

# C# SDK Version 2



Zoho CRM  
-zoho.com/crm-

# Table of Contents

<b>1. Overview</b> .....	3
a. Environmental Setup	
<b>2. Register your Application</b> .....	5
<b>3. Persistence</b> .....	6
a. Zoho OAuth File Persistence	
b. Zoho OAuth DB Persistence	
c. Zoho OAuth In memory Persistence	
<b>4. Configuration</b> .....	8
<b>5. Initialization</b> .....	11
a. Generating Grant Tokens	
b. Generating Access Tokens	
<b>6. Class Hierarchy</b> .....	17
<b>7. REST API Samples</b> .....	19
<b>8. Responses and Exceptions</b> .....	31
<b>9. Errors and Solutions</b> .....	34
<b>10. Release Notes</b> .....	52
a. Current Version	
b. Previous Versions	

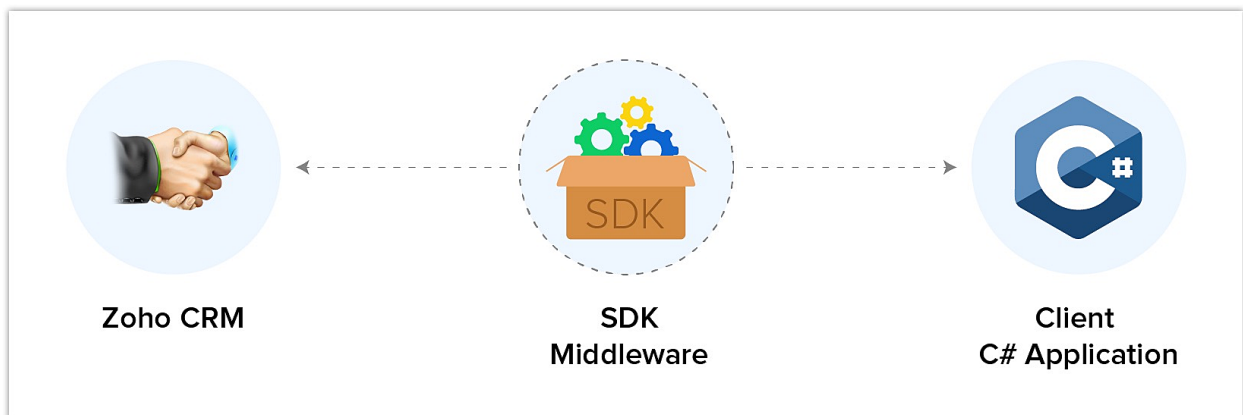


## Overview

C# SDK offers a way to create client C# applications that can be integrated with Zoho CRM. This SDK makes the access and use of necessary CRM APIs with ease. In other words, it serves as a wrapper for the REST APIs, making it easier to use the services of Zoho CRM.

A point to note would be that the developer of the client application should create programming code elements along with configuration-related properties files, interface implementations, instances or objects. Authentication to access Zoho CRM APIs is through OAuth authentication mechanism. Invariably, HTTP requests and responses are taken care by SDK.

A sample of how an SDK acts a middle ware or interface between Zoho CRM and a client C# application.



C# SDK allows you to:

1. Exchange data between Zoho CRM and the client application where the CRM entities are modelled as classes.
2. CRM API equivalents are declared and defined as simple functions in your C# application.
3. Push data into Zoho CRM, by accessing appropriate APIs of the CRM Service.

**Note:**

You must have **.net framework 4.6.1** or above for the SDK to work.

## Environmental Setup

C# SDK requires .NET Framework 4.6.1(or above) or .Net Core 2.X(or above) to be set up in your development environment. The compatibility warning can be ignored. C# SDK is available as a Nuget package. The SDK requires the following from the client app:

The ZCRMSDK assembly can be installed through the Nuget Package Manager and through the following options:

### Packet Manager

```
1 Install-Package ZCRMSDK
2 Install-Package Newtonsoft.Json
3 Install-Package MySql.Data
4 Install-Package System.IO.Compression.ZipFile -Version
  4.3.0
5 Install-Package System.IO.Compression -Version 4.3.0
```

### .NET CLI

```
1 dotnet add package ZCRMSDK
2 dotnet add package Newtonsoft.Json
3 dotnet add package MySql.Data [Only if DB persistence is
  used]
4 dotnet add package System.IO.Compression.ZipFile --version
  4.3.0
5 dotnet add package System.IO.Compression --version 4.3.0
```



## Note

- The **access and refresh tokens are environment-specific and domain-specific**. When you handle various environments and domains such as Production, Sandbox, or Developer and IN, CN, US, EU, or AU, respectively, you must use the access token and refresh token generated only in those respective environments and domains. The SDK throws an error, otherwise. For example, if you generate the tokens for your Sandbox environment in the CN domain, you must use only those tokens for that domain and environment. You cannot use the tokens generated for a different environment or a domain.
- For **Contact Roles** and **Records API**, you will need to provide the **ZohoCRM.settings.fields.ALL** scope along with the **ZohoCRM.modules.ALL** scope while generating the OAuth token. Otherwise, the system returns the **OAUTH-SCOPE-MISMATCH** error.
- For **Related Records API**, the scopes required for generating OAuth token are **ZohoCRM.modules.ALL**, **ZohoCRM.settings.fields.ALL** and **ZohoCRM.settings.related\_lists.ALL**. Otherwise, the system returns the **OAUTH-SCOPE-MISMATCH** error

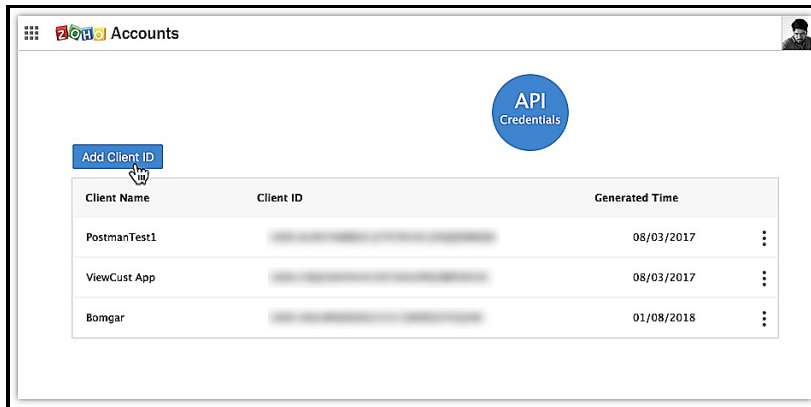
## Register your application

All the Zoho CRM APIs are authenticated with OAuth2 standards, so it is mandatory to register and authenticate your client app with Zoho.

### To register:

1. Go to the site: <https://api-console.zoho.com>
2. Click **Add Client ID**.





3. Enter the **Client Name**, **Client Domain** and **Authorized Redirect URL**.

4. Select the **Client Type** as **Web based**

Create Zoho Client ID

Client Name  
Internal Data Compiler

Client Domain  
www.abc.com

Authorized redirect URIs  
https://www.abc.com

Client Type  
 WEB Based

Create Cancel

5. Click **Create**.

6. Your Client app would have been created and displayed by now.

7. The newly registered app's Client ID and Client Secret can be found by clicking **Options** → **Edit**.

**Note:**

Options is the three dot icon at the right corner.

Registered applications will receive the following credentials:

- **Client id** – The consumer key generated from the connected app.
- **Client Secret** – The consumer secret generated from the connected app.
- **Redirect URI** – The Callback URL that you registered during the app registration.

## Persistence

### Implementing OAuth Persistence

Once the application is authorized, OAuth access and refresh tokens can be used for subsequent user data requests to Zoho CRM. Hence, they need to be persisted by the client app.

The persistence is achieved by writing an implementation of the inbuilt `IZohoPersistenceHandler` interface, which has the following callback methods.

- **SaveOAuthTokens(ZohoOAuthTokens tokens)** – invoked while fetching:
  - access and refresh tokens using grant token.
  - access token using refresh token.
- **DeleteOAuthTokens()**– invoked before saving the newly received tokens.
- **GetOAuthTokens()** – invoked before firing a request to fetch the saved tokens.  
This method should return `ZohoOAuthTokens` object for the library to process it.

Our C# SDK provides three sample implementations of `IZohoPersistenceHandler` interface within the client library. They are:

- `ZohoOAuthFilePersistence`
- `ZohoOAuthDBPersistence`
- `ZohoOAuthInMemoryPersistence`

The name (along with its assembly comma separated) of the implemented class or the handlers provided by the SDK should be given as value for the key `persistence_handler_class`.

Like '`persistence_handler_class=<persistence_handler_class, assembly_name>`', under the `oauth_configuration` section in the `app.config` file.

**Note:**

If the persistence handler class is not specified, InMemory Persistence handler handles the persistence implementation by default.

Pre-defined persistence handler classes belong to the assembly ZCRMSDK.

## ZohoOAuthFilePersistence

This method of persistence uses a local file to write and read the OAuth tokens.

The complete path of the file to be used by the library to write and read the tokens should be specified under the `oauth_configuration` section in `app.config` file as the value of the key `oauth_tokens_file_path`.

## ZohoOAuthDBPersistence

This method of persistence uses a custom MySQL persistence. To use this, you should make sure of the following.

- MySQL should be running in the same machine serving at the default port 3306.
- The database name should be `zohooauth`.
- There must be a table `*oauthtokens*` with the columns `useridentifier` (`varchar(100)`), `accesstoken` (`varchar(100)`), `refreshtoken` (`varchar(100)`) and `expirytime` (`bigint`).

## ZohoOAuthInMemoryPersistence

Uses a singleton class to store and retrieve tokens. Default implementation and requires no external file.

**Note:**

- ZohoOAuthFilePersistence and ZohoOAuthInMemoryPersistence implementations only support to store and refresh only a single user's token. Hence they should be used only if the app accesses Zoho APIs on behalf of a single user.
- In case if the app has to support for multiple users, please use the ZohoOAuthDBPersistence or write your own implementation of





## Configuration

Before you get started with creating your C# application, you need to first authenticate the app with Zoho. And to do that there are some configuration procedures that need to be in place. Basically, the OAuth Client details must be given as a section within the app.config file.

### Updates for SDK Version 2.0.1+

- The SDK is compatible with Visual Studio 2015.
- Also, the sdk will need the application configuration be specified only as a dictionary and not as an app.config file.

### Note:

Only .Net Core applications will have the app.config file.

For ASP.NET, ASP Web App, ASP Website, etc, please use configuration dictionary to configure your application.

Add a section **oauth\_configuration** in the app.config file and make sure that the section has the attribute type as **'ZCRMSDK.CRM.Library.Common.ConfigFileHandler.ConfigFileSection, ZCRMSDK'**. For example:

```
1 <configuration>
2   <configSections>
3     <section name="oauth_configuration"
4       type="ZCRMSDK.CRM.Library.Common.ConfigFileHandler.ConfigFileSect
```



```

5 </configSections>
6 <oauth_configuration>
7     <settings>
8         <add key = "client_id" value = "" />
9         <add key = "client_secret" value = "" />
10        <add key = "redirect_uri" value = "" />
11        <add key = "access_type" value = "" />
12        <add key = "iamUrl" value = "" />
13        <add key = "persistence_handler_class" value = ""/>
14        <add key = "mysql_username" value = "" />
15        <add key = "mysql_password" value = "" />
16        <add key = "mysql_database" value = "" />
17        <add key = "mysql_server" value = "" />
18        <add key = "mysql_port" value = "" />
19        <add key = "oauth_tokens_file_path" value = "" />
20    </settings>
21 </oauth_configuration>
22 <zcrm_configuration>
23     <settings>
24         <add key = "apiBaseUrl" value = ""/>
25         <add key = "photoUrl" value = ""/>
26         <add key = "apiVersion" value = ""/>
27         <add key = "logFilePath" value = ""/>
28         <add key = "timeout" value = ""/>
29         <add key = "minLogLevel" value = ""/>
30         <add key = "currentUserEmail" value = "" />
31         <add key = "domainSuffix" value = "" />
32     </settings>
33 </zcrm_configuration>
34 </configuration>

```

- **client\_id**, **client\_secret** and **redirect\_uri** are your OAuth client's configurations that you get after registering your Zoho client.
- **access\_type** will be set to offline by default. Access and Refresh tokens will be received only when it is offline.
- **iamUrl** - Url to be used when calling an Oauth accounts. It is used to denote the domain of the user. Url may be:
  - <https://accounts.zoho.com> for US.



- <https://accounts.zoho.eu> for European countries.
- <https://accounts.zoho.com.cn> for China.
- <https://accounts.zoho.jp> for Japan.
- **persistence\_handler\_class** is your implementation of the ZohoPersistenceHandler interface, which has handler methods to store OAuth data.
- **For example:**  
persistence\_handler\_class=ZCRMSDK.OAuth.ClientApp.ZohoOAuthFilePersistence, ZCRMSDK (or) ZCRMSDK.OAuth.ClientApp.ZohoOAuthDBPersistence, ZCRMSDK (or) your own persistence handler class.
- If you prefer to use our DB persistence (**ZohoOAuthDBPersistence.cs**), you need to give the mysql\_username and mysql\_password keys for mysql connectivity.
  - By default, **mysql\_username** = "root", **mysql\_password** = "", **mysql\_database** = "zohooauth", **mysql\_server** = "localhost" and **mysql\_port** = "3306".
  - The tokens are generated and placed in the database table automatically(which is explained in the ZohoOAuthDBPersistence section) once the authentication process is complete.
- The **oauth\_tokens\_file\_path** is required if the SDK's File Persistence is used as the persistence handler. It is the path of the file for storing the tokens of the user.

