Kaizala REST API programming tutorial (C#)

# Kaizala API documentation

Kaizala APIs are documented at <https://docs.microsoft.com/en-us/kaizala/connectors/api> and consuming them would need a valid Office365 organizational subscription or a Kaizala Pro subscription.

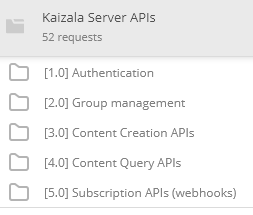
To validate your subscription, you should be able to sign-in to the Kaizala Management Portal - <https://manage.kaiza.la> with your account.

# Prerequisites

To use Kaizala APIs, you need to be registered on the Kaizala platform as a “connector”. For connector setup, please refer the documentation at <https://docs.microsoft.com/en-us/kaizala/connectors/setup>

For the rest of the documentation you will need the below:

1. Connector ID and secret (got from your connector on <https://manage.kaiza.la> )
2. Postman REST client (available at <https://www.getpostman.com/> )
3. Kaizala Postman collection [[link](https://app.getpostman.com/run-collection/f68a8abec784cc00b0b9#?env%5BKaizala-APIs-environment%5D=W3siZW5hYmxlZCI6dHJ1ZSwia2V5IjoibW9iaWxlLW51bWJlciIsInZhbHVlIjoiIiwidHlwZSI6InRleHQifSx7ImVuYWJsZWQiOnRydWUsImtleSI6ImFjY2Vzcy10b2tlbiIsInZhbHVlIjoiIiwidHlwZSI6InRleHQifSx7ImVuYWJsZWQiOnRydWUsImtleSI6InRlc3QtZ3JvdXAtaWQi)]. You should see this collection in your Postman window



(*Caveat: do not open this link in incognito mode – it would render the json instead of launching Postman client*)

1. Visual Studio 2015 (or above)

# Brief overview of APIs

Kaizala service uses token based authentication. Below are the different ways of authenticating with Kaizala:

1. User token
2. Group token
3. OAuth
4. Tenant token

For this article, we would be using the first mechanism – user token. For this you will need a mobile phone with Kaizala installed.

Steps involved in authentication:

1. Generate PIN for the mobile endpoint
2. Login with PIN and Application / Connector ID to get the RefreshToken
3. Use the RefreshToken to generate the AccessToken

The AccessToken generated will be subsequently used for calling Kaizala APIs.

Note: Code given below is for reference purposes only.

* AccessToken is valid for 24 hours
* RefreshToken is valid for 365 days

Samples for generating these tokens are documented below.

# Setting up Visual Studio

1. Create a new project (Windows Console application)
2. Add nuget references to below packages
   1. NewtonSoft json [*used to serialize / deserialize json*]
   2. RestSharp [*used to send http requests*]

# Getting code from Postman

Postman provides a way to generate source code in various languages. For this article, we will be looking at C#.

1. To get the code, click on “**Step 1 – Generate pin**” in the Postman collection.
2. Click on code to launch the code popup and select **C#(RestSharp)** from the drop down

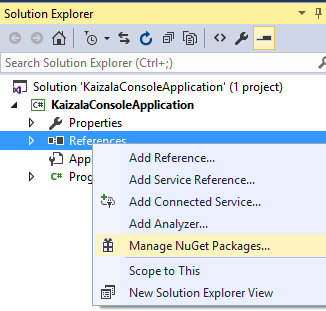


1. Click on **Copy to Clipboard** to copy the code snippet

# Console application to create and send message to a group

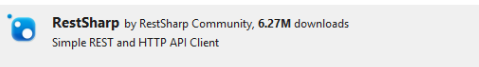
### Step1: Add references to RestSharp and Newtonsoft.Json

* In solution explorer right click on References and click “Manage NuGet Packages”



Search for Newtonsoft and RestSharp and install them





Add references in the visual studio program

using Newtonsoft.Json;

using RestSharp;

### Step2: Declare variables to cache values

### string accessToken = "";

string applicationId = " XYZABCXYZABCXYZABCXYZABCXYZABCXYZABCXYZABCXYZABCXYZABC";

string applicationSecret = "QWERTYUI";

string endpointUrl = "";

string mobileNumber2 = "+917995552658"; // to be added as member to the group

string refreshToken = "";

### Step3: Generate PIN with mobile number

Console.WriteLine("Enter your mobile number: ");

string MobileNumber = Console.ReadLine();

var client = new RestClient("https://api.kaiza.la/v1/generatePin");

var request = new RestRequest(Method.POST);

request.AddHeader("postman-token", "090af9a6-a234-7c7c-4431-9d6af937a185");

request.AddHeader("cache-control", "no-cache");

request.AddHeader("content-type", "application/json");

request.AddParameter("application/json", "{\"mobileNumber\":\"" + MobileNumber + "\", applicationId:\""+ applicationId + "\"}", ParameterType.RequestBody);

