using System;  
using System.Collections;  
using System.Collections.Generic;  
using System.Reflection;  
  
/\*  
 \* External dependencies, OAuth 2.0 support, and core client libraries are at:  
 \*   https://developers.google.com/api-client-library/dotnet/apis/  
 \* Also see the Samples.zip file for the Google.Apis.Samples.Helper classes at:  
 \*   https://github.com/youtube/api-samples/tree/master/dotnet  
 \*/  
  
using DotNetOpenAuth.OAuth2;  
  
using Google.Apis.Authentication;  
using Google.Apis.Authentication.OAuth2;  
using Google.Apis.Authentication.OAuth2.DotNetOpenAuth;  
using Google.Apis.Samples.Helper;  
using Google.Apis.Services;  
using Google.Apis.Util;  
using Google.Apis.Youtube.v3;  
using Google.Apis.Youtube.v3.Data;  
  
namespace dotnet  
{  
  class my\_uploads  
  {  
    static void Main(string[] args)  
    {  
      CommandLine.EnableExceptionHandling();  
      CommandLine.DisplayGoogleSampleHeader("YouTube Data API: My Uploads");  
  
      var credentials = PromptingClientCredentials.EnsureFullClientCredentials();  
      var provider = new NativeApplicationClient(GoogleAuthenticationServer.Description)  
      {  
        ClientIdentifier = credentials.ClientId,  
        ClientSecret = credentials.ClientSecret  
      };  
      var auth = new OAuth2Authenticator<NativeApplicationClient>(provider, GetAuthorization);  
  
      var youtube = new YoutubeService(new BaseClientService.Initializer()  
      {  
        Authenticator = auth  
      });  
  
      var channelsListRequest = youtube.Channels.List("contentDetails");  
      channelsListRequest.Mine = true;  
  
      var channelsListResponse = channelsListRequest.Fetch();  
  
      foreach (var channel in channelsListResponse.Items)  
      {  
        var uploadsListId = channel.ContentDetails.RelatedPlaylists.Uploads;  
  
        CommandLine.WriteLine(String.Format("Videos in list {0}", uploadsListId));  
  
        var nextPageToken = "";  
        while (nextPageToken != null)  
        {  
          var playlistItemsListRequest = youtube.PlaylistItems.List("snippet");  
          playlistItemsListRequest.PlaylistId = uploadsListId;  
          playlistItemsListRequest.MaxResults = 50;  
          playlistItemsListRequest.PageToken = nextPageToken;  
  
          var playlistItemsListResponse = playlistItemsListRequest.Fetch();  
  
          foreach (var playlistItem in playlistItemsListResponse.Items)  
          {  
            CommandLine.WriteLine(String.Format("{0} ({1})", playlistItem.Snippet.Title, playlistItem.Snippet.ResourceId.VideoId));  
          }  
  
          nextPageToken = playlistItemsListResponse.NextPageToken;  
        }  
      }  
  
      CommandLine.PressAnyKeyToExit();  
    }  
  
    private static IAuthorizationState GetAuthorization(NativeApplicationClient client)  
    {  
      var storage = MethodBase.GetCurrentMethod().DeclaringType.ToString();  
      var key = "storage\_key";  
  
      IAuthorizationState state = AuthorizationMgr.GetCachedRefreshToken(storage, key);  
      if (state != null)  
      {  
        client.RefreshToken(state);  
      }  
      else  
      {  
        state = AuthorizationMgr.RequestNativeAuthorization(client, YoutubeService.Scopes.YoutubeReadonly.GetStringValue());  
        AuthorizationMgr.SetCachedRefreshToken(storage, key, state);  
      }  
  
      return state;  
    }  
  }  
}