**connect to git repository from console app using .net core**

1.creating the free account for git hub by using [Join GitHub · GitHub](https://github.com/signup)

2.GIT(Global information tracker)

3.created new repository

4.added this word file in that repository

5. To interact with GitHub, we'll need a Personal Access Token (PAT).

6.Following is a process to get personal access token of your Git hub Account.

1. Click on your profile picture in the upper-right corner of any GitHub page, then click Settings.
2. In the left sidebar, click Developer settings.
3. In the left sidebar, click Personal access tokens.
4. Click Generate new token.
5. Give your token a descriptive name, select the scopes or permissions you'd like to grant this token, and click Generate token.
6. Important: Make sure you copy your new personal access token now. You won’t be able to see it again!

Following is the simple console application that uses the Octokit.NET library to interact with the GitHub API. The Octokit.NET library is a GitHub API client library for .NET, allowing developers to work with GitHub repositories, users, and other resources in a .NET application.

Code:

using Octokit;

using System;

using System.Threading.Tasks;

namespace GitHubConsoleApp

{

class Program

{

static async Task Main(string[] args)

{

var client = new GitHubClient(new ProductHeaderValue("MyGitHubApp"));

var tokenAuth = new Credentials(""); // Replace with your actual token

client.Credentials = tokenAuth;

try

{

// Retrieve the user's information

var user = await client.User.Current();

Console.WriteLine($"You are logged in as {user.Login}");

// Example: Retrieve repositories for the authenticated user

var repositories = await client.Repository.GetAllForCurrent();

foreach (var repo in repositories)

{

Console.WriteLine(repo.Name);

}

}

catch (Exception ex)

{

Console.WriteLine("An error occurred: " + ex.Message);

}

}

}

}

Output:

A computer screen with white text

Description automatically generated