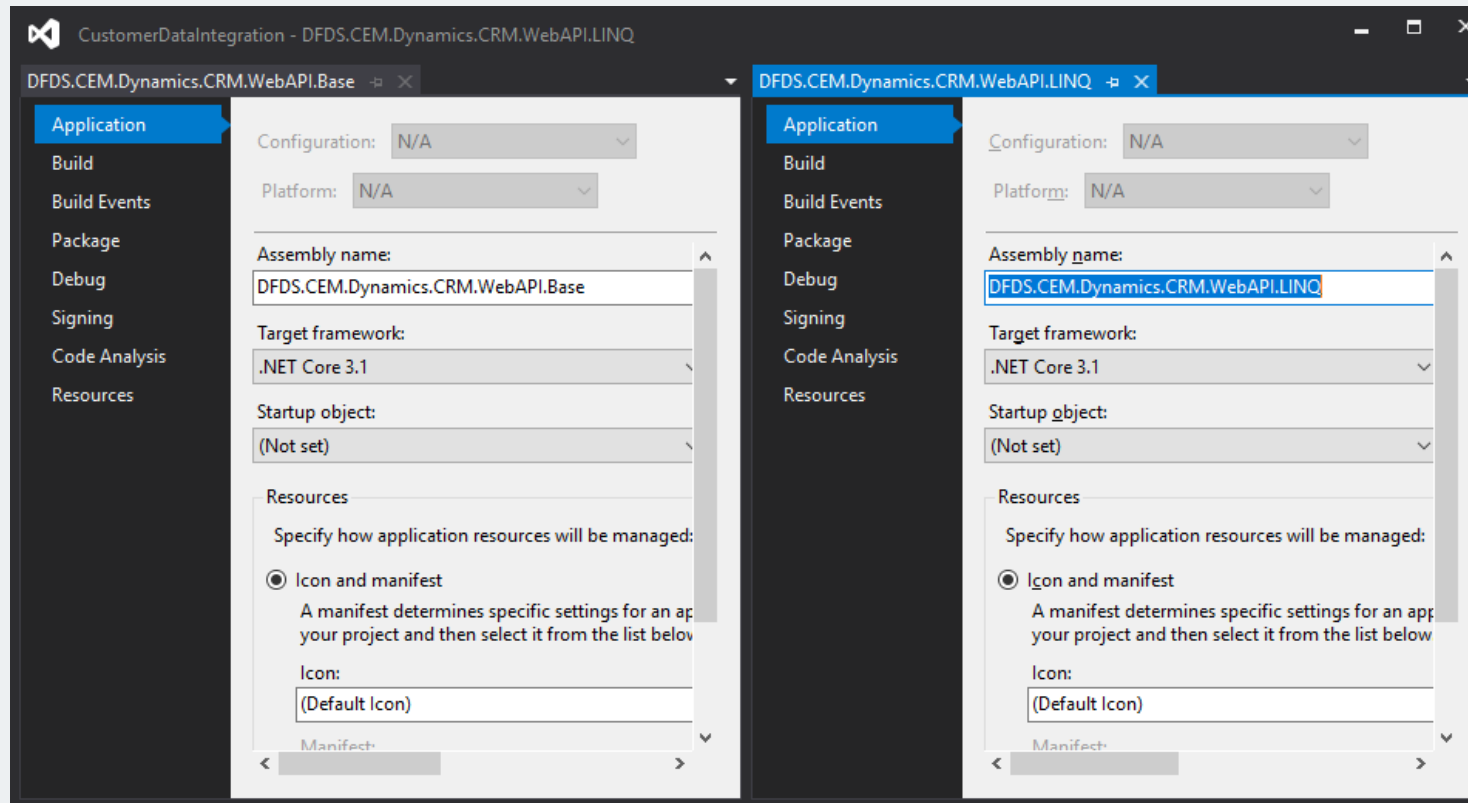
# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