Other than the above OAuth configurations, the SDK also provides options to override certain HTTP request attributes. These configurations should be provided under a section named zcrm\_configuration, in the app.config file.

The type of the section should be

'ZCRMSDK.CRM.Library.Common.ConfigFileHandler.ConfigFileSection, ZCRMSDK'.

The following are the supported configurations in the zcrm\_configuration section:

- **apiBaseUrl** - Url to be used when calling an API. It is used to denote the domain of the user. Url may be:
  - <https://www.zohoapis.com>
  - <https://www.zohoapis.eu>
  - <https://www.zohoapis.com.cn>
  - <https://www.zohoapis.jp>



- **photoUrl** - Url for the image representing the record. The domain might be different based on the apiBaseUrl. Url may be:
  - <https://profile.zoho.com/api/v1/user/self/photo>
  - <https://profile.zoho.eu/api/v1/user/self/photo>
  - <https://profile.zoho.com.cn/api/v1/user/self/photo>
- **apiVersion** is "v2".
- **timeOut** - Represents the request timeout in milliseconds. Let this be omitted or empty if not needed.
- **minLogLevel** - Represents the minimum log level for logging of SDK. The supported values are ALL, INFO, WARNING, ERROR and OFF. The default minimum log level is WARNING.
- **logFilePath** - Represents the file to which the SDK can log. Optional configuration and can be omitted. If omitted, the SDK logs the working in the execution directory of the application under the filename LogFile.log. Only the path of the file, without the file name, is needed for storing the logs.
- **currentUserEmail** - In case of single user, this configuration can be set.
- **domainSuffix** - Optional configuraion. Provides Multi-DC Support. Ex: com, eu or cn.
- **fileUploadUrl** - URL to be used when uploading the zip file. This key is mandatory when you use the Bulk Write API.
  - <https://content.zohoapis.com>

**Note:**

- If the file path for "**logFilePath**" is not specified, then the "logfile.log" is created in the "{Project}/bin/Debug/netcoreapp2.1/" folder of the project.
- The **access and refresh tokens are environment-specific and domain-specific**. When you handle various environments and domains such as Production, Sandbox, or Developer and IN, CN, US, EU, or AU, respectively, you must use the access token and refresh token generated only in those respective environments and domains. The SDK throws an error, otherwise.
- For example, if you generate the tokens for your Sandbox environment in the CN domain, you must use only those tokens for that domain and environment. You cannot use the tokens generated for a different environment or a domain.



## Initialize the Application

Now the app is ready to be initialized after defining configuration file/dictionary for your app.

## Generating Grant tokens

### *For a Single User(self-authorized)*

For self client apps, the self authorized grant token should be generated from the Zoho Developer Console (<https://accounts.zoho.com/developerconsole>). The developer console has an option to generate grant token for a user directly. This option may be handy when your app is going to use only one CRM user's credentials for all its operations or for your development testing.

1. Login to the User's account.
2. Visit <https://accounts.zoho.com/developerconsole>
3. Click on the **Options** → **Self Client** option of the client for which you wish to authorize.
4. Enter one or more (comma separated) valid Zoho CRM scopes that you wish to authorize in the "Scope" field and choose the time of expiry. Provide "aaaserver.profile.READ" scope along with Zoho CRM scopes.
5. Copy the **grant token** that is displayed on the screen.
6. Generate **refresh\_token** from grant token by making a POST request with the URL below:

```
1 https://accounts.zoho.com/oauth/v2/token?code={grant\_token}&redire
```

7. Copy the **refresh token** for backup.

### **Note:**

- The generated grant token is valid only for the stipulated time you chose while generating it. Hence, the access and refresh tokens should be generated within that time.
- The OAuth client registration and grant token generation must be done in the same Zoho account's (meaning - login) developer console.

## ***For Multiple Users***

For multiple users, it is the responsibility of your client app to generate the grant token from the users trying to login.

- Your Application's UI must have a "Login with Zoho" option to open the grant token URL of Zoho, which would prompt for the user's Zoho login credentials.
- Upon successful login of the user, the grant token will be sent as a param to your registered redirect URL.

### **Note**

- The **access and refresh tokens** are environment-specific and domain-specific. When you handle various environments and domains such as Production, Sandbox, or Developer and IN, CN, US, EU, or AU, respectively, you must use the access token and refresh token generated only in those respective environments and domains. The SDK throws an error, otherwise.
- For example, if you generate the tokens for your Sandbox environment in the CN domain, you must use only those tokens for that domain and environment. You cannot use the tokens generated for a different

## **Generating Access tokens**

Access token can be generated by grant token or refresh token. Following any one of the two methods given below is sufficient.

### ***From grant token***

The following code snippet should be executed from your main class to get access token.



```
1 ZCRMRestClient.Initialize(config);
2 ZohoOAuthClient client = ZohoOAuthClient.GetInstance();
3 string grantToken = <paste_grant_token_here>;
4 ZohoOAuthTokens tokens = client.GenerateAccessToken(grantToken);
5 string accessToken = tokens.AccessToken;
6 string refreshToken = tokens.RefreshToken;"
```

Please paste the generated grant token in the string literal mentioned. This is one time process only.

In case of multiple users using the application, you need to keep note of the following:

- In order for the SDK to identify the particular user who made the request, the requester's email address should be given through the following code snippet before making the actual method call of the SDK.

```
1 ZCRMRestClient.SetCurrentUser("provide_current_user_email_here")
```

In case of Single users, the current user email can be set either through the above code, or in the `zcrm_configuration` section in the `app.config` file with the key `currentUserEmail` as a one time configuration.

### ***From refresh token***

The following code snippet should be executed from your main class to get access token.

```
1 ZCRMRestClient.Initialize(config);
2 ZohoOAuthClient client = ZohoOAuthClient.GetInstance();
3 string refreshToken = <paste_refresh_token_here>;
```



```
4 string userEmailId = <provide_user_email_here>;
5 ZohoOAuthTokens tokens = client.
  GenerateAccessTokenFromRefreshToken(refreshToken,userMailId);
```

Please paste the generated refresh token in the string literal mentioned. This is one time process only.

#### Note

- The above code snippet is valid only once per grant token. Upon its successful execution, the generated access and refresh tokens would have been persisted through your persistence handler class.
- Once the OAuth tokens have been persisted, subsequent API calls would use the persisted access and refresh tokens. The SDK will take care of refreshing the access token using refresh token, as and when required.

## Start the App

The SDK requires the following line of code being invoked every time your app gets started.

```
1 "ZCRMRestClient.Initialize(config);"
```

#### Note

This method should be called from the main class of your c# application to start the application. It needs to be invoked without any exception.

The SDK also allows for custom initialization, overriding the data from the app.config file. Or, you could also override when there is no need for the config file. The custom initialization scenarios are:

```
1 "public static Dictionary<string, string> config = new
  Dictionary<string, string>()
2     {
3         {"client_id","1000.8ETLN5A9356890756HRWXWZ69VJCBN"},
```





```

4      {"client_secret","b477d8bac9a8ad722334582b3430fdca7dde44de4e"},
5          {"redirect_uri","{redirect_url}"},
6          {"access_type","offline"},
7
8      {"persistence_handler_class","ZCRMSDK.OAuth.ClientApp.ZohoOAuthDB
9
10         {"oauth_tokens_file_path","{file_path}"},
11         {"mysql_username","root"},
12         {"mysql_password",""},
13         {"mysql_database","zohooauth"},
14         {"mysql_server","localhost"},
15         {"mysql_port","3306"},
16         {"apiBaseUrl","{https}://www.zohoapis.com"},
17         {"fileUploadUrl","{https}://content.zohoapis.com"},
18         {"photoUrl","{photo_url}"},
19         {"apiVersion","v2"},
20         {"logFilePath","{log_file_path}" },
21         {"timeout",""},
22         {"minLogLevel",""},
23         {"domainSuffix","com"},
24         {"currentUserEmail","user@user.com"}
25     };
26 ZCRMRestClient.Initialize(config);"

```

**Note:**

Once the SDK has been initialized, you can use any APIs of the SDK to get proper results.

## Class Heirarchy

All Zoho CRM entities are modelled as modules having classes, methods and instance

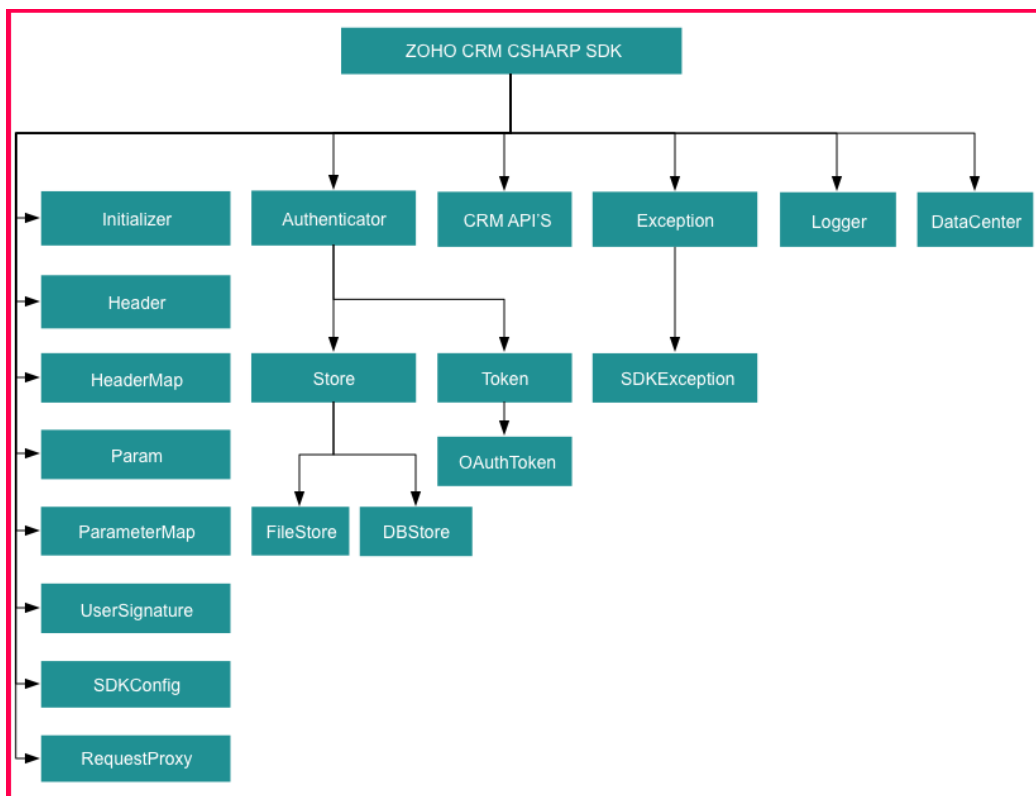


Zoho CRM  
-zoho.com/crm-

variables applicable to that particular entity. ZCRMRestClient is the base class of the C# SDK. ZCRMRestClient has methods to get instances of various other Zoho CRM entities. It is in RestClient module.

- ZCRMRestClient is the base class of the SDK.
- This class has, methods to get instances of various other Zoho CRM entities.
- The class relations and hierarchy of the SDK follows the entity hierarchy inside Zoho CRM.
- Each class entity has functions to fetch its own properties and to fetch data of its immediate child entities through an API call. For example: a Zoho CRM module (ZCRMModule) object will have member functions to get a module's properties like display name, module Id, etc, and will also have functions to fetch all its child objects (like ZCRMLayout).

The class relations and hierarchy of the SDK follows the entity hierarchy inside Zoho CRM. The class hierarchy of various Zoho CRM entities are given below:



As appearing in the hierarchy, every entity class will have instance variables to fetch its own properties and to fetch data of its immediate child entities through an API call.

## Instance Objects

It is not always effective to follow the complete class hierarchy from the top to fetch the data of an entity at some lower level, since this would involve API calls at each level. In order to handle this, every entity class will have a `GetInstance()` method to get its own dummy object and instance variables to get dummy objects of its child entities.

### Note:

`GetInstance()` methods would not have any of its properties filled, since it would not fire an API call. This would just return a dummy object that shall be only used to access the non-static methods of the class.

Summing it up,

- **`ZCRMRestClient.GetInstance().GetModule("Contacts")`** would return the actual Contacts module, that has all the properties of the Contacts module filled through an API call.
- **`ZCRMRestClient.GetModuleInstance("Contacts")`** would return a dummy `ZCRMModule` object that would refer to the Contacts module, with no properties filled, since this doesn't make an API call.

Hence, to get records from a module, you need not to start all the way from `ZCRMRestClient`. Instead, you could get a `ZCRMModule` instance with `ZCRMModule.GetInstance(module_api_name)` and then invoke its nonstatic `GetRecords()` method from the created instance. This would avoid the API call which would have been triggered to populate the `ZCRMModule` object.

## Accessing record properties

Since record properties are dynamic across modules, we have only given the common fields like **CreatedTime**, **CreatedBy**, **Owner** etc, as **ZCRMRecord's** default members. All other record properties are available as a map in `ZCRMRecord` object.

To access the individual field values of a record, use the getter and setter methods available. The keys of the record properties map are the API names of the module's fields. API names of all fields of all modules are available under,



**Setup → Marketplace → APIs → CRM API → API Names.**

- To get a field value, use `record.GetFieldValue(field_api_name);`
- To set a field value, use `record.SetFieldValue(field_api_name, new_value);` While setting a field value, please make sure of that the set value is of the data type of the field to which you are going to set it.

## Sample Codes

All of Zoho CRM's APIs can be used through the C# SDK, to enable your custom application to perform data sync to the best degree. Here are the sample codes for all the API methods available in our SDK.