IRestResponse response = client.Execute(request);

### Step4: Generate refreshToken with mobile number and PIN

Console.WriteLine("Enter the PIN: ");

string PIN = Console.ReadLine();

client = new RestClient("https://api.kaiza.la/v1/loginWithPinAndApplicationId");

request = new RestRequest(Method.POST);

request.AddHeader("postman-token", "35918c70-e897-b115-03be-cc03d0fb1c82");

request.AddHeader("cache-control", "no-cache");

request.AddHeader("content-type", "application/json");

request.AddParameter("application/json", "{\"mobileNumber\":\"" + MobileNumber + "\",\"applicationId\":\"" + applicationId + "\", \"pin\":" + PIN.ToString() + "}", ParameterType.RequestBody);

IRestResponse responsePin = client.Execute(request);

var loginWithPinAndApplicationIdResponse = JsonConvert.DeserializeObject<LoginWithPinAndApplicationIdResponse>(responsePin.Content);

refreshToken = loginWithPinAndApplicationIdResponse.RefreshToken;

endpointUrl = loginWithPinAndApplicationIdResponse.EndpointUrl;

### Step5: Generate accessToken with refreshToken

client = new RestClient("https://api.kaiza.la/v1/accessToken");

request = new RestRequest(Method.GET);

request.AddHeader("postman-token", "9324c36e-0c56-d145-871c-25353f69f8be");

request.AddHeader("cache-control", "no-cache");

request.AddHeader("refreshtoken", refreshToken);

request.AddHeader("applicationsecret", applicationSecret);

request.AddHeader("applicationid", applicationId);

IRestResponse responseAccessToken = client.Execute(request);

var refreshApplicationTokenApiResponse = JsonConvert.DeserializeObject<RefreshApplicationTokenApiResponse>(responseAccessToken.Content);

accessToken = refreshApplicationTokenApiResponse.AccessToken;

### Step6: Create a group with a number as member

client = new RestClient(endpointUrl + "/v1/groups");

request = new RestRequest(Method.POST);

request.AddHeader("postman-token", "6192e748-6f06-a90d-0d84-a57a7f04cdb5");

request.AddHeader("cache-control", "no-cache");

request.AddHeader("content-type", "application/json");

request.AddHeader("accesstoken", accessToken);

request.AddParameter("application/json", "{name:\"Kaizala Test group\", welcomeMessage:\"Welcome to group created via C# console application\", members:[\"" + mobileNumber2 + "\"], groupType:\"Group\"}", ParameterType.RequestBody);

IRestResponse responseCreateGroup = client.Execute(request);

var createGroupResponse = JsonConvert.DeserializeObject<CreateGroupResponse>(responseCreateGroup.Content);

string groupId = createGroupResponse.groupId;

### Step7: Send a text message to the group

client = new RestClient(endpointUrl + "/v1/groups/449b16d5-772d-4b25-8154-6d39ae357dfc/messages");

request = new RestRequest(Method.POST);

request.AddHeader("postman-token", "d2ec1078-71a6-07ff-fcd2-0505c9d89ef4");

request.AddHeader("cache-control", "no-cache");

request.AddHeader("content-type", "application/json");

request.AddHeader("accesstoken", accessToken);

request.AddParameter("application/json", "{message:\"Test message via C# console application\"}\r\n", ParameterType.RequestBody);

IRestResponse responseSendMessageToGroup = client.Execute(request);

### Below are the classes used for de-serializing response JSONs:

public class LoginWithPinAndApplicationIdResponse

{

[JsonProperty("refreshToken")]

public string RefreshToken { get; set; }

}

public class RefreshApplicationTokenApiResponse

{

[JsonProperty("accessToken")]

public string AccessToken { get; set; }

[JsonProperty("endpointUrl")]

public string EndpointUrl { get; set; }

}

public class CreateGroupResponse

{

/// <summary>

/// GroupId which was created

/// </summary>

[JsonProperty("groupId")]

public string groupId;

/// <summary>

/// Name of the conversation

/// </summary>

[JsonProperty("groupName")]

public string conversationName;

/// <summary>

/// if the members mentioned in request are added to group

/// </summary>

[JsonProperty("membersAdded")]

public bool membersAdded;

}

# References

1. Kaizala documentation: <https://docs.microsoft.com/en-us/Kaizala>
2. Postman REST client download: <https://www.getpostman.com/>