## LINQ Provider > (simple) Visual Studio (C#) Project



# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

## LINQ Provider > (simple) Visual Studio (C#) Project > .NET Core v.3.1 (LTS)



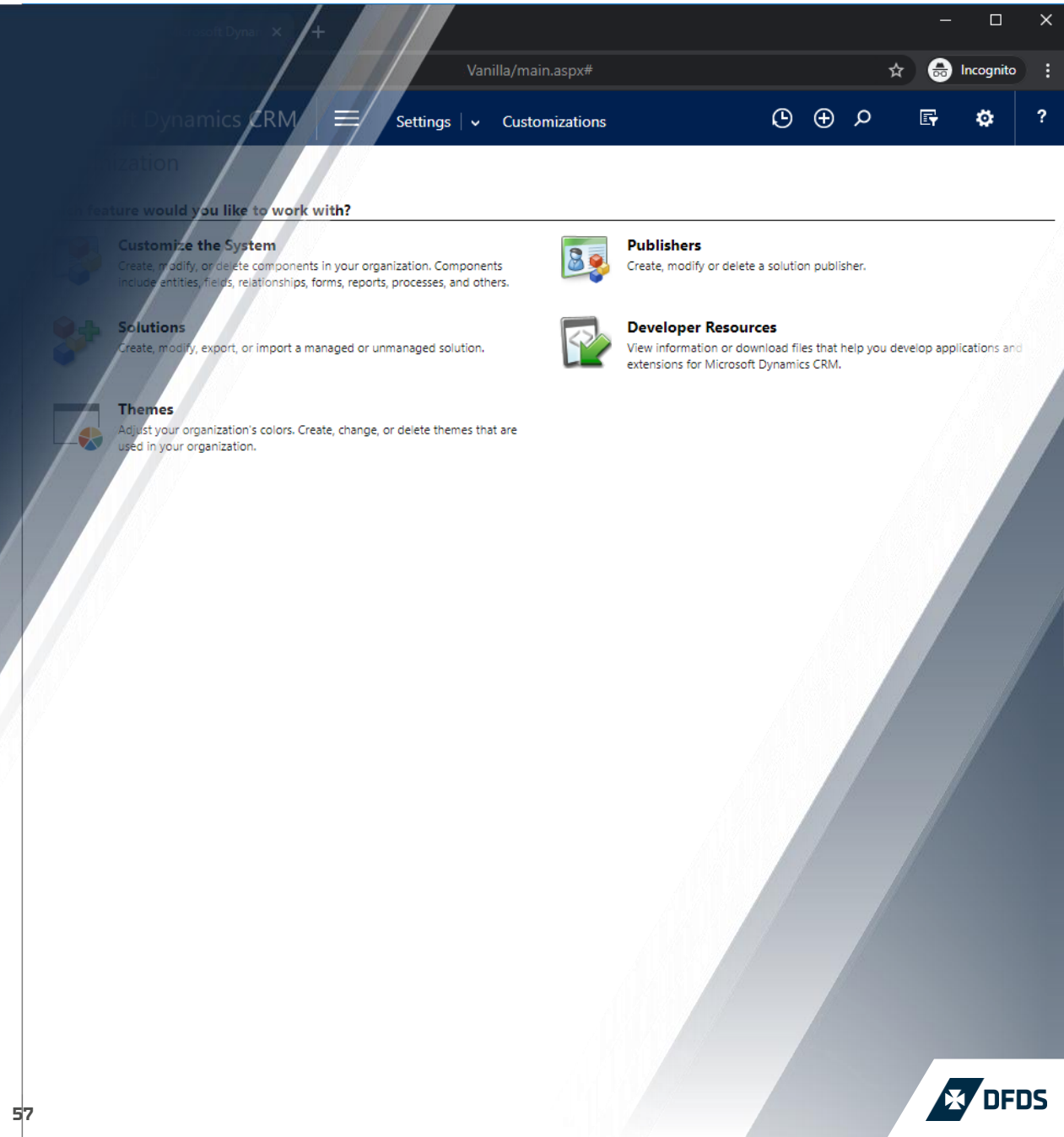
# LINQ Provider > DFDS.CEM.Dynamics.CRM.WebAPI.LINQ (lib)

## Background > Platform > Architecture > REST / WebAPI (JSON) > Ideally what would we like to have?

- Generate **type-safe C# classes** for xRM data entities of **any CRM application instance**
  - ✓ Support for **LINQ code**
  - ✓ Data **transformation from and to JSON** should be done **automatically** (sanitization)
- **Lightweight** size of generated types
  - ✓ **Concurrency / parallelism**
  - ✓ Support **.NET Core v.3.1 (LTS)**
    - Runtime infrastructure
    - Development setup
  - ✓ **Few dependencies**
    - At most, Json.NET - Newtonsoft library, as Microsoft recommends to use it
    - None to the xRM SDKs as they don't support .NET Core
- **No SDK 2011 endpoint**
  - Only **/api/data/v8.1/** and above must be used



# Cloud Ready



The screenshot shows the Microsoft Dynamics CRM interface in an Incognito browser window. The address bar shows 'Vanilla/main.aspx#'. The navigation bar includes 'Dynamics CRM', 'Settings', and 'Customizations'. The main content area is titled 'Customization' and features a section 'Which feature would you like to work with?' with four options: 'Customize the System', 'Solutions', 'Themes', 'Publishers', and 'Developer Resources'. Each option includes a brief description of its function.