## Rest Client Operations

These methods involve authentications procedures that are to be included in your application, to provide access to Zoho CRM's data.

Methods	Description
<a href="#">GetOrganizationDetails</a>	To fetch details of your organization.
<a href="#">GetCurrentUser</a>	To fetch information about the user who is currently accessing Zoho CRM's data through your application.
<a href="#">GetAllModules</a>	To fetch the list of all the modules



	available in your CRM.
<a href="#">GetModule</a>	To fetch all the meta data of modules.

## Bulk Read Operations

Bulk Read API allows you to fetch a large set of data. This API is very useful whenever you require to export a large amount of data or to take a backup of your data without a big impact on your API limits.

This is an asynchronous API, meaning that the response to your request is not available immediately. You will be notified of the callback URL when the data is ready to download or you can choose to check the job status periodically.

### Note

The results are available only as a downloadable CSV file or an ICS file.

You can export the records as an ICS file only for the Events module.

Methods	Description
<a href="#">Create a Bulk Read Job</a>	To schedule a bulk read job to export records that match the criteria
<a href="#">Get Bulk Read Job Details</a>	To know the status of the bulk read job scheduled previously.
<a href="#">Download Result</a>	To download the result of the bulk read job
<a href="#">Download and Get Records</a>	To download the zip file, extract the CSV file, and display the records as instances.



<a href="#">Download and Get Records (for ICS file type)</a>	To download the zip file, extract the ICS file, and display the records as instances.
<a href="#">Get List of Records from the CSV File</a>	To fetch all the records from the CSV file mentioned in the file path.
<a href="#">Get List of Records from the ICS File</a>	To fetch all the records from the ICS file mentioned in the file path.

## Bulk Write Operations

Bulk Write API allows you to insert, update, or upsert a large set of data. This is an asynchronous API, meaning that the response of your request is not available immediately. When the records are processed, the system notifies you of job completion in the callback URL or you can check the job status periodically.

This API allows you to insert, update, or upsert a maximum of 25000 records in a single API call.

Methods	Description
<a href="#">Upload File</a>	To upload a CSV file in ZIP format
<a href="#">Upload File as Stream</a>	To upload the zip containing the CSV file as a stream
<a href="#">Create a Bulk Write Job - Bulk Insert</a>	To create a bulk write job to insert records
<a href="#">Create a Bulk Write Job - Bulk Update</a>	To create a bulk write job to update records



Create a Bulk Write Job - Bulk Upsert	To create a bulk write job to upsert records
Get Bulk Write Job Details	To know the status of the bulk write job scheduled previously
Download Result	To download the result of the bulk write job
Download the CSV file and Get Records	To download the zip file, extract the CSV file, and display the records as instances.
Get Records from File	To fetch all the records from the file mentioned in the file path.
Download the CSV file and Get Failed Records	To download the CSV from the zip file and fetch only the instances of records that failed getting inserted/updated/upserted.
Get Failed Records	To fetch the list of records that failed getting inserted/updated/upserted from the specified file.



## Organization Operations

These methods involve actions that can be performed in your application, to modify the data that pertains to your Zoho CRM's organization. For instance, you can get the list of all the users (employees) that are present in your organization at any point of time.

Methods	Description
<a href="#">GetUser</a>	To fetch information about a specific user in your CRM account.
<a href="#">GetAllUsers</a>	To fetch the list of all the users from your CRM account.
<a href="#">GetAllActiveUsers</a>	To fetch the list of all the active users in your CRM account.
<a href="#">GetAllDeactiveUsers</a>	To fetch the list of all the non-active users in your CRM account.
<a href="#">GetAllConfirmedUsers</a>	To fetch the list of all the confirmed users in your CRM account.
<a href="#">GetAllNotConfirmedUsers</a>	To fetch the list of all the non-confirmed users in your CRM account.
<a href="#">GetAllDeletedUsers</a>	To fetch the list of all the users who were deleted from your CRM account.





<a href="#">GetAllAdminUsers</a>	To fetch the list of all the users who have admin level permissions in your CRM account.
<a href="#">GetAllActiveConfirmedAdmins</a>	To fetch the list of all the users who have admin level permissions and are confirmed, in your CRM account.
<a href="#">GetCurrentUser</a>	To fetch the information about the users who is currently accessing CRM's data through your application.
<a href="#">CreateUser</a>	To create a new user in your CRM account.
<a href="#">UpdateUser</a>	To update details of an existing user in your CRM account.
<a href="#">DeleteUser</a>	To delete a user from your CRM account.
<a href="#">GetAllProfiles</a>	To fetch the list of all the profiles that were created in your CRM account.
<a href="#">GetProfile</a>	To fetch information about a particular profile in your CRM account.
<a href="#">GetAllRoles</a>	To fetch the list of all the roles that were created in your CRM account.
<a href="#">GetRole</a>	To fetch information about a particular role in your CRM account.



<a href="#">GetAllTaxes</a>	To fetch the list of all organization taxes.
<a href="#">GetTax</a>	To fetch information about a particular organization tax.
<a href="#">CreateTaxes</a>	To create a new organization tax.
<a href="#">UpdateTaxes</a>	To update multiple organization taxes.
<a href="#">DeleteTaxes</a>	To delete multiple organization taxes simultaneously.

## Module Operations

These methods involve actions that can be performed in your application, to modify the data in your CRM at the module level. For instance, you can get all the records from a module, search for specific ones, delete them, and do more.

Methods	Description
<a href="#">GetAllModuleFields</a>	To fetch the list of all the fields that are available in a module.
<a href="#">GetLayout</a>	To fetch information about a particular layout of a module.
<a href="#">GetAllModuleLayouts</a>	To fetch the list of all the layouts that are available for a module.
<a href="#">GetCustomView</a>	To fetch information about a particular



	custom view of a module.
<a href="#">GetAllModuleCustomViews</a>	To fetch the list of all the custom views that are available for a module.
<a href="#">GetAllModuleRelatedLists</a>	To fetch the list of all the related lists that are available for a module.
<a href="#">GetALLRecords</a>	To fetch the list of all the records that are available in a module.
<a href="#">SearchByWord</a>	To search for records in a module based on a Word(text).
<a href="#">SearchByPhone</a>	To search for records in a module based on the Phone number.
<a href="#">SearchByEmail</a>	To search for records in a module based on Email address.
<a href="#">SearchByCriteria</a>	To search for records in a module based on a criteria specified by the user.
<a href="#">MassUpdateRecords</a>	To update content in a particular field for multiple records in a module.
<a href="#">UpdateRecords</a>	To update details of multiple records in a module.
<a href="#">CreateRecords</a>	To create a new record in a module.



DeleteRecords	To delete existing records from a module.
GetAllDeletedRecords	To fetch the list of all the records that were deleted from a module.
GetRecycleBinRecords	To fetch the list of all the records that were deleted from a module and stored in the recycle bin.
GetPermanentlyDeletedRecords	To fetch the list of all the records that were permanently deleted from a module.
UpsertRecords	To insert/update records in a module.
GetTags	To fetch the list of all the tags that were created for a module.
GetCount	To fetch total count of the tags that were created for a module.
CreateTags	To create new tags for a module.
UpdateTags	To update details of existing tags for a module.
AddTagMultipleRecord	To associate tags to records in a module.
RemoveTagMultipleRecord	To disassociate tags from records in a module.



## Record Operations

These methods involve actions that can be performed in your application, to access or modify data that are stored in a particular record. You could fetch the details of a record, create new ones, update existing ones, upload notes, attachments, photos, etc.

Methods	Description
<a href="#">GetRecord</a>	To fetch information about a particular record in your CRM.
<a href="#">InsertRecord</a>	To create new records.
<a href="#">UpdateRecord</a>	To update existing records.
<a href="#">DeleteRecord</a>	To delete existing records.
<a href="#">ConvertLead</a>	To convert records(Leads to Contacts/Deals).
<a href="#">GetNotes</a>	To fetch the notes that were attached to a record.
<a href="#">AddNote</a>	To add a note to a record.
	To update a note that was previously



UpdateNote	added to a record.
DeleteNote	To delete a note from a record.
GetAllAttachments	To fetch the list of attachments of a record.
UploadAttachment	To upload an attachment to a record.
UploadAttachmentLink	To upload a link as an attachment to a record.
DownloadAttachment	To download an attachment that was uploaded to a record.
DeleteAttachment	To delete an attachment that was added to a record.
UploadPhoto	To upload a photo to a record.
DownloadPhoto	To download a photo that was added to a record.
DeletePhoto	To delete a photo that was added to a record.
AddRelation	To make a relation between two records.
RemoveRelation	To remove a relation between two records.



<a href="#">AddTagSpecificRecord</a>	To add tags to a specific record.
<a href="#">RemoveTagSpecificRecord</a>	To remove tags from a specific record.
<a href="#">GetRelatedListRecords</a>	To get the list of related list records.

## Note & Tag Operations

These methods involve actions that can be performed in your application, to access or modify notes or tags of data in your CRM.

Methods	Description
<a href="#">NoteAttachment</a>	To upload a note attachment to a record.
<a href="#">DownloadAttachment</a>	To download an attachment that was attached to a record.
<a href="#">DeleteAttachment</a>	To delete an attachment that was attached to a record.
<a href="#">UpdateTag</a>	To update details of an existing tag.
<a href="#">DeleteTag</a>	To delete a tag.
<a href="#">MergeTag</a>	To merge two tags.



## Responses and Exceptions

**APIResponse**, **BulkAPIResponse** and **FileAPIResponse** are the wrapper objects for Zoho CRM APIs' responses. All API calling methods would return one of these two objects.

- A method-seeking entity would return **APIResponse** object, whereas a method-seeking list of entities would return **BulkAPIResponse** object.
- **FileAPIResponse** will be returned for file download APIs to download a photo or an attachment from a record or note such as **record.DownloadPhoto()**, **record.DownloadAttachment(Attachment\_Id)** etc.
- Use the instance variable "**Data**" or "**BulkData**" property to get the entity data alone from the response wrapper objects. **APIResponse.Data** would return a single Zoho CRM entity object, while **BulkAPIResponse.BulkData** would return a list of Zoho CRM entity objects.
- **FileAPIResponse** has two defined methods namely **FileAPIResponse.GetFileName()** which returns the name of the file that is downloaded and **FileAPIResponse.GetFilesAsStream()** that gives the file content as **InputStream**.

### Note:

**BulkAPIResponse** is a generic class. Hence, to get the records, the corresponding type has to be used.

```
1 ZCRMModule module = ZCRMModule.GetInstance("Contacts");
2 BulkAPIResponse<ZCRMRecord> response =
  module.GetRecords()
```





Other than data, these response wrapper objects have the following properties:

1. **ResponseHeaders** - remaining API counts for the present day/window and time elapsed for the present window reset. It is available through:

```
1 response.GetResponseHeaders()
```

2. **ResponseInfo** - any other information, if provided by the API, in addition to the actual data.

```
1 response.Info
```

3. **List<EntityResponse>** - status of individual entities in a bulk API. For example: an insert records API may partially fail because of a few records. This dictionary gives the individual records' creation status. It is available through:

```
1 response.BulkEntitiesResponse
```

## Check Exceptions

All unexpected behaviours like faulty API responses, SDK anomalies are handled by the SDK and are thrown only as a single exception – `ZCRMException`. Hence, it's enough to catch this exception alone in the client app code.

For more clarity in handling the exceptions, you may include the following in your catch statement.

```
1 try
2 {
3 // code block;
4 }
5 catch (ZCRMException ex)
```



```

6 {
7 Console.WriteLine(ex.HttpStatusCode);
8 Console.WriteLine(ex.Code);
9 Console.WriteLine(ex.IsAPIException);
10 Console.WriteLine(ex.IsSDKException);
11 Console.WriteLine(ex.Message);
12 Console.WriteLine(ex.ErrorDetails);
13 Console.WriteLine(ex.ErrorMessage);
14 }

```


## Errors and Solutions

### 1. ClientId is mandatory

```

static void Main(string[] args)
{
    try{
        ZCRMRestClient.Initialize();
        ZCRMRestClient restClientIns = ZCRMRestClient.GetInstance();
        BulkAPIResponse<ZCRMModule> response = restClientIns.GetAllModules();
        foreach (EntityResponse entityResponse in response.BulkEntitiesResponse){
            if (entityResponse.Status.Equals("success")){
                ZCRMModule moduleIns = (ZCRMModule)entityResponse.Data;
                Console.WriteLine(moduleIns.Id);
                Console.WriteLine(moduleIns.ApiName);
            }
        }
    }catch(ZCRMEException e){
        Console.WriteLine(JsonConvert.SerializeObject(e.InnerException));Console.ReadKey();
    }
}

```



#### Reason

Configuration properties not given in the file.

