# 将 Unity 游戏与 Steamworks 集成

## Cloud Saves – 是一种免费存储

官方Steamworks文档

<https://steamworks.github.io/>

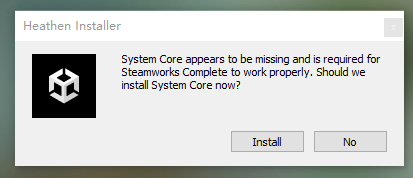
桌面的应用内购买解决方案，看起来 Steamworks API 是一个很好的解决方案

<https://blog.theknightsofunity.com/integrating-unity-game-steamworks/>

# 收费Assetstore.unity Steam API - Steamworks Complete

<https://assetstore.unity.com/packages/tools/integration/steam-api-steamworks-complete-246652>

<https://www.youtube.com/watch?v=6ujmZI1qUYI&t=2s>



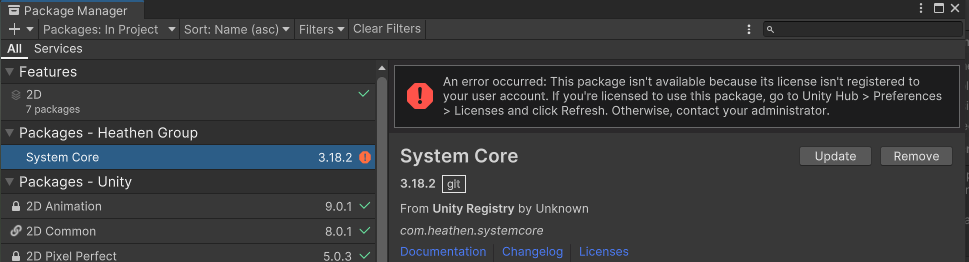
PackageManager's System Core install failed, Error Message: Unable to add package [https://github.com/heathen-engineering/SystemCore.git?path=/com.heathen.systemcore]:

Error when executing git command. error: RPC failed; curl 56 Recv failure: Connection was reset

fatal: expected flush after ref listing

## 导入包SystemCore

<https://github.com/heathen-engineering/SystemCore>



# 比较复杂https://steamworks.github.io/

<https://github.com/rlabrecque/Steamworks.NET>

<https://github.com/rlabrecque/Steamworks.NET-Example>

<https://github.com/rlabrecque/Steamworks.NET-Test>

# 免费Facepunch.Steamworks

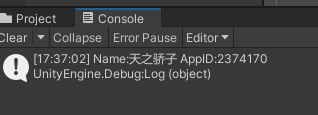
<https://github.com/Facepunch/Facepunch.Steamworks>

<https://wiki.facepunch.com/steamworks/>

## MP4 Help

<https://www.youtube.com/watch?v=mr5UpczYQME>

## 登录PC端Steam



登陆不对应APP ID 的账号，错误提示如下：

|  |
| --- |
| public class SteamAPI\_System2023 : MonoBehaviour  {  [Header("应用ID")]  public uint app\_id = 2374170;  void Start()  {  try  {  Steamworks.SteamClient.Init(app\_id);  ShowYourName();  }  catch(System.Exception e)  {  Debug.LogError(e);  Debug.Log(e.Message);  }  } |

SteamApi\_Init returned false. Steam isn't running, couldn't find Steam, AppId is ureleased, Don't own AppId.

# Steamworks Login

## Login

<https://partner.steamgames.com/steamdirect/>

# Unity CloudSaveService

CloudSaveException: Player ID is missing - ensure you are signed in through the Authentication SDK and try again.

Authentication SDK 没有玩家ID

Player id: eDQJddC7lYXVw8Han7mA56335r7f

## 关联Steam Account

IAuthenticationService

Task SignInWithSteamAsync(string sessionTicket, SignInOptions options = null);

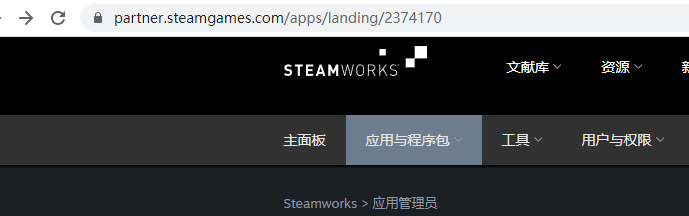
Task LinkWithSteamAsync(string sessionTicket, LinkOptions options = null);

# Houdini FxGPT 插件

<https://www.youtube.com/watch?v=mr5UpczYQME>

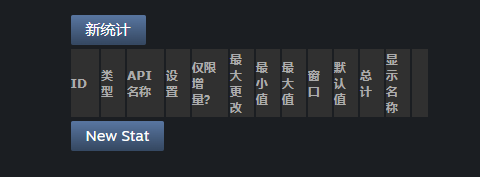
# Steam 统计(test 失败)

<https://partner.steamgames.com/apps/stats/2374170>











api\_game\_coin

# session ticked 如何获取

<https://www.youtube.com/watch?v=4CgeAxiS19s>

要在你的项目中设置 Steamworks.NET 库，你需要按照以下步骤操作：

1. 访问 Steamworks.NET 的 GitHub 页面（https://github.com/rlabrecque/Steamworks.NET）并下载最新版本的库。

2. 将下载的文件解压缩到你的项目中。

3. 在 Unity 编辑器中，将 `Plugins` 文件夹中的 `Steamworks.NET.dll` 和 `CSteamworks.dll` 文件拖放到你的项目的 `Assets` 文件夹中。

4. 在 Unity 编辑器中，选择 `Edit > Project Settings > Player`，然后在 `Other Settings` 面板中，将 `Scripting Define Symbols` 字段设置为 `DISABLESTEAMWORKS`。

现在，你应该可以在你的项目中使用 Steamworks.NET 库了。要验证库是否已正确安装，可以尝试编写一些使用 Steamworks.NET 的代码并检查它是否能够正常运行。

请注意，要使用 Steamworks.NET 库，你需要拥有一个有效的 Steamworks 开发者帐户，并且你的应用程序必须已在 Steamworks 后台注册。

DISABLESTEAMWORKS = 禁用Steam

## Demo 1 session

14000000DE458C2192984E80DD86EC0E0100100123EF236418000000010000000200000034A77CB3F33715E40A624C0121000000B20000003200000004000000DD86EC0E010010011A3A2400F61E4BD30409A8C000000000BF6321643F133D64010080060D00000000001691456567CF79A8629BED42479B58EC5E0179982DEB9A664D96A2E99522700D8C879A3FCAFF7F77E6EA267D0B2AE748D7A1D12AFCDEEFD4E3FD4397511A8C9801692E945AF6DB4721A91A415AE85650B5CCC2B498E6D2F331781C3CD8722036AF3AA95E5E699654329CA7B79488173168893A51122E22FCDDC2B701867D55AF

# Can you show me how to link a Steam account to a Unity account?

Callback<GetAuthSessionTicketResponse\_t> m\_AuthTicketResponseCallback;

HAuthTicket m\_AuthTicket;string m\_SessionTicket;

void SignInWithSteam()

{

// Create the callback to receive events when the session ticket

// is ready to use in the web API.

m\_AuthTicketResponseCallback = Callback<GetAuthSessionTicketResponse\_t>.Create(OnAuthCallback);

var buffer = new byte[1024];

m\_AuthTicket = SteamUser.GetAuthSessionTicket(buffer, buffer.Length, out var ticketSize);

Array.Resize(ref buffer, (int)ticketSize);

// The ticket is not ready yet, wait for OnAuthCallback.

m\_SessionTicket = BitConverter.ToString(buffer).Replace("-", string.Empty);

}

void OnAuthCallback(GetAuthSessionTicketResponse\_t callback)

{

// Call Unity Authentication SDK to sign in or link with Steam.

Debug.Log("Steam Login success. Session Ticket: " + m\_SessionTicket);

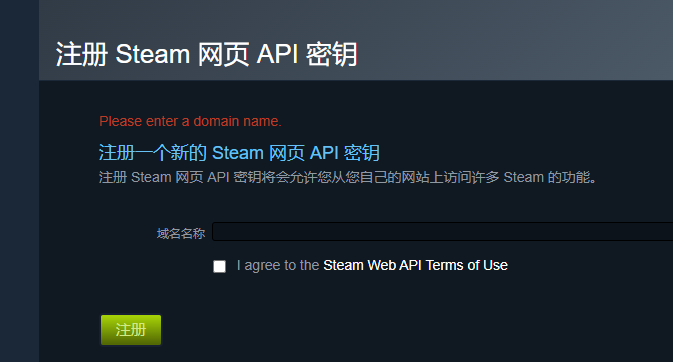
}

[Authentication]: Request failed: 401, {"title":"PERMISSION\_DENIED","detail":"unable to validate token","details":[],"status":401}

AuthenticationException: Invalid state for this operation. The player is already signing in.

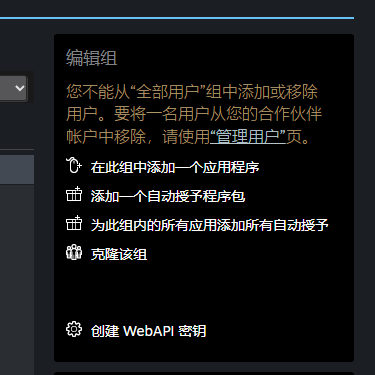
# unity AuthenticationService, unable to validate token 10002

<https://steamcommunity.com/dev/registerkey>



# Steam后台编辑组权限





# Unity 配置

<https://docs.unity.com/authentication/en/manual/set-up-steam-signin>