Microsoft Dynamics CRM

Vanilla/main.aspx#


Settings | Customizations

### Customization

Which feature would you like to work with?

- Customize the System**  
Create, modify, or delete components in your organization. Components include entities, fields, relationships, forms, reports, processes, and others.
- Solutions**  
Create, modify, export, or import a managed or unmanaged solution.
- Themes**  
Adjust your organization's colors. Create, change, or delete themes that are used in your organization.
- Publishers**  
Create, modify or delete a solution publisher.
- Developer Resources**  
View information or download files that help you develop applications and extensions for Microsoft Dynamics CRM.

57



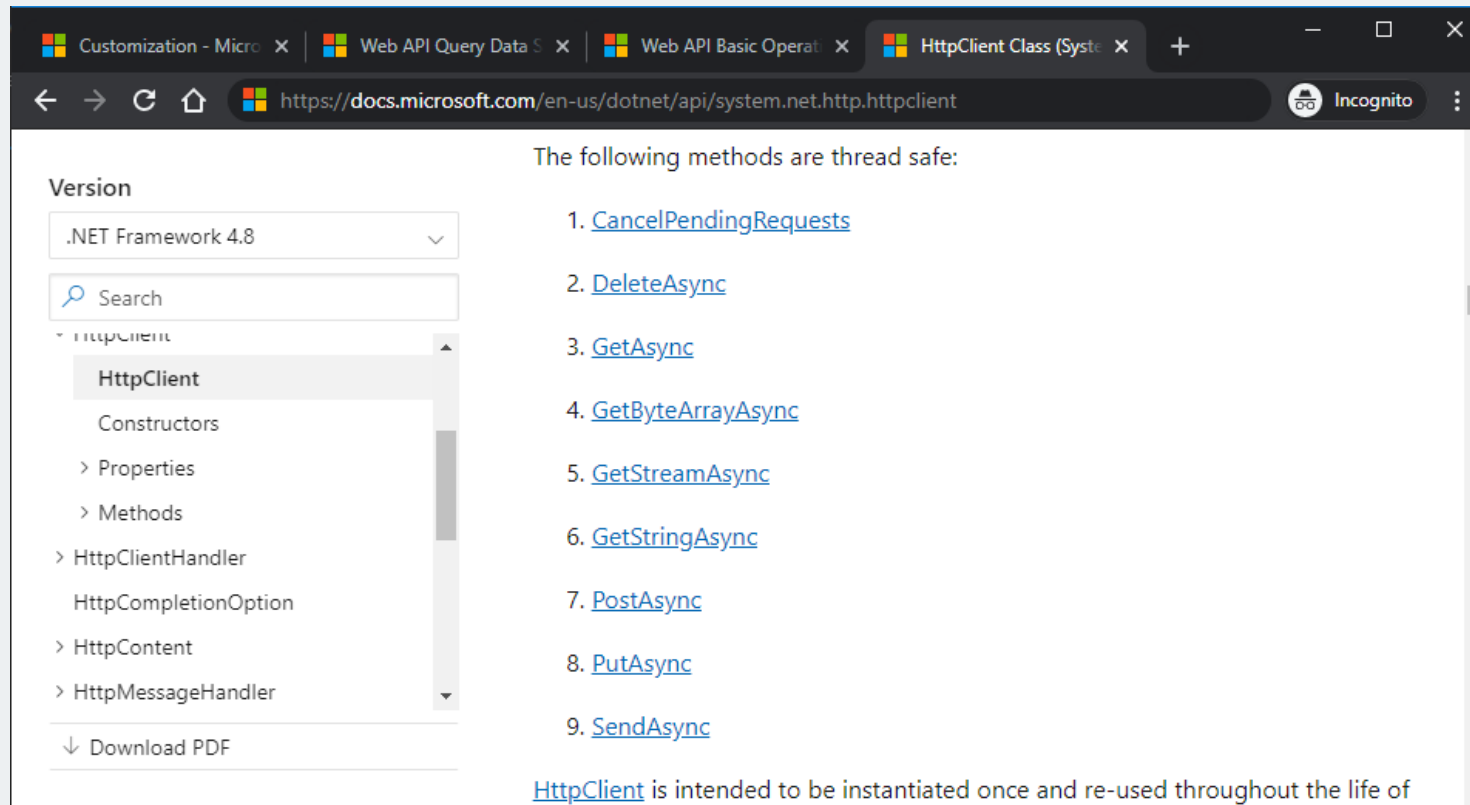
# Cloud Ready

## Cloud Ready > Design and develop CRM/xRM micro-apps

- **CQRS + DDD** (Strongly typed to catch data inconsistencies at build time)
- **Micro-applications**, read as in micro-services