#### Solution

Set configuration properties as displayed in the below image.



```

1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3   <configSections>
4     <section name="oauth_configuration" type="ZCRMSDK.CRM.Library.Common.ConfigFileHandler.ConfigFileSection, ZCRMSDK"></section>
5     <section name="zcrm_configuration" type="ZCRMSDK.CRM.Library.Common.ConfigFileHandler.ConfigFileSection, ZCRMSDK"></section>
6   </configSections>
7   <startup>
8     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
9   </startup>
10  <oauth_configuration>
11    <settings>
12      <add key="client_id" value="1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" />
13      <add key="client_secret" value="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" />
14      <add key="redirect_uri" value="https://www.zoho.com" />
15      <add key="access_type" value="offline" />
16      <add key="iamUrl" value="https://accounts.zoho.com" />
17      <add key="persistence_handler_class" value="ZCRMSDK.OAuth.ClientApp.ZohoOAuthFilePersistence, ZCRMSDK" />
18      <add key="oauth_tokens_file_path" value = "D:/token.txt"/>
19      <add key="mysql_username" value="root" />
20      <add key="mysql_password" value="" />
21      <add key="mysql_database" value="zohooauth" />
22      <add key="mysql_server" value="localhost" />
23      <add key="mysql_port" value="3306" />
24    </settings>
25  </oauth_configuration>
26  <zcrm_configuration>
27    <settings>
28      <add key="apiBaseUrl" value="https://www.zohoapis.com/" />
29      <add key="photoUrl" value="https://profile.zoho.com/api/v1/user/self/photo" />
30      <add key="apiVersion" value="v2" />
31      <add key="logFilePath" value="" />
32      <add key="timeout" value="" />
33      <add key="minLogLevel" value="ALL" />
34      <add key="currentUserEmail" value="usermail@domain.com" />
35      <add key="domainSuffix" value="" />
36    </settings>
37  </zcrm_configuration>

```

## 2. Exception while fetching access tokens from refresh tokens

The screenshot shows the following code snippet in `ZohoOAuthClient`:

```

119 conn.AddParam(ZohoOAuthConstants.REFRESH_TOKEN, refreshToken);
120 string response = conn.Post();
121 JObject responseJSON = JObject.Parse(response);
122 if(responseJSON.ContainsKey(ZohoOAuthConstants.ACCESS_TOKEN)){
123   ZohoOAuthTokens tokens = GetTokensFromJSON(responseJSON);
124   tokens.RefreshToken = refreshToken;
125   tokens.UserMailId = userMailId;
126   ZohoOAuth.GetPersistenceHandlerInstance().SaveOAuthTokens(tokens);
127   return tokens;
128 }
129 throw new ZohoOAuthException("Exception while fetching access tokens from Refresh Token" + response);
130 }catch(WebException e){
131   ZCRMLogger.LogError(e);
132   throw new ZohoOAuthException(e);
133 }
134 }
135 }

```

The **Exception Unhandled** dialog shows the error: `ZCRMSDK.OAuth.Common.ZohoOAuthException: 'Exception while fetching access tokens from Refresh Token' error:'invalid_code'`.

The **Autos** window shows the state of variables:

Name	Value	Type
ZohoOAuthConstants.ACCESS...	"access_token"	string
response	{\"error\": \"invalid_code\"}	string
responseJSON	{\"error\": \"invalid_code\"}	Newtonsoft.Json...
this	{ZCRMSDK.OAuth.Client.ZohoOAuthClient}	ZCRMSDK.OAuth...

The **Output** window shows the following log messages:

```

19-02-19 17:28:22 [INFO]: CH Client Library Configuration Properties : {apiBaseUrl : https
19-02-19 17:28:26 [INFO]: POST - URL = https://accounts.zoho.com/oauth/v2/token, HEADERS =
19-02-19 17:28:26 [INFO]: STATUS_CODE = OK, RESPONSE_JSON = {\"error\": \"invalid_code\"}
'Test.exe' (CLR v4.0.30319: Test.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\
Exception thrown: 'ZCRMSDK.OAuth.Common.ZohoOAuthException' in ZCRMSDK.d11
An unhandled exception of type 'ZCRMSDK.OAuth.Common.ZohoOAuthException' occurred in ZCRMST
'Test.exe' (CLR v4.0.30319: Test.exe): Loaded 'C:\Program Files (x86)\Microsoft Visual Stur
'Test.exe' (CLR v4.0.30319: Test.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\
'Test.exe' (CLR v4.0.30319: Test.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\

```

### Reason

The grant token has expired.

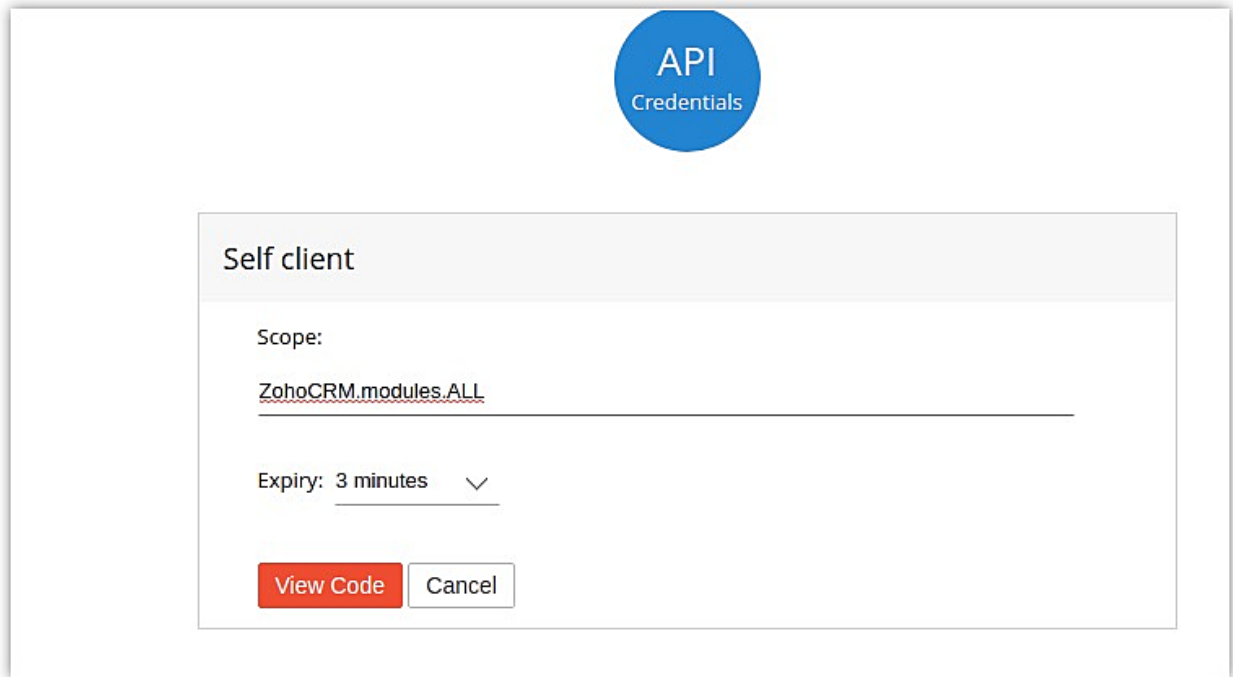
### Solution

Regenerate grant token and use within the stipulated time.



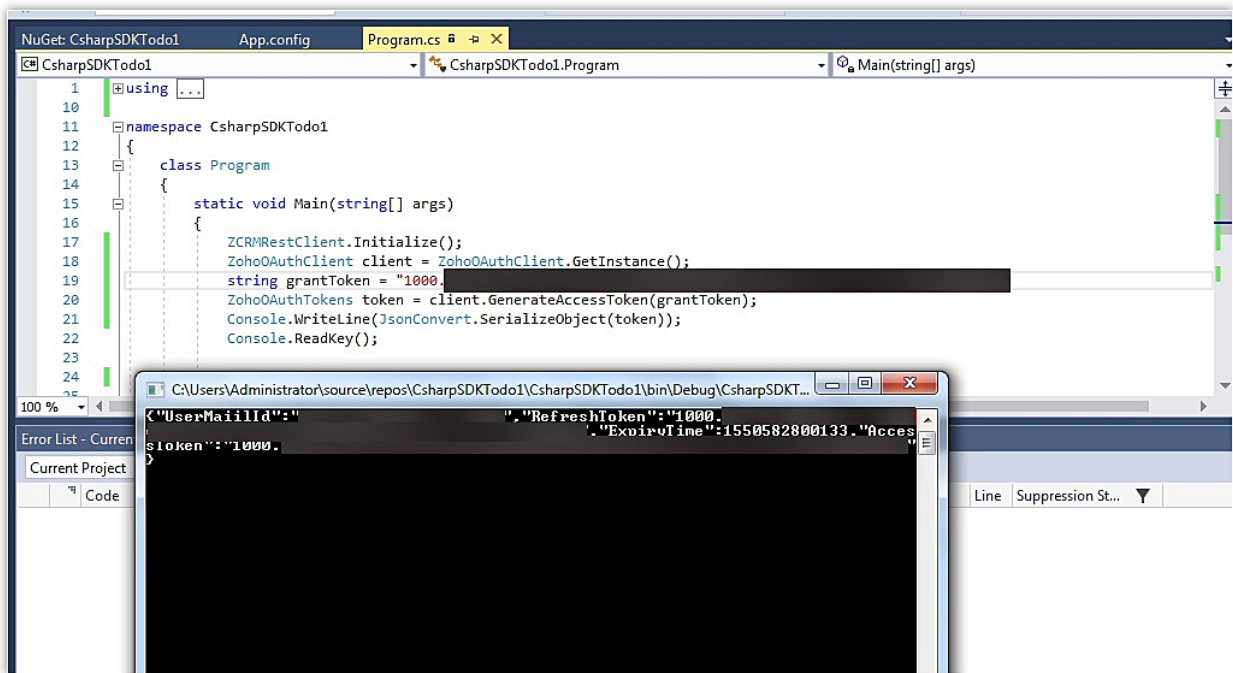


The user has not included the scope “Aaaserver.profile.Read” while generating the grant token(authorization code).

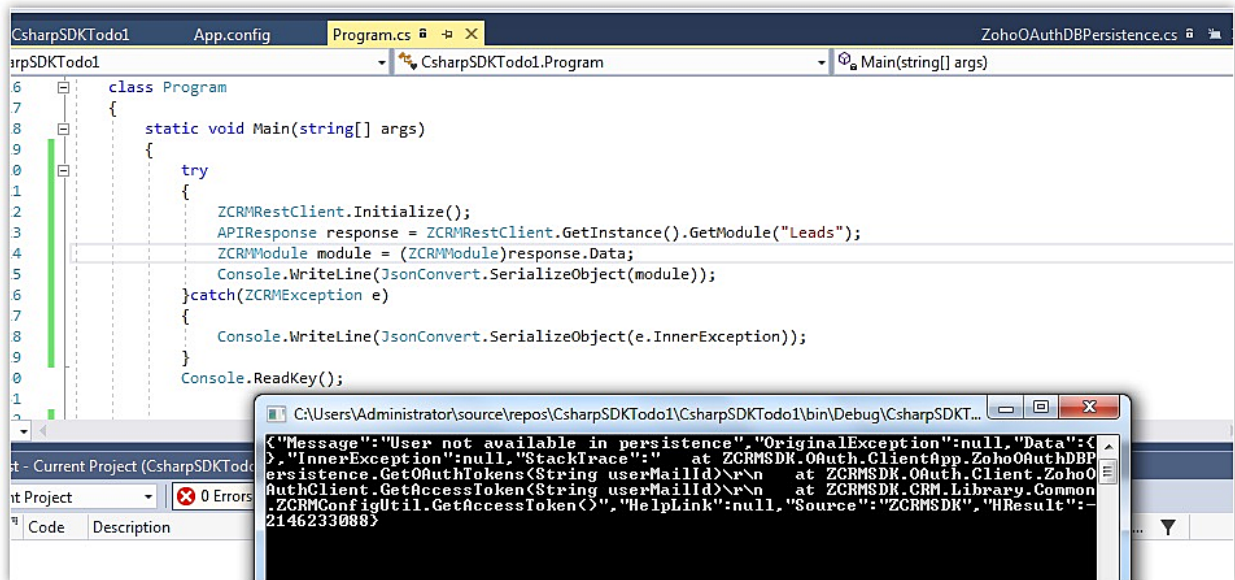


## Solution

Include the scope “Aaaserver.profile.Read” while generating the grant token(authorization code).



