

MisskeySharp

What's this ?

MisskeySharp is a client library for .NET applications to utilize Misskey.

This library is not an official offering by the Misskey developers.

Example

The usage examples are as follows:

Get a new access token with "MiAuth"

Step 1: Get a URL for authorize app page

We will prepare to obtain the access token using "MiAuth".

```
var misskey = new MisskeyService("https://misskey.io/");
var authUri = misskey.GetAuthorizeUri(
    "サンプル (my app) " + DateTime.Now.ToString("yyyyMMdd-HHmss-fff"),
    "https://www.a32kita.net/favicon.ico",
    "https://dummy.a32kita.net/callback",
    MisskeyPermissions.Write_notes | MisskeyPermissions.Read_account);
```

In *MiAuth*, you can use any value for the `GetAuthorizeUri()` parameter because it allows you to set the application name, icon, and callback URL during access token acquisition.

Step 2: After the user approves the application in the browser, obtain the access token

```
await misskey.AuthorizeWithAuthorizeUriAsync(authUri);

// misskey.AccessToken: "98eY*****"; <= Access Token
```

The obtained access token will be stored in the `AccessToken` property.

Authenticate with the already obtained access token

If the initially obtained access token is being recorded, user approval will not be required from the second time onwards.

```
await misskey.AuthorizeWithAccessTokenAsync("98eY*****");
```

Post a note

We will post a new note. In the following example, we will post a note with the content 'hello, world!!' to the timeline for followers.

```
await misskey.Notes.Create(new Note()
{
    Text = "hello, world!!",
    Visibility = "followers",
});
Console.WriteLine("Completed: {0}", resp.CreatedNote.Id);
```

For the parameters specified here, please refer to the official documentation:

<https://misskey-hub.net/docs/api/endpoints/notes/create.html>

Retrieve a list of followed users

Retrieve a list of followed users by specifying the user ID.

▶ Example code

Receiving a timeline through the streaming API

Using the Streaming API to receive the timeline in real-time. Events will be triggered upon receiving new notes.

▶ Example code

Install

Now available on NuGet

```
NuGet\Install-Package MisskeySharp
```

Platform

MisskeySharp is designed for .NET Standard 2.0, making it available for a wide range of .NET applications.

Please refer to Microsoft's documentation for information on the targets that can apply .NET Standard 2.0 libraries: <https://learn.microsoft.com/ja-jp/dotnet/standard/net-standard?tabs=net-standard-2-0#select-net-standard-version>

Dependencies

- System.Text.Json

If you are using MisskeySharp via NuGet, it will be automatically installed.