  - **Stateless** applications with **low CPU** and **memory footprint**
- 100% **.NET Core 3.1** (LTS) = Docker ready
- **Scalable** as all exposed methods are **concurrent**:  
`public static async Task<..> MethodLogic`
- And therefore can be **used** with the (**thread-safe**) **HttpClient** to achieve **parallelism** ...

# Cloud Ready

## Cloud Ready > Design and develop CRM/xRM micro-apps > Scalable > Concurrency and parallelism



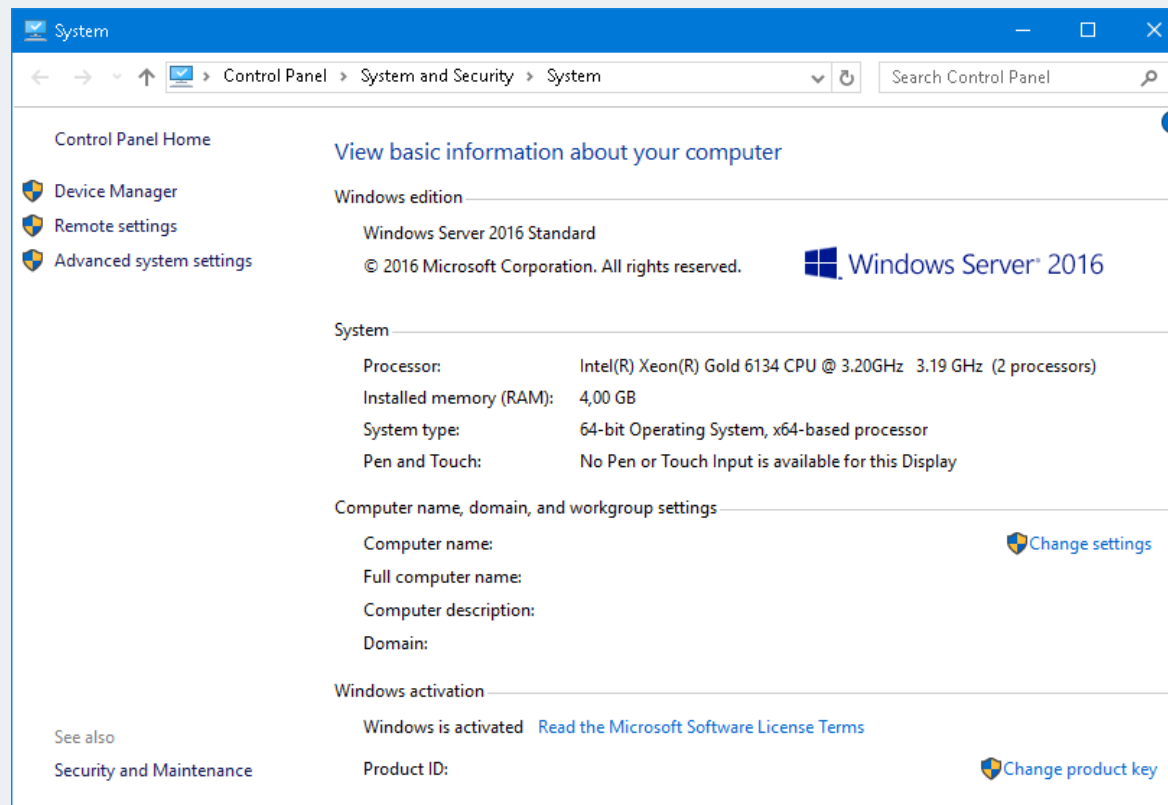
The screenshot shows a web browser window displaying the Microsoft Docs page for the `HttpClient` class. The browser tabs include "Customization - Micro", "Web API Query Data S", "Web API Basic Operat", and "HttpClient Class (System)". The address bar shows the URL <https://docs.microsoft.com/en-us/dotnet/api/system.net.http.httpclient>. The page content includes a "Version" dropdown set to ".NET Framework 4.8", a search bar, and a navigation menu with "HttpClient" selected. The main content area lists the following thread-safe methods:

- [CancelPendingRequests](#)
- [DeleteAsync](#)
- [GetAsync](#)
- [GetByteArrayAsync](#)
- [GetStreamAsync](#)
- [GetStringAsync](#)
- [PostAsync](#)
- [PutAsync](#)
- [SendAsync](#)

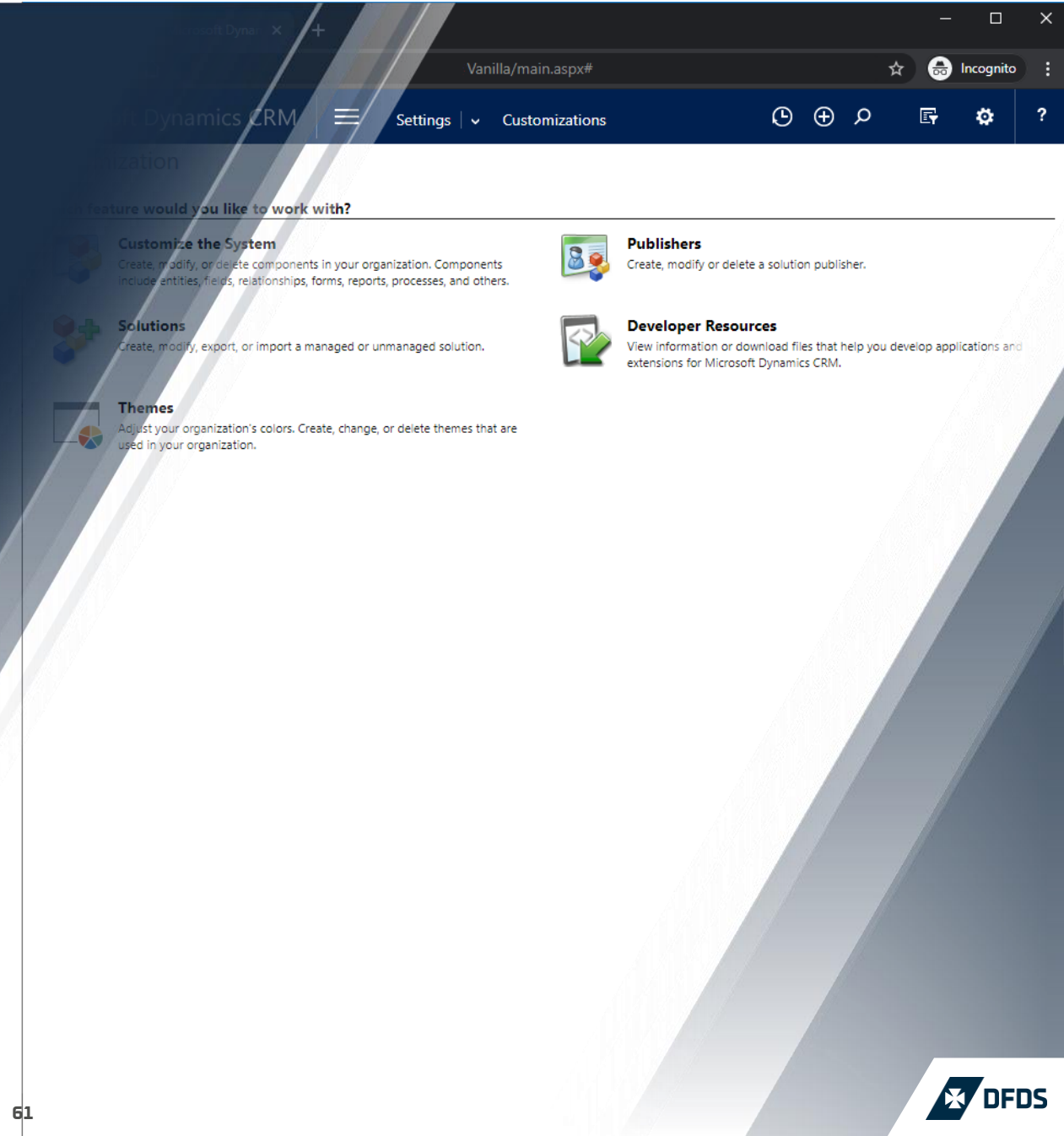
Below the list, it states: `HttpClient` is intended to be instantiated once and re-used throughout the life of