## 4. User not available in persistence



### Reason

The provided 'current' user email not found in database. Hence, token wasn't generated.

```
mysql> desc oauthtokens;
+-----+-----+-----+-----+-----+-----+
| Field           | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| useridentifier  | varchar(100)  | YES  |     | NULL     |       |
| accesstoken     | varchar(100)  | YES  |     | NULL     |       |
| refreshtoken    | varchar(100)  | YES  |     | NULL     |       |
| expirytime     | bigint(20)    | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql> select * from oauthtokens;
Empty set (0.00 sec)
```

### Solution

Generate tokens for the respective user.

```
Get: CsharpSDKTodo1 App.config Program.cs* X ZohoOAut
CsharpSDKTodo1 CsharpSDKTodo1.Program Main(string[] args)
11
12 namespace CsharpSDKTodo1
13 {
14     class Program
15     {
16         static void Main(string[] args)
17         {
18             try
19             {
20                 ZCRMRestClient.Initialize();
21                 ZohoOAuthClient client = ZohoOAuthClient.GetInstance();
22                 string grantToken = "1000. ....";
23                 ZohoOAuthTokens token = client.GenerateAccessToken(grantToken);
24                 Console.WriteLine(JsonConvert.SerializeObject(token));
25             }
26         }
27     }
28 }
29
30
31
32
```

```
C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe
4 rows in set (0.04 sec)
mysql> select * from oauthtokens;
Empty set (0.00 sec)
mysql> select * from oauthtokens;
+-----+-----+-----+-----+
| useridentifier | accesstoken | refreshtoken | expirytime |
+-----+-----+-----+-----+
| 1550587139612 |             |              |            |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

ow output from: Deb  
CsharpSDKTodo1.ex  
CsharpSDKTodo1.ex  
CsharpSDKTodo1.ex  
CsharpSDKTodo1.ex  
CsharpSDKTodo1.ex  
he program '[6820

```

using ...

namespace CsharpSDKTodo1
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                ZCRMRestClient.Initialize();
                APIResponse response = ZCRMRestClient.GetInstance().GetModule("Leads");
                Console.WriteLine(JsonConvert.SerializeObject(response.Data));
            }
        }
    }
}

```

```

{"ApiName": "Leads", "Id": "...", "SystemName": "Leads", "SingularLabel": "Lead", "PluralLabel": "Leads", "Convertable": true, "Deletable": true, "Editable": true, "Viewable": true, "CustomModule": false, "ModifiedBy": {"Id": "...", "Name": "Administrator"}, "Zuid": null, "FirstName": null, "LastName": null, "FullName": "Raja", "EmailId": null, "Mobile": null, "Language": null, "Alias": null, "City": null, "Country": null, "CountryLocale": null, "Fax": null, "DateFormat": null, "TimeFormat": null, "DateOfBirth": null, "Locale": null, "NameFormat": null, "Phone": null, "State": null, "Confirm": false, "Street": null, "TimeZone": null, "Website": null, "Status": null, "Profile": null, "Role": null, "CreatedBy": null, "CreatedTime": null, "ModifiedBy": null, "ModifiedTime": null, "ReportingTo": null, "ModifiedTime": "11/15/2018 18:55:20", "BusinessCardFields": [], "RelatedLists": [], "Creatable": true, "AccessibleProfiles": [{"Id": "...", "Name": "Administrator"}, {"Id": "...", "Name": "Standard"}], "Category": false, "Description": null}, {"Id": "...", "Name": "Standard", "CreatedBy": null, "ModifiedBy": null, "CreatedTime": null, "ModifiedTime": null, "Category": false, "Description": null}], "ApiSupported": true}

```

## 5. Current user must be set in ZCRMRestClient or ZCRMConfiguration

```

class Program
{
    static void Main(string[] args)
    {
        try
        {
            ZCRMRestClient.Initialize();
            APIResponse response = ZCRMRestClient.GetInstance().GetModule("Leads");
            Console.WriteLine(JsonConvert.SerializeObject(response.Data));
        }
        catch (ZCRMException e)
        {
            Console.WriteLine(JsonConvert.SerializeObject(e));
        }
    }
}

```

```

{"Message": "Current user must be either set in ZCRMRestClient or zcrm_configuration section in zoho_configuration.config", "Data": {}, "InnerException": null, "StackTrace": " at ZCRMSDK.CRM.Library.Api.APIRequest.AuthenticateRequest() at ZCRMSDK.CRM.Library.Api.APIRequest.GetResponseFromServer() at ZCRMSDK.CRM.Library.Api.APIRequest.GetAPIResponse() at ZCRMSDK.CRM.Library.Api.Handler.MetadataAPIHandler.GetModule(String apiName) at ZCRMSDK.CRM.Library.Setup.RestClient.ZCRMRestClient.GetModule(String moduleName) at CsharpSDKTodo1.Program.Main(String[] args) in C:\\Users\\Administrator\\source\\repos\\CsharpSDKTodo1\\CsharpSDKTodo1\\Program.cs:line 22", "HelpLink": null, "Source": "ZCRMSDK", "HResult": -2146233088}

```



## Reason

The currentUserEmail must have been left empty.

```
NuGet: CsharpSDKTodo1 App.config Program.cs ZCRMConfid
15 <add key="access_type" value="Online" />
16 <add key="iamUrl" value="https://accounts.zoho.com" />
17 <add key="persistence_handler_class" value="ZCRMSDK.OAuth.ClientApp.ZohoOAuthDBPersistence, ZCRMSDK" />
18 <add key="oauth_tokens_file_path" value="D:/token.txt"/>
19 <add key="mysql_username" value="root" />
20 <add key="mysql_password" value="" />
21 <add key="mysql_database" value="zohooauth" />
22 <add key="mysql_server" value="localhost" />
23 <add key="mysql_port" value="3306" />
24 </settings>
25 </oauth_configuration>
26 <zcrm_configuration>
27 <settings>
28 <add key="apiBaseUrl" value="https://www.zohoapis.com" />
29 <add key="photoUrl" value="https://profile.zoho.com/api/v1/user/self/photo" />
30 <add key="apiVersion" value="v2" />
31 <add key="logFilePath" value="" />
32 <add key="timeout" value="" />
33 <add key="minLogLevel" value="ALL" />
34 <add key="currentUserEmail" value="" />
35 <add key="domainSuffix" value="" />
36 </settings>
37 </zcrm_configuration>
```

## Solution

Specify the currentUserEmail.

```
NuGet: CsharpSDKTodo1 App.config Program.cs ZCRMConfid
15 <add key="access_type" value="Online" />
16 <add key="iamUrl" value="https://accounts.zoho.com" />
17 <add key="persistence_handler_class" value="ZCRMSDK.OAuth.ClientApp.ZohoOAuthDBPersistence, ZCRMSDK" />
18 <add key="oauth_tokens_file_path" value="D:/token.txt"/>
19 <add key="mysql_username" value="root" />
20 <add key="mysql_password" value="" />
21 <add key="mysql_database" value="zohooauth" />
22 <add key="mysql_server" value="localhost" />
23 <add key="mysql_port" value="3306" />
24 </settings>
25 </oauth_configuration>
26 <zcrm_configuration>
27 <settings>
28 <add key="apiBaseUrl" value="https://www.zohoapis.com" />
29 <add key="photoUrl" value="https://profile.zoho.com/api/v1/user/self/photo" />
30 <add key="apiVersion" value="v2" />
31 <add key="logFilePath" value="" />
32 <add key="timeout" value="" />
33 <add key="minLogLevel" value="ALL" />
34 <add key="currentUserEmail" value="usermail@domain.com" />
35 <add key="domainSuffix" value="" />
36 </settings>
37 </zcrm_configuration>
```



```

CsharpSDKTodo1 App.config Program.cs
CsharpSDKTodo1 Program Main(string[] args)
using ...
namespace CsharpSDKTodo1
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                ZCRMRestClient.Initialize();
                APIResponse response = ZCRMRestClient.GetInstance().GetModule("Leads");
                Console.WriteLine(JsonConvert.SerializeObject(response.Data));
            }
        }
    }
}

```

```

{"ApiName":"Leads","Id":null,"SystemName":"Leads","SingularLabel":"Lead","PluralLabel":"Leads","Convertible":true,"Deletable":true,"Editable":true,"Viewable":true,"CustomModule":false,"ModifiedBy":{"Id":null,"FirstName":null,"LastName":null,"FullName":"Raja","EmailId":null,"Mobile":null,"Language":null,"Alias":null,"City":null,"Country":null,"CountryLocale":null,"Fax":null,"DateFormat":null,"TimeFormat":null,"DateOfBirth":null,"Locale":null,"NameFormat":null,"Phone":null,"State":null,"Confirm":false,"Street":null,"TimeZone":null,"Website":null,"Status":null,"Profile":null,"Role":null,"CreatedBy":null,"CreatedTime":null,"ModifiedBy":null,"ModifiedTime":null,"ReportingTo":null,"ModifiedTime":"11/15/2018 18:55:20","BusinessCardFields":[],"RelatedLists":[],"Creatable":true,"AccessibleProfiles":[{"Id":null,"Name":"Administrator","CreatedBy":null,"ModifiedBy":null,"CreatedTime":null,"ModifiedTime":null,"Category":false,"Description":null}],"Standard":true,"CreatedBy":null,"ModifiedBy":null,"CreatedTime":null,"ModifiedTime":null,"Category":false,"Description":null},"ApiSupported":true}

```

## 6. Invalid URI format

```

NuGet: CsharpSDKTodo1 App.config Program.cs
CsharpSDKTodo1 Program Main(string[] args)
namespace CsharpSDKTodo1
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                APIResponse response = ZCRMRestClient.GetInstance().GetModule("Leads");
                Console.WriteLine(JsonConvert.SerializeObject(response.Data));
            }
            catch (ZCRMException e)
            {
                Console.WriteLine(e.Message);
            }
        }
    }
}

```

Exception Unhandled

**System.UriFormatException: 'Invalid URI: The format of the URI could not be determined.'**

Output

Show output from: Debug

```

C:\Users\Administrator\source\repos\CsharpSDKTodo1\CsharpSDKTodo1\bin\Debug\CsharpSDKTodo1.exe (CLR v4.0.30319: DefaultDomain): Loaded 'C:\Users\Administrator\source\repos\CsharpSDKTodo1\CsharpSDKTodo1.exe'.
C:\Users\Administrator\source\repos\CsharpSDKTodo1\CsharpSDKTodo1\bin\Debug\CsharpSDKTodo1.exe (CLR v4.0.30319: CsharpSDKTodo1.exe): Loaded 'C:\Users\Administrator\source\repos\CsharpSDKTodo1\CsharpSDKTodo1.exe'.
C:\Users\Administrator\source\repos\CsharpSDKTodo1\CsharpSDKTodo1\bin\Debug\CsharpSDKTodo1.exe (CLR v4.0.30319: CsharpSDKTodo1.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\netsync\mscorlib.resources.dll'.
C:\Users\Administrator\source\repos\CsharpSDKTodo1\CsharpSDKTodo1\bin\Debug\CsharpSDKTodo1.exe (CLR v4.0.30319: CsharpSDKTodo1.exe): Loaded 'C:\Users\Administrator\source\repos\CsharpSDKTodo1\CsharpSDKTodo1.exe'.
C:\Users\Administrator\source\repos\CsharpSDKTodo1\CsharpSDKTodo1\bin\Debug\CsharpSDKTodo1.exe (CLR v4.0.30319: CsharpSDKTodo1.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System\mscorlib.dll'.
C:\Users\Administrator\source\repos\CsharpSDKTodo1\CsharpSDKTodo1\bin\Debug\CsharpSDKTodo1.exe (CLR v4.0.30319: CsharpSDKTodo1.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System\mscorlib.resources.dll'.
Exception thrown: 'System.UriFormatException' in System.dll
An unhandled exception of type 'System.UriFormatException' occurred in System.dll
Invalid URI: The format of the URI could not be determined.

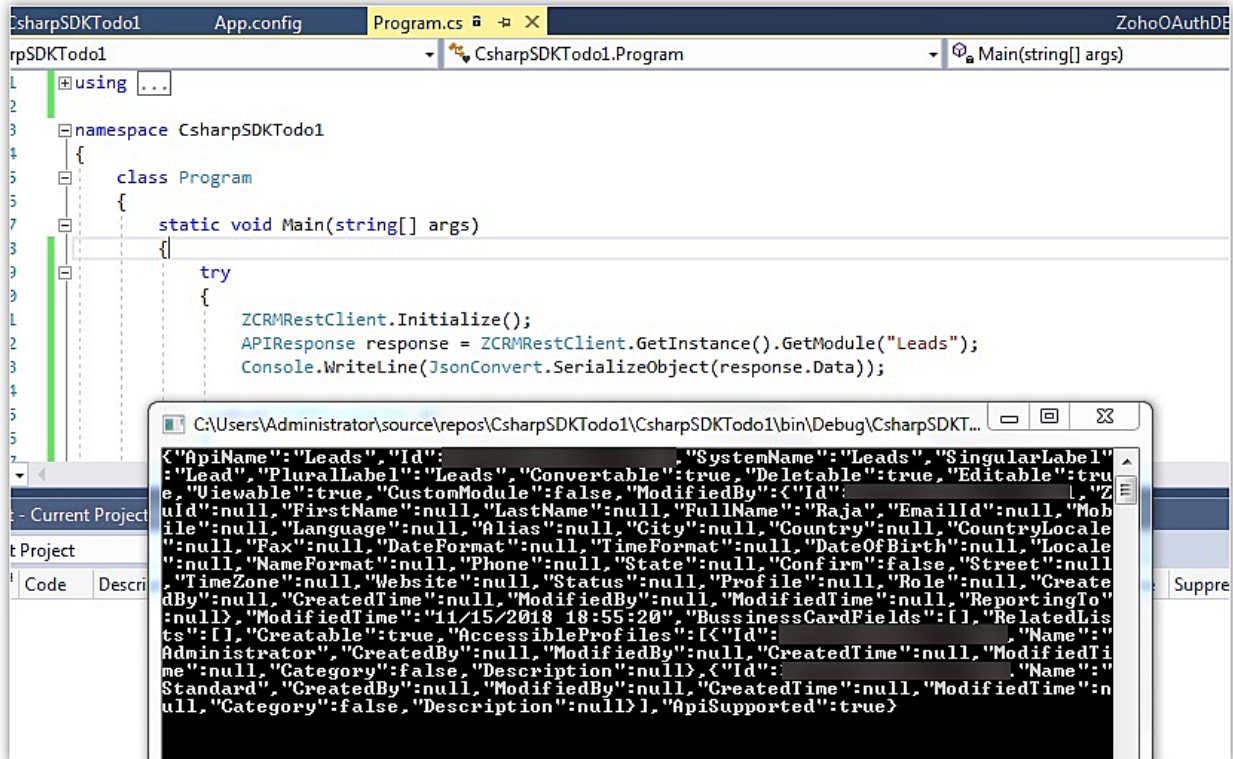
```

## Reason

The SDK is not initialized by not calling `ZCRMRestClient::initialize()`.

## Solution

Follow the steps mentioned in Initialization - C# SDK to initialize the SDK.



```
using ...
namespace CsharpSDKTodo1
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                ZCRMRestClient.Initialize();
                APIResponse response = ZCRMRestClient.GetInstance().GetModule("Leads");
                Console.WriteLine(JsonConvert.SerializeObject(response.Data));
            }
        }
    }
}
```

```
<"ApiName":"Leads","Id":null,"SystemName":"Leads","SingularLabel":"Lead","PluralLabel":"Leads","Convertible":true,"Deletable":true,"Editable":true,"Viewable":true,"CustomModule":false,"ModifiedBy":{"Id":null,"Zuid":null,"FirstName":null,"LastName":null,"FullName":"Raja","EmailId":null,"Mobile":null,"Language":null,"Alias":null,"City":null,"Country":null,"CountryLocale":null,"Fax":null,"DateFormat":null,"TimeFormat":null,"DateOfBirth":null,"Locale":null,"NameFormat":null,"Phone":null,"State":null,"Confirm":false,"Street":null,"TimeZone":null,"Website":null,"Status":null,"Profile":null,"Role":null,"CreatedBy":null,"CreatedTime":null,"ModifiedBy":null,"ModifiedTime":null,"ReportingTo":null,"ModifiedTime":"11/15/2018 18:55:20","BusinessCardFields":{"RelatedLists":{"Creatable":true,"AccessibleProfiles":[{"Id":null,"Name":"Administrator","CreatedBy":null,"ModifiedBy":null,"CreatedTime":null,"ModifiedTime":null,"Category":false,"Description":null}],"Standard":{"Id":null,"Name":"Standard","CreatedBy":null,"ModifiedBy":null,"CreatedTime":null,"ModifiedTime":null,"Category":false,"Description":null}},"ApiSupported":true}
```

## 8. Unable to generate access token



```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="oauth_configuration" type="ZCRMSDK.CRM.Library.Common.ConfigFileHandler.ConfigFileSection, ZCRMSDK"></s
    <section name="zcrm_configuration" type="ZCRMSDK.CRM.Library.Common.ConfigFileHandler.ConfigFileSection, ZCRMSDK"></se
  </configSections>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <oauth_configuration>
    <settings>
      <add key="client_id" value="1000 [REDACTED]" />
      <add key="client_secret" value="[REDACTED]" />
      <add key="redirect_uri" value="https://www.google.com" />
      <add key="access_type" value="offline" />
      <add key="iamURL" value="https://accounts.zoho.eu" />
      <add key="persistence_handler_class" value="ZCRMSDK.OAuth.ClientApp.ZohoOAuthDBPersistence, ZCRMSDK" />
      <add key="oauth_tokens_file_path" value = "D:/token.txt"/>
      <add key="mysql_username" value="root" />
      <add key="mysql_password" value="" />
      <add key="mysql_database" value="zohooauth" />
      <add key="mysql_server" value="localhost" />
      <add key="mysql_port" value="3306" />
    </settings>
  </oauth_configuration>
</configuration>
```

## Solution

Generate both grant token and access token from the same domain.

```
NuGet: CsharpSDKTodo1 App.config Program.cs
1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3   <configSections>
4     <section name="oauth_configuration" type="ZCRMSDK.CRM.Library.Common.ConfigFileHandl
5     <section name="zcrm_configuration" type="ZCRMSDK.CRM.Library.Common.ConfigFileHandl
6   </configSections>
7   <startup>
8     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
9   </startup>
10  <oauth_configuration>
11    <settings>
12      <add key="client_id" value="1000 [REDACTED]" />
13      <add key="client_secret" value="[REDACTED]" />
14      <add key="redirect_uri" value="https://www.google.com" />
15      <add key="access_type" value="offline" />
16      <add key="iamURL" value="https://accounts.zoho.com" />
17      <add key="persistence_handler_class" value="ZCRMSDK.OAuth.ClientApp.ZohoOAuthDBPer
18      <add key="oauth_tokens_file_path" value="D:/token.txt" />
19      <add key="mysql_username" value="root" />
20      <add key="mysql_password" value="" />
21      <add key="mysql_database" value="zohooauth" />
22      <add key="mysql_server" value="localhost" />
23      <add key="mysql_port" value="3306" />
24    </settings>
  </oauth_configuration>
</configuration>
```



```

119 conn.AddParam(ZohoOAuthConstants.REFRESH_TOKEN, refreshToken);
120 string response = conn.Post();
121 JObject responseJSON = JObject.Parse(response);
122 if(responseJSON.ContainsKey(ZohoOAuthConstants.ACCESS_TOKEN)){
123     ZohoOAuthTokens tokens = GetTokensFromJSON(responseJSON);
124     tokens.RefreshToken = refreshToken;
125     tokens.UserMailId = userMailId;
126     ZohoOAuth.GetPersistenceHandlerInstance().SaveOAuthTokens(tokens);
127     return tokens;
128 }
129 throw new ZohoOAuthException("Exception while fetching access tokens from Refresh Token" + response);
130 }catch(WebException e){
131     ZCRMLogger.LogError(e);
132     throw new ZohoOAuthException(e);
133 }
134 }
135 }

```

Exception Unhandled  
ZCRMSDK.OAuth.Common.ZohoOAuthException: 'Exception while fetching access tokens from Refresh Token' error:'invalid\_code'

Autos

Name	Value	Type
ZohoOAuthConstants.ACCESS...	"access_token"	string
response	"{"error":"invalid_code"}"	string
responseJSON	{ "error": "invalid_code" }	Newtonsoft.Json...
this	{ZCRMSDK.OAuth.Client.ZohoOAuthClient}	ZCRMSDK.OAuth...

Output

```

19-02-19 17:28:22 [INFO]: C# Client Library Configuration Properties : {apiBaseUrl : https
19-02-19 17:28:22 [INFO]: POST - URL = https://accounts.zoho.com/oauth/v2/token, HEADERS =
19-02-19 17:28:26 [INFO]: STATUS_CODE = OK, RESPONSE_JSON = {"error":"invalid_code"}
'Test.exe' (CLR v4.0.30319: Test.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\
Exception thrown: 'ZCRMSDK.OAuth.Common.ZohoOAuthException' in ZCRMSDK.dll
An unhandled exception of type 'ZCRMSDK.OAuth.Common.ZohoOAuthException' occurred in ZCRMS
'Test.exe' (CLR v4.0.30319: Test.exe): Loaded 'C:\Program Files (x86)\Microsoft Visual Stur
'Test.exe' (CLR v4.0.30319: Test.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\
'Test.exe' (CLR v4.0.30319: Test.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\

```

## 9. Invalid OAuth token

```

14 {
15     class Program
16     {
17         static void Main(string[] args)
18         {
19             try
20             {
21                 ZCRMRestClient.Initialize();
22                 APIResponse response = ZCRMRestClient.GetInstance().GetModule("Leads");
23                 Console.WriteLine(JsonConvert.SerializeObject(response.Data));
24             }
25             catch(ZCRMSDK.CRM.Library.CRMException e)
26             {
27                 Console.WriteLine(e.Message);
28             }
29             Console.ReadKey();
30         }
31     }
32 }

```

Output

```

CsharpSDKTodo1.exe (CLR v4.0.30319:
CsharpSDKTodo1.exe (CLR v4.0.30319:
9-02-20 14:38:34 [INFO]: Token fetched successfully..
9-02-20 14:38:34 [INFO]: URL=https://www.zohoapis.eu/crm/v2/settings/modules/Leads ,HEADERS = {Authorization : ## can't disclose ##
CsharpSDKTodo1.exe (CLR v4.0.30319: CsharpSDKTodo1.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Numerics\v4.0.4
9-02-20 14:38:35 [ERROR]: INVALID_TOKEN invalid oauth token
Exception thrown: 'ZCRMSDK.CRM.Library.CRMException.ZCRMException' in ZCRMSDK.dll

```

Stack Trace:

```

{"Message":"invalid oauth token","Data":{},"InnerException":null,"StackTrace":
at ZCRMSDK.CRM.Library.Api.Response.APIResponse.HandleFaultyResponse()
at ZCRMSDK.CRM.Library.Api.Response.CommonAPIResponse.ProcessResponse()
at ZCRMSDK.CRM.Library.Api.Response.CommonAPIResponse..ctor(HttpWebResponse response)
at ZCRMSDK.CRM.Library.Api.Response.APIResponse..ctor(HttpWebResponse response)
at ZCRMSDK.CRM.Library.Api.APIRequest.GetAPIResponse()
at ZCRMSDK.CRM.Library.Api.Handler.MetadataAPIHandler.GetModule(String apiName)
at ZCRMSDK.CRM.Library.Setup.RestClient.ZCRMRestClient.GetModule(String module Name)
at CsharpSDKTodo1.Program.Main(String[] args) in C:\Users\Administrator\source\repos\CsharpSDKTodo1\CsharpSDKTodo1\Program.cs:line 22,"HelpLink":null,"Source":"ZCRMSDK","HResult":-2146233088}

```

### Reason

The user has generated access token and is trying to access module record from different domains. For example, the user generates the access token from accounts.zoho.com and tries to fetch module record data from accounts.zoho.eu.



```
App.config  Program.cs
23     <add key="mysql_port" value="3306" />
24     </settings>
25 </oauth_configuration>
26 <zcrm_configuration>
27   <settings>
28     <add key="apiBaseUrl" value="https://www.zohoapis.eu" />
29     <add key="photoUrl" value="https://profile.zoho.com/api/v1/user/self/photo" />
30     <add key="apiVersion" value="v2" />
31     <add key="logFilePath" value="" />
32     <add key="timeout" value="" />
33     <add key="minLogLevel" value="ALL" />
34     <add key="currentUserEmail" value="rajaprabu26@gmail.com" />
35     <add key="domainSuffix" value="" />
36   </settings>
37 </zcrm_configuration>
38 </runtime>
```

### Solution

Use the same domain.

```
p.config  Program.cs
23     <add key="mysql_port" value="3306" />
24     </settings>
25 </oauth_configuration>
26 <zcrm_configuration>
27   <settings>
28     <add key="apiBaseUrl" value="https://www.zohoapis.com" />
29     <add key="photoUrl" value="https://profile.zoho.com/api/v1/user/self/photo" />
30     <add key="apiVersion" value="v2" />
31     <add key="logFilePath" value="" />
32     <add key="timeout" value="" />
33     <add key="minLogLevel" value="ALL" />
34     <add key="currentUserEmail" value="rajaprabu26@gmail.com" />
35     <add key="domainSuffix" value="" />
36   </settings>
37 </zcrm_configuration>
```

```

14 {
15     class Program
16     {
17         static void Main(string[] args)
18         {
19             try
20             {
21                 ZCRMRestClient.Initialize();
22                 APIResponse response = ZCRMRestClient.GetInstance().GetModule("Leads");
23                 Console.WriteLine(JsonConvert.SerializeObject(response.Data));
24             }
25             catch (ZCRMException e)
26             {
27                 Console.R
28             }
29         }
30     }

```

```

{"ApiName": "Leads", "Id": "1", "SystemName": "Leads", "SingularLabel": "Lead", "PluralLabel": "Leads", "Convertible": true, "Deletable": true, "Editable": true, "Viewable": true, "CustomModule": false, "ModifiedBy": "<Id>:3", "CreatedTime": "11/15/2018 18:55:20", "ModifiedTime": "11/15/2018 18:55:20", "BussinessCardFields": 1, "RelatedLists": 1, "Creatable": true, "AccessibleProfiles": [{"Id": "Administrator", "CreatedBy": null, "ModifiedBy": null, "CreatedTime": null, "ModifiedTime": null, "Category": false, "Description": null, "<Id>": "1"}]}

```

```

19-02-20 14:41:32 [INFO]: URL=https://www.zohoapis.com/crm/v2/settings/modules/Leads ,HEADERS = {Authorization : ## can't disclose
'CsharpSDKTodo1.exe' (CLR v4.0.30319: CsharpSDKTodo1.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Numerics\v4.0
19-02-20 14:41:34 [INFO]: STATUS_CODE=OK, RESPONSE_JSON={ "modules": [ { "global_search_supported": true, "kanban_view": false, "deletable"
'CsharpSDKTodo1.exe' (CLR v4.0.30319: CsharpSDKTodo1.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Runtime.Serial

```

## 10. Unable to connect to remote server

```

14 {
15     public static Dictionary<string, string> config = new Dictionary<string, string>()
16     {
17         {"client_id", "1008xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"},
18         {"client_secret", "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"},
19         {"redirect_uri", "https://crm.zoho.com/crm/"},
20         {"access_type", "offline"},
21         {"persistence_handler_class", "ZCRMSDK.OAuth.ClientApp.ZohoOAuthDBPersistence, ZCRMSDK"},
22         {"mysql_username", "root"},
23         {"mysql_password", ""},
24         {"mysql_database", "zohooauth"},
25         {"mysql_server", "localhost"},
26         {"mysql_port", "3306"},
27         {"sandbox", "false"},
28         {"apiBaseUrl", "https://www.zohoapis.com/"},
29         {"iamURL", "https://accounts.zoho.com"},
30         {"photoURL", "https://profile.zoho.com/api/v1/user/self/photo"},
31         {"apiVersion", "v2"},
32         {"currentUserEmail", ""}
33     };
34     static void Main(string[] args)
35     {

```

```

'sampletest.vshost.exe' (CLR v4.0.30319: sampletest.vshost.exe): Loaded 'E:\sampletest\bin\Debug\MySQL.Data.dll'. Skipped loading symbols. Module is optimized an
'sampletest.vshost.exe' (CLR v4.0.30319: sampletest.vshost.exe): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_32\System.EnterpriseServices\v4.0.4.0.0_0_b03f5f7
'sampletest.vshost.exe' (CLR v4.0.30319: sampletest.vshost.exe): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_32\System.EnterpriseServices\v4.0.4.0.0_0_b03f5f7
19-05-20 14:59:02 [INFO]: Token fetched successfully..
19-05-20 14:59:02 [INFO]: URL=https://www.zohoapis.com/crm/v2/Contacts?page=1&per_page=200 ,HEADERS = {Authorization : ## can't disclose ##} ,PARAMS = { page : 1
19-05-20 14:59:25 [ERROR]: Exception in Type: HTTPWebRequest at Method: System.Net.WebResponse.GetResponse() in File: at Line: 0

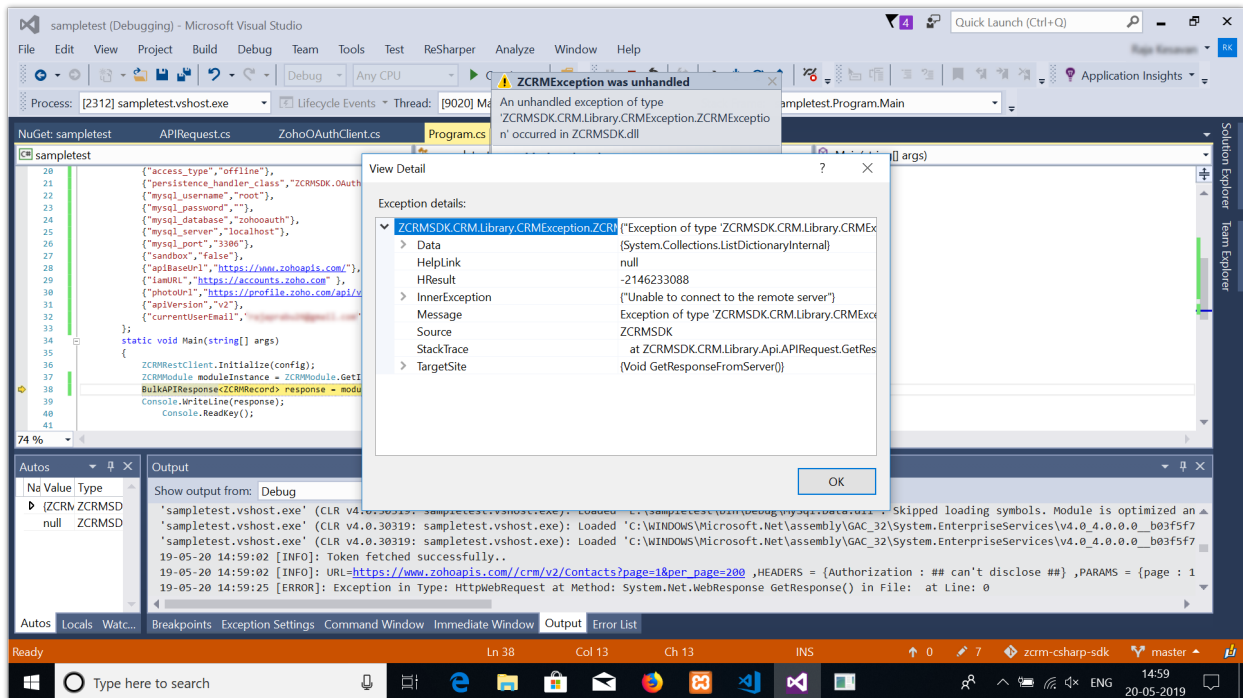
```

### Reason

The apiBaseUrl key in the configuration dictionary must have had the value as "https://www.zohoapis.com/". The slash at the end of the line is automatically added by



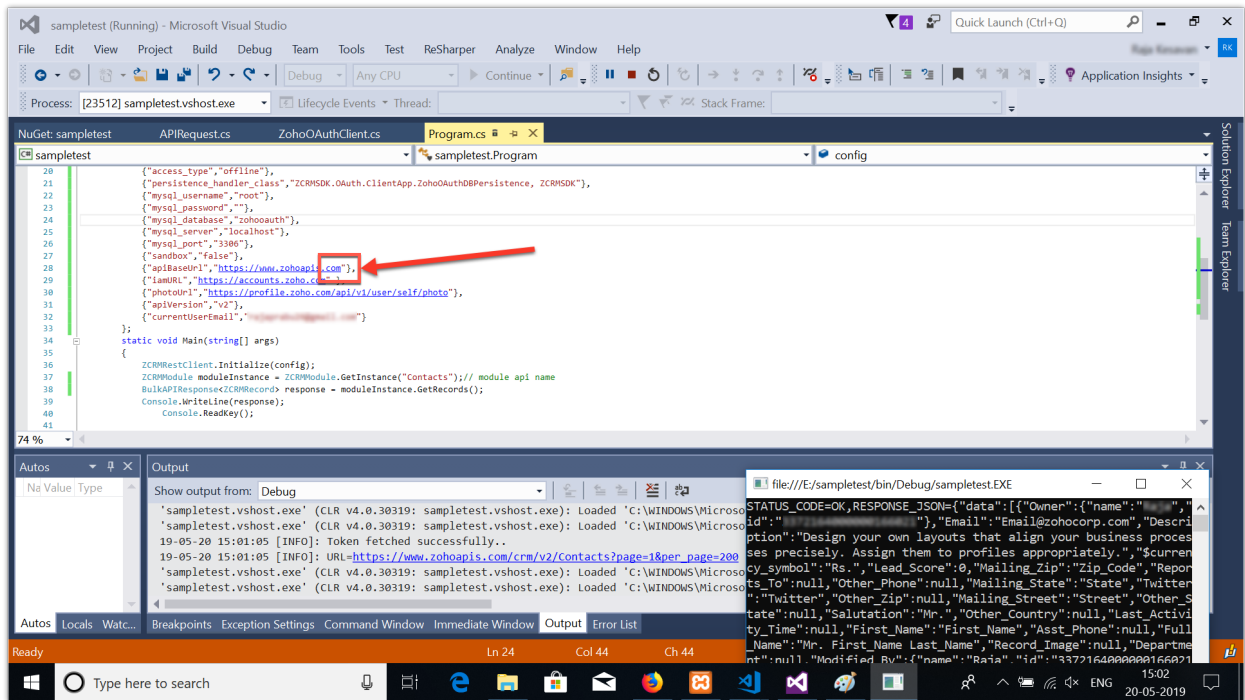
the SDK. Hence if the user adds a slash as well, the final input to the SDK will have two slashes. Ex: "https://www.zohoapis.com//".



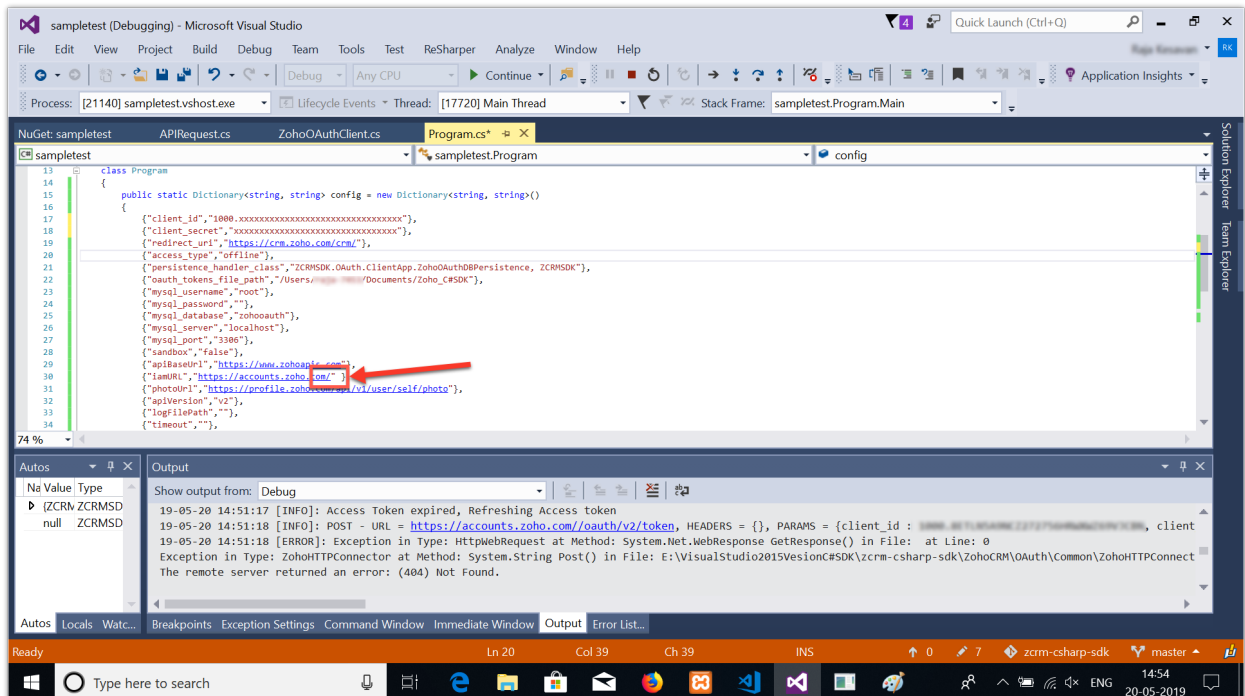
## Solution

Remove the (/) slash after https://www.zohoapis.com in the "apiBaseUrl" key of the configuration dictionary.



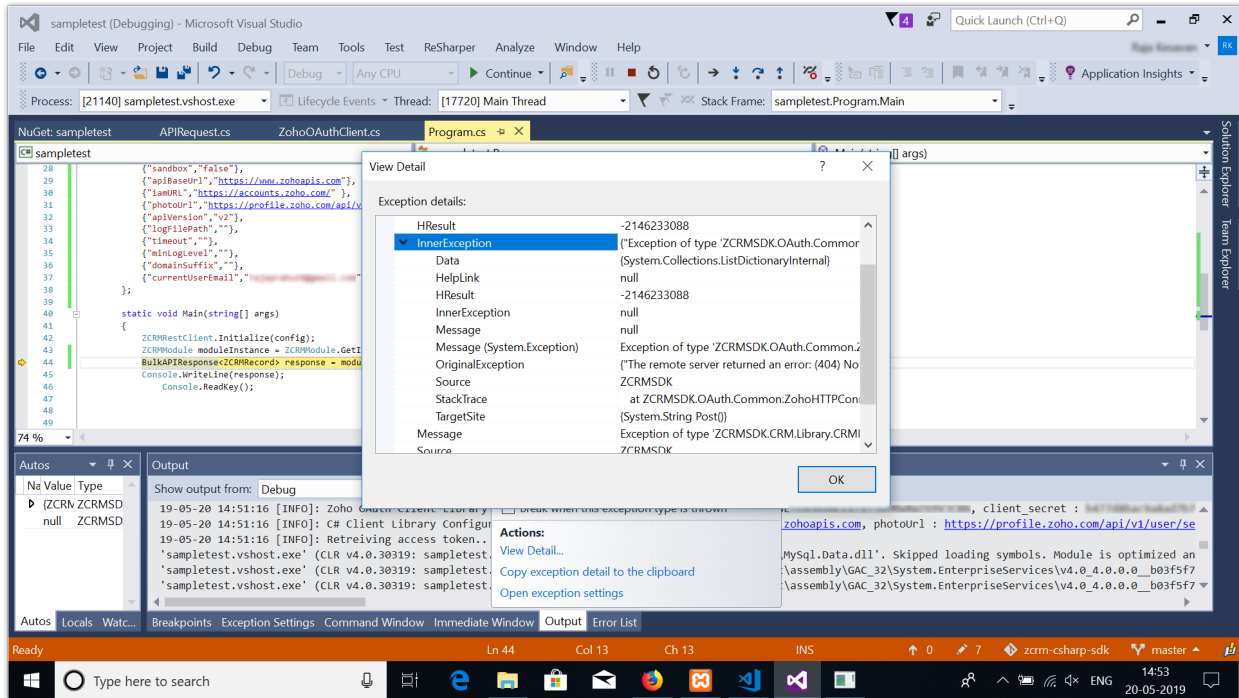


## 11. The remote server returned an error: 404 not found



## Reason

The iamURL key in the configuration dictionary must have had the value as "https://accounts.zoho.com/". The slash at the end of the line is automatically added by the SDK. Hence if the user adds a slash as well, the final input to the SDK will have two slashes. Ex: "https://accounts.zoho.com//".



## Solution

Remove the (/) slash after https://accounts.zoho.com in the "iamURL" key of the configuration dictionary.



The screenshot shows the Visual Studio IDE with the 'Program.cs' file open. The code defines a configuration dictionary for the OAuth client. A red arrow points to the 'iamURL' property, which is set to 'https://accounts.zoho.com'. The 'Output' window shows the following log messages:

```

19-05-20 14:58:14 [INFO]: Access Token expired, Refreshing Access token
19-05-20 14:58:15 [INFO]: POST - URL = https://accounts.zoho.com/oauth/v2/token, HEADERS = {}, PARAMS = {client_id = 3000_8E1L5A9MKZ27750H9G6G209V3CBM, client_
19-05-20 14:58:15 [INFO]: STATUS_CODE = OK, RESPONSE_JSON = {"access_token": "1000_52402112a1279a4b14aF97430961aF28_eF9a0F80677515ee3002a6294a744aac", "expires_in
'sampletest.vshost.exe' (CLR v4.0.30319: sampletest.vshost.exe): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Numerics\v4.0.4.0.0_b77a5c561934e08
19-05-20 14:58:16 [INFO]: Token fetched successfully..
  
```

## 12. Unable to connect to remote server

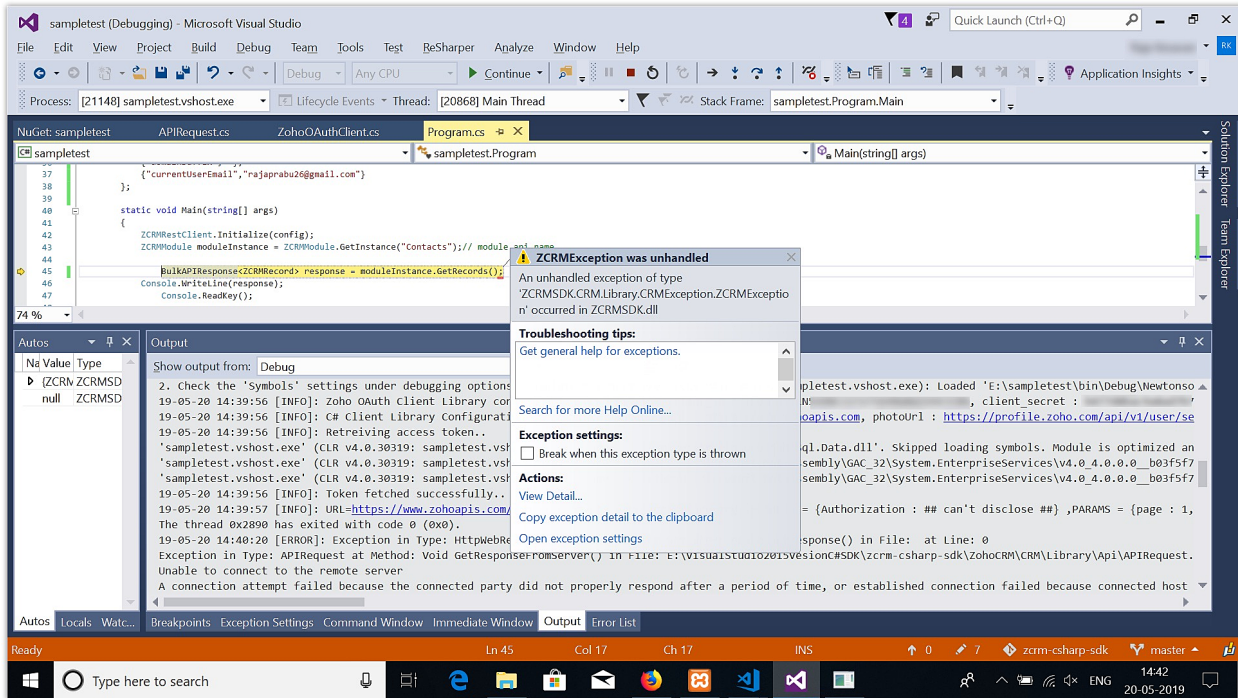
The screenshot shows the Visual Studio IDE with the 'Program.cs' file open. The code defines a configuration dictionary for the OAuth client. A red arrow points to the 'iamURL' property, which is set to 'https://accounts.zoho.com'. The 'Output' window shows the following log messages:

```

2. Check the 'Symbols' settings under debugging options. 'sampletest.vshost.exe' (CLR v4.0.30319: sampletest.vshost.exe): Loaded 'E:\sampletest\bin\Debug\Newtonso
19-05-20 14:39:56 [INFO]: Zoho OAuth Client Library configuration properties: {client_id = , client_secret :
19-05-20 14:39:56 [INFO]: C# Client Library Configuration Properties : {apiBaseUrl = https://www.zohoapis.com, photoUrl = https://profile.zoho.com/api/v1/user/se
19-05-20 14:39:56 [INFO]: Retrieving access token..
'sampletest.vshost.exe' (CLR v4.0.30319: sampletest.vshost.exe): Loaded 'E:\sampletest\bin\Debug\MvSql.Data.dll'. Skipped loading symbols. Module is optimized an
  
```

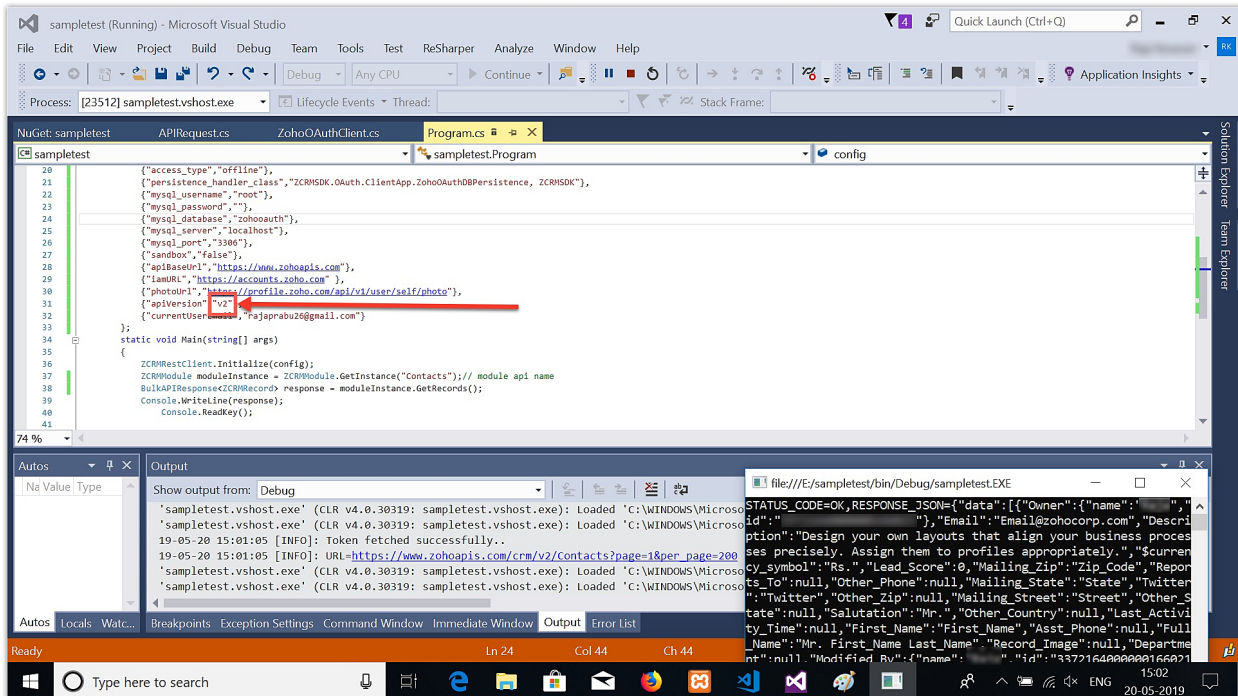
## Reason

The "apiVersion" key in the configuration dictionary is "V2", with a capital "V".



## Solution

Change the value of apiVersion to "v2" in the configuration dictionary.



# Release Notes

## Current Version

### 1. ZCRMSDK - VERSION 2.2.4

#### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.2.4
2 dotnet add package ZCRMSDK -- VERSION 2.2.4
```

#### Notes

- Added Parsing for Null Values in Inventory Line Items

## Previous Versions

### 2. ZCRMSDK - VERSION 2.2.3

#### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.2.3
2 dotnet add package ZCRMSDK -- VERSION 2.2.3
```

#### Notes

- This version supports custom log trace listeners.

### 3. ZCRMSDK - VERSION 2.2.2

#### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.2.2
2 dotnet add package ZCRMSDK -- VERSION 2.2.2
```

#### Notes

- This version performs extra validation before adding data to the request body.



## 4. ZCRMSDK - VERSION 2.2.1

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.2.1
2 dotnet add package ZCRMSDK -- VERSION 2.2.1
```

### Notes

- Fixed improper header processing in the GetAllDeletedRecords method.

## 5. ZCRMSDK - VERSION 2.2.0

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.2.0
2 dotnet add package ZCRMSDK -- VERSION 2.2.0
```

### Notes

- Supported headers and parameters in the getAllDeletedRecords method.

## 6. ZCRMSDK - VERSION 2.1.9

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.1.9
2 dotnet add package ZCRMSDK -- VERSION 2.1.9
```

### Notes

- SDK throws exception if the accounts scope (aaaserver.profile.READ) is not included in the generated grant token or when the user's email cannot be fetched with the generated access token.

## 7. ZCRMSDK - VERSION 2.1.8



### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.1.8
2 dotnet add package ZCRMSDK -- VERSION 2.1.8
```

### Notes

- Added validation for "Discount" in Inventory Line Items.

### 8. ZCRMSDK - VERSION 2.1.7

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.1.7
2 dotnet add package ZCRMSDK -- VERSION 2.1.7
```

### Notes

- Added data type handling for input field in Mass Update API.

### 9. ZCRMSDK - VERSION 2.1.6

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.1.6
2 dotnet add package ZCRMSDK -- VERSION 2.1.6
```

### Notes

- Supported fileUpload field in Record operations.

### 10. ZCRMSDK - VERSION 2.1.5

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.1.5
2 dotnet add package ZCRMSDK -- VERSION 2.1.5
```





## Notes

- The Operation time out exception.
- redirect\_uri key is no longer mandatory in the configuration.

## 11. ZCRMSDK - VERSION 2.1.4

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.1.4
2 dotnet add package ZCRMSDK -- VERSION 2.1.4
```

## Notes

- Handled long data type in Organization API's Employee count for the exception "OverflowException - Value was either too large for an Int32".
- Handled JSON Object data type in Field API's tool tip for the exception "System.ArgumentException - Can not convert Object to String()".

## 12. ZCRMSDK - VERSION 2.1.2

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.1.2
2 dotnet add package ZCRMSDK -- VERSION 2.1.2
```

## Notes

- Handled NullReferenceException.

## 13. ZCRMSDK - VERSION 2.1.1

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.1.1
2 dotnet add package ZCRMSDK -- VERSION 2.1.1
```

## Notes

- Added unhandled exceptions.



## 14. ZCRMSDK - VERSION 2.1.0

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.1.0
2 dotnet add package ZCRMSDK -- VERSION 2.1.0
```

### Notes

- Custom parameters supported in the below search records methods.
  - SearchByCriteria()
  - SearchByWord()
  - SearchByEmail()
  - SearchByPhone()

## 15. ZCRMSDK - VERSION 2.0.9

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.0.9
2 dotnet add package ZCRMSDK -- VERSION 2.0.9
```

### Notes

- Handled the change the OAuth token response.

## 17. ZCRMSDK - VERSION 2.0.8

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.0.8
2 dotnet add package ZCRMSDK -- VERSION 2.0.8
```

### Notes

- This versions supports file upload as a stream in the Bulk Write API.



## 18. ZCRMSDK - VERSION 2.0.7

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.0.7
2 dotnet add package ZCRMSDK -- VERSION 2.0.7
```

### Notes

- Added list validation.

## 19. ZCRMSDK - VERSION 2.0.6

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.0.6
2 dotnet add package ZCRMSDK -- VERSION 2.0.6
```

### Notes

- This version supports Bulk Read and Bulk Write APIs.

## 20. ZCRMSDK - VERSION 2.0.5

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.0.5
2 dotnet add package ZCRMSDK -- VERSION 2.0.5
```

### Notes

- Supported duplicate\_check\_fields in UpsertRecords().

## 21. ZCRMSDK - VERSION 2.0.4

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.0.4
2 dotnet add package ZCRMSDK -- VERSION 2.0.4
```



## Notes

- Supported double data type in list\_price.

## 22. ZCRMSDK - VERSION 2.0.3

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.0.3
2 dotnet add package ZCRMSDK -- VERSION 2.0.3
```

## Notes

- Added unit\_price data type check.

## 23. ZCRMSDK - VERSION 2.0.2

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.0.2
2 dotnet add package ZCRMSDK -- VERSION 2.0.2
```

## Notes

- User API support provided in SDK(create, update, delete).
- Convert mapping, section details in module layout, criteria details in module custom-view can be extracted.
- Picklist handling in module field.
- Appropriate response provided for files.

## 24.ZCRMSDK - VERSION 2.0.1

### Install command

```
1 Install-Package ZCRMSDK - VERSION 2.0.1
2 dotnet add package ZCRMSDK -- VERSION 2.0.1
```

## Notes

- SDK is compatible with Visual Studio 2015.



- Configuration details have to be passed only as a dictionary.
- Support ISO 8601-time format.

## 25. ZCRMSDK - VERSION 1.0.4

### Install command

```
1 Install-Package ZCRMSDK - VERSION 1.0.4
2 dotnet add package ZCRMSDK -- VERSION 1.0.4
```

### Notes

- Changing encoding standard from ASCII to UTF-8.

## 26. ZCRMSDK - VERSION 1.0.3

### Install command

```
1 Install-Package ZCRMSDK - VERSION 1.0.3
2 dotnet add package ZCRMSDK -- VERSION 1.0.3
```

### Notes

- Fix for the exception "exePath must be specified when not running inside a stand alone exe" in ASP.NET web applications.

## 27. ZCRMSDK - VERSION 1.0.2

### Install command

```
1 Install-Package ZCRMSDK - VERSION 1.0.2
2 dotnet add package ZCRMSDK -- VERSION 1.0.2
```

### Notes

- Fix for the invalid type cast Exception issue.

## 28. ZCRMSDK - VERSION 1.0.1



## Install command

```
1 Install-Package ZCRMSDK - VERSION 1.0.1
2 dotnet add package ZCRMSDK -- VERSION 1.0.1
```

## Notes

- Support is given for Organization Tax API.

## 29. ZCRMSDK - VERSION 1.0.0

## Install command

```
1 Install-Package ZCRMSDK - VERSION 1.0.0
2 dotnet add package ZCRMSDK -- VERSION 1.0.0
```

## Notes

- Initial release.