# Cloud Ready

**Cloud Ready > Design and develop CRM/xRM micro-apps > Robust and Stable (≈ 10% CPU and ≈ 50 MB when peaking)**



# Summary



The screenshot shows a web browser window displaying the Microsoft Dynamics CRM Customizations page. The browser's address bar shows the URL "Vanilla/main.aspx#". The page title is "Microsoft Dynamics CRM" and the navigation menu includes "Settings" and "Customizations". The main content area is titled "Customization" and features a section "Which feature would you like to work with?". This section contains four cards: "Customize the System" (with a gear icon), "Solutions" (with a plus icon), "Themes" (with a color palette icon), "Publishers" (with a person icon), and "Developer Resources" (with a document icon). Each card includes a brief description of the feature.

Microsoft Dynamics CRM

Vanilla/main.aspx#

Settings | Customizations

Customization

Which feature would you like to work with?

- Customize the System**  
Create, modify, or delete components in your organization. Components include entities, fields, relationships, forms, reports, processes, and others.
- Solutions**  
Create, modify, export, or import a managed or unmanaged solution.
- Themes**  
Adjust your organization's colors. Create, change, or delete themes that are used in your organization.
- Publishers**  
Create, modify or delete a solution publisher.
- Developer Resources**  
View information or download files that help you develop applications and extensions for Microsoft Dynamics CRM.

61

DFDS

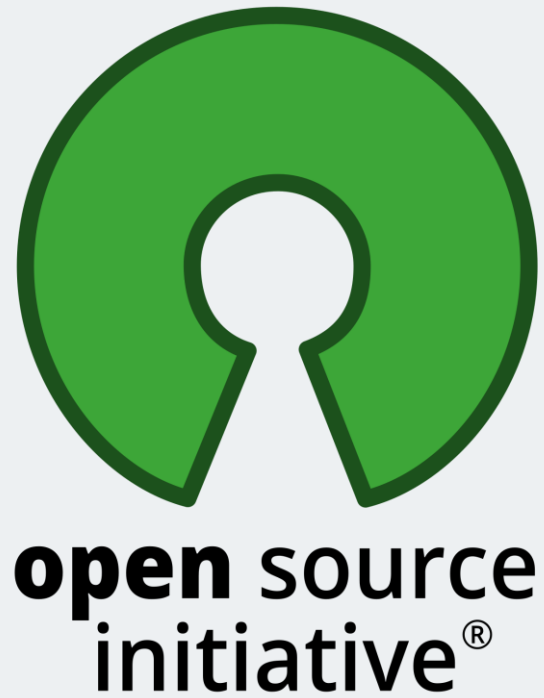
# CRM WebAPI Util + LINQ Provider = Cloud Ready

**Background > Platform > Architecture > REST / WebAPI (JSON) > Ideally what would we like to have?**

- ✓ Generate **type-safe C# classes** for xRM data entities of **any CRM application instance**
- ✓ Support for **LINQ code**
- ✓ Data **transformation from and to JSON** should be done **automatically** (sanitization)
- ✓ **Lightweight** size of generated types
- ✓ **Concurrency / parallelism**
- ✓ Support **.NET Core v.3.1 (LTS)**
  - Runtime infrastructure
  - Development setup
- ✓ **Few dependencies**
  - At most, Json.NET - Newtonsoft library, as Microsoft recommends to use it
  - None to the xRM SDKs as they don't support .NET Core
- ✓ **No SDK 2011 endpoint**
  - Only **/api/data/v8.1/** and above must be used

# Summary

Summary > Open Source? It's up to DFDS A/S (<https://github.com/dfds>)



# Summary

Summary > Help **us**, help you > In the age of collaboration/cooperation, ensure your platform is as easy-to-use as possible





# Q